Privacy-Enhancing Technologies for Medical and Genomic Data: From Theory to Practice

THÈSE N^{O} 8307 (2018)

PRÉSENTÉE LE 13 AVRIL 2018

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 1
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Jean Louis RAISARO

acceptée sur proposition du jury:

Prof. E. Bugnion, président du jury
Prof. J.-P. Hubaux, Prof. A. Telenti, directeurs de thèse
Prof. S. Murphy, rapporteur
Prof. E. De Cristofaro, rapporteur
Prof. C. Troncoso, rapporteuse

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

Science is not only a disciple of reason but, also, one of romance and passion.
Stephen Hawking

To my lovely wife, my parents, my brother,
and in loving memory of my grandfather

Abstract

The impressive technological advances in genomic analysis and the significant drop in the cost of genome sequencing are paving the way to a variety of revolutionary applications in modern healthcare. In particular, the increasing understanding of the human genome, and of its relation to diseases, health and to responses to treatments brings promise of improvements in better preventive and personalized medicine. Unfortunately, the impact on privacy and security is unprecedented. The genome is our ultimate identifier and, if leaked, it can unveil sensitive and personal information such as our genetic diseases, our propensity to develop certain conditions (e.g., cancer or Alzheimer's) or the health issues of our family. Even though legislation, such as the EU General Data Protection Regulation (GDPR) or the US Health Insurance Portability and Accountability Act (HIPAA), aims at mitigating abuses based on genomic and medical data, it is clear that this information also needs to be protected by technical means.

In this thesis, we investigate the problem of developing new and practical privacy-enhancing technologies (PETs) for the protection of medical and genomic data. Our goal is to accelerate the adoption of PETs in the medical field in order to address the privacy and security concerns that prevent personalized medicine from reaching its full potential. We focus on two main areas of personalized medicine: clinical care and medical research.

For clinical care, we first propose a system for securely storing and selectively retrieving raw genomic data that is indispensable for in-depth diagnoses and treatments of complex genetic diseases such as cancer. Then, we focus on genetic variants and devise a new model based on additively-homomorphic encryption for privacy-preserving genetic testing in clinics. Our model, implemented in the context of HIV treatment, is the first to be tested and evaluated by practitioners in a real operational setting.

For medical research, we first propose a method that combines somewhat-homomorphic encryption with differential privacy to enable secure feasibility studies on genetic data stored at an untrusted central repository. Second, we address the problem of sharing genomic and medical data when the data is distributed across multiple mistrustful institutions. We begin by analyzing the risks that threaten patients' privacy in systems for the discovery of genetic variants, and we propose practical mitigations to the re-identification risk. Then, for clinical sites to be able to share the data without worrying about the risk of data breaches, we develop a new system based on collective homomorphic encryption: it achieves trust decentralization and enables researchers to securely find eligible patients for clinical studies. Finally, we design a new framework, complementary to the previous ones, for quantifying the risk of unintended disclosure caused by potential inference attacks that are jointly combined by a malicious adversary, when exact genomic data is shared.

In summary, in this thesis we demonstrate that PETs, still believed unpractical and immature, can be made practical and can become real enablers for overcoming the privacy and security concerns blocking the advancement of personalized medicine. Addressing privacy issues in healthcare remains a great challenge that will increasingly require long-term collaboration among geneticists, healthcare providers, ethicists, lawmakers, and computer scientists.

Keywords: genomic privacy, medical data protection, genetic testing, inference attacks, homomorphic encryption, property-preserving encryption, differential privacy, re-identification, data sharing, privacy-enhancing technologies

Sommario

Gli impressionanti progressi tecnologici nell'analisi genomica e il notevole abbassamento del costo di sequenziamento stanno spianando la strada a una varietà di applicazioni rivoluzionarie nell'ambito della medicina moderna. In particolare, la crescente conoscenza del genoma umano e della sua interazione con la malattia, la salute e la risposta ai trattamenti promette notevoli miglioramenti nell'ambito della medicina preventiva e personalizzata. Sfortunatamente, tutto ciò ha un impatto sulla sfera privata dell'individuo senza precedenti. Il genoma è il nostro identificatore finale e, se fatto trapelare, può rivelare informazioni personali e sensibili sulle nostre malattie genetiche, la nostra propensione a sviluppare gravi patologie (per es. il cancro o il morbo di Alzheimer), o i problemi di salute dei nostri famigliari. Sebbene esistano leggi come la nuova *General Data Protection Regulation (GDPR)* europea o l'*Health Insurance Portability and Accountability Act (HIPAA)* americana che mirano a ridurre gli abusi basati sui dati medici e genetici, è sempre più evidente che queste informazioni richiedano anche una protezione di tipo tecnologico.

In questa tesi, esaminiamo le problematiche legate allo sviluppo di nuove tecnologie per la protezione della privacy dei dati medici e genetici con l'obbiettivo di accelerarne l'adozione e rispondere alle preoccupazioni sulla privacy e la sicurezza digitale che impediscono alla medicina personalizzata di raggiungere il suo pieno potenziale. Ci focalizziamo sull'utilizzo e la protezione dei dati in ambito clinico e di ricerca medica.

In ambito clinico, proponiamo inizialmente un nuovo sistema per la gestione sicura dei dati genetici grezzi, indispensabili per diagnosi approfondite e trattamenti di patologie genetiche complesse come il cancro. Successivamente, ci concentriamo sull'utilizzo clinico delle varianti genetiche, proponendo un nuovo modello basato sulla crittografia omomorfica che permette di effettuare test genetici e preservare, simultaneamente, la privacy dei pazienti. Questo modello, implementato nel contesto del trattamento dell'HIV, è il primo ad essere stato testato e valutato in un ambito operativo.

In ambito di ricerca, prima proponiamo un metodo che combina crittografia omomorfica e privacy differenziale per permettere l'esplorazione sicura di una base di dati genetici centralizzata. Poi, esaminiamo la condivisione sicura dei dati medici e genetici quando i dati sono distribuiti tra diversi siti. In primo luogo, analizziamo i rischi che minacciano la privacy nei sistemi di "discovery" per varianti genetiche e proponiamo delle contromisure pratiche contro il rischio di reidentificazione. In secondo luogo, proponiamo un sistema basato sulla crittografia omomorfica collettiva che permette a diversi siti clinici di proteggere collettivamente i propri dati da attacchi informatici e, allo stesso tempo, di condividerli in maniera sicura con ricercatori interessati a trovare soggetti idonei per studi clinici. Infine, presentiamo un nuovo framework, complementare ai sistemi precedenti, per quantificare la perdita involontaria di privacy, in seguito ad attacchi d'inferenza combinati, quando i dati genomici vengono condivisi con terzi.

In sintesi, in questa tesi dimostriamo che le tecnologie per la protezione della privacy, finora ritenute troppo complesse o dispendiose, possono essere usate efficacemente in campo medico rappresentando strumenti necessari per superare i problemi di privacy e sicurezza che bloccano l'avanzamento della medicina personalizzata. Tuttavia, affrontare le questioni legate alla privacy nel settore sanitario rimane un grande sfida che richiederà una sempre maggiore collaborazione a lungo termine tra esperti di genetica, etica, legislazione e tecnologia.

Parole Chiave: privacy genetica, protezione dati medici, test genetici, attacchi d'inference, crittografia omomorfica, crittografia deterministica, privacy differenziale, reidentificazione, condivisione dati, privacy-enhancing technologies

Acknowledgments

I would like to express my gratitude to all the exceptional people who have contributed to this thesis and who have made my PhD years such a great adventure.

First and foremost, I am extremely grateful to my advisor Prof. Jean-Pierre Hubaux for giving me the opportunity to work on such a fascinating topic in such a stimulating and positive environment, and above all, for providing me with the freedom, guidance, experience, inspiration and invaluable support throughout the entire PhD process. I have learnt a lot from him both on the professional and personal levels.

I am also grateful to my co-advisor Prof. Amalio Telenti, for his support, his precious advice and for his ability to always motivate and encourage me.

I would like to thank my thesis committee members: Prof. Edouard Bugnion, Prof. Emiliano De Cristofaro, Prof. Shawn Murphy, Prof. Carmela Troncoso for their time and effort spent reviewing this dissertation. In particular, I would like to thank Prof. Shawn Murphy for having given me the possibility to work with him and his team during my stay in Boston at Harvard University. It was an excellent and formative experience.

This thesis would not exist without the valuable contributions of all my co-authors. I am sincerely thankful to all of them for their strong support and for all I learnt by working with them. In particular, I would like to thank Prof. Erman Ayday for his friendship and crucial support at the beginning of the PhD process; during his post-doctoral stay at EPFL, we elaborated several of the ideas that are developed in this thesis. Special thanks also go to Prof. Jacques Fellay and Prof. Paul McLaren for all our fruitful discussions we had along the way.

The impact of my thesis would not have been the same without the precious hands-on experience that I had the chance to acquire by working hand in hand with our medical collaborators at the Lausanne University Hospital (CHUV). My deep appreciation goes to all of them: Olivier Michielin, Vincent Moser, Zoltan Kutalik, Nicolas Rosat, Raphaël Colsenet, Sylvain Pradervand and Nathalie Jacquemont. I thank them for the opportunity to work on relevant and concrete problems and their important contribution to this thesis. I am very grateful also to my collaborators at Harvard Medical School, Jeffrey Klann, Hossein Estiri, Kavishwar Wagholikar and Sarah Weiler for our very rich interactions and the productive environment during my internship. I would like to thank also all the talented bachelor's and master's students I had the pleasure to supervise and work with. Their contribution was instrumental, too.

EPFL is a fantastic working environment also thanks to the constant support of the non-academic staff. I would like to express my gratitude to Patricia and Angela for all their help with administration issues, to Marc-André and Yves for providing a first-rate computing infrastructure, and to Holly for her persistent efforts to improve my English.

I am thankful to all my colleagues and friends at LCA for sharing with me the good and the bad aspects of PhD life, and for the stimulating discussions and challenging ideas: Anh, Alexandra, Patricia, Andrey, Italo, Juan Ramòn, Huang, David, Joao, Christian, Florian, Mickaël, Ludovic, Mathias, Kévin, and Igor. Special thanks go to Zhicong who has been a great and reliable teammate since the beginning; We started and finished this adventure together, working with unceasing passion on the same topics. I have learnt a lot from him.

My PhD years would not have been so pleasant without the support of my EPFL Italian friends: Marco, Alberto and Amos. Thanks for all the fun moments spent together at the BC cafeteria and at bars in Lausanne and at our unforgettable Christmas dinners, to not mention my bachelor party in Barcelona, discussing about any sort of topic. I thank them for their sincere friendship and for making me feel a bit more home every day. Of course, these thanks also include Geraldine and Audrey. I am also thankful to my lifelong friends in Pavia with whom I have shared many special moments every time I went back home. Thank you for cherishing and nurturing our friendship regardless of the distance.

I would like to extend my warmest gratitude to my Sicilian family: Sergio, Giuliana and Daniela for their continuous encouragements and for always welcoming me with open arms. Thank you for your big heart and for making me fall in love with Sicily every time I come to Pozzallo.

Finally, I am wholeheartedly grateful to my family, and especially to my parents Arturo and Catherine, and my brother Claude, for their unconditional support and love at each step of the way. Thank you for all the invaluable things you have taught me in these years. I would not be the person I am without you.

Last, but not least, my sincerest and greatest thanks go to my wife, Nadia, for her endless understanding, support and encouragement during all the good and bad moments of this endeavor. Thank you for always being there for me. Thank you for being my love and my best friend. Thank you for enlightening my path and bringing great joy and happiness into my life. Without your constant support I would not have even imagined achieving this.

Contents

Chapter 1

Introduction

Privacy issues have a rather long history. The photo camera, introduced at the end of the 19th century, was the first revolutionary observation and identification tool that threatened the privacy of individuals. Since then, several other tools have become widespread, including video cameras, credit cards, Web browsers, and mobile phones. All these tools can reveal much of our personal information: for example, our presence and habits in various spheres of life, or our communication and mobility patterns [31]. The exponential use of DNA sequencing of these last few years has the potential to greatly exacerbate this problem. The genome represents our ultimate biological identity [160] and contains much sensitive information about our health and kin. If misused, it can pave the way to a variety of abuses and threats not yet fully understood.

The genomic era began in April 2003, when the Human Genome Project was declared complete. Subsequently, as a result of the impressive decrease in genome-sequencing costs and the rapid development of next-generation sequencing technologies, medicine has undergone an outstanding genomic revolution. At the time of writing, an increasing number of individuals are having their genome sequenced and it is not unrealistic to believe that, in the near future, most of us will be systematically screened in order to be able to benefit from new preventive diagnoses and treatments tailored to our genetic makeup. Even though the current understanding of the complex relation between genome, disease, health and phenotype is still in its infancy, it is already possible to collect, store, process and share genomic data in a way that was unthinkable only a decade ago. This rise in availability, use, and sharing of such genetic information, combined with their increasing integration with electronic health records (EHRs) systems brings great promise of improvements in better preventive and personalized medicine. Yet, it also raises unprecedented ethical and privacy concerns.

In general, access to genomic data prompts several important privacy problems: (i) The genome can be used to re-identify individuals, (ii) it can reveal information about their genetic diseases such as cystic fibrosis, and their predispositions to severe medical conditions such as Alzheimer's, cancer, or schizophrenia, (iii) it contains information about ancestors, siblings, and progeny, and sharing it could unveil telling insights into a whole family's health issues (possibly against the family's will), (iv) the genome does not (almost) change over time, hence revoking or replacing it (as with other forms of identification) is impossible, and (vi) it is already being used both in law enforcement and healthcare, thus prompting also numerous ethical issues. Furthermore, today, it is hard to assess or estimate the extent of the personal information that could be extracted or derived from the genome in the future.

In the last few years, millions of people have seen their medical data compromised by cyber attacks on data stored at hospitals, insurers, and clinical laboratories [16, 18, 17]. The healthcare sector has the highest incidence of attacks across all industries and EHRs are favorite targets; once genomic data needed for precision medicine is fully integrated with these records, the privacy impact of cyber attacks will be devastating. Some experts worry that genetic information could be exploited, for example, for identity theft, spear phishing, fraud [15, 12] or even genetic discrimination. Health or life insurance companies could obtain the genetic information of their customers and deny their services to people with a high susceptibility of developing a chronic disease, or employers could hire applicants based on their genetic features. Today, the science-fiction scenarios depicted by the movie of 1997 "GATTACA" do not look very unrealistic anymore. Leakage of genomic data not only could damage individuals but also medical institutions. For example, a hospital setting up a medical study on genomic data could be severely discredited if participants' genomic and clinical information was leaked or compromised. Moreover, the potential integration of genomic data with other privacy-sensitive data (e.g., location, ancestry and other online social network – OSN – data) could exponentially increases the risk of a privacy breach through cross-layer attacks.

Of course, in order to mitigate the risk of such discriminations, tight legislation such as the EU General Data Protection Regulation (GDPR), the US Health Insurance Portability and Accountability Act (HIPAA) and the US Genetic Information Non-discrimination Act (GINA), regulates the activities of companies and hospitals that manage personal health information. Yet, today there exist only a few specific regulatory standards that protect the sensitive health data (including genomic data) needed for precision medicine and they are mostly outdated and insufficient. For example, though fingerprints, long known to be useful for re-identification, are protected under the 2003 federal HIPAA Privacy and Security rules as "biometric identifiers", genomic data is still not [16]. Furthermore, even if regulations are in place, it is extremely difficult to protect medical and genomic data against the misdeeds of a hacker or a disgruntled employee. Hence, it is clear that legislation alone is not enough and this data also needs to be protected by technical means.

> *"There are privacy issues. We've got to figure out how do we make sure that if I donate my data to this big pool that it's not going to be misused, that it's not going to be commercialized in some way that I don't know about. And so we've got to set up a series of structures that make me confident that if I'm making that contribution to science that I'm not going to end up getting a bunch of spam targeting people who have a particular disease I may have."*

said US President Barack Obama, during a precision-medicine panel discussion of the precision medicine Initiative Cohort Program in February 2016 [13].

Information security is the practice of protecting information from unauthorized access, use, disclosure, disruption, modification, inspection, recording, or destruction [60]. Traditionally, information security has been widely used by governmental, military and financial institutions. Only in the last few years has the field of information security grown and evolved significantly in the medical context. Tools that protect informational privacy by eliminating or minimizing personal data without the loss of the functionality of the information system are usually called privacy-enhancing technologies (PETs) [204]. PETs generally protect users' privacy by either breaking the link between individuals' identities and the data they provide (e.g., by removing users' identities from published data), or by decreasing the amount of provided information (e.g., by using cryptographic tools or obfuscation techniques).

The idea of using technical solutions to guarantee the privacy of genomic data raises interesting debates. On one hand, the potential of genomic research for mankind is tremendous, and PETs can be considered as an obstacle to achieving the promise of personalized medicine. On the other hand, to expedite advances in personalized medicine, genome-phenome association studies often require the participation of a large number of research participants. To encourage individuals to enroll in such studies, it is crucial to adhere to ethical principles, such as autonomy, reciprocity and, more generally, trust (e.g., guarantee that medical and genomic data will not be misused) that PETs can provide.

Unfortunately, traditional approaches to privacy, such as de-identification or aggregation, currently used with EHR data, are ineffective in the genomic context because of the identifying nature of the genome itself [107, 148, 79]. For example, in 2013, Gymrek et al. [99] demonstrated the feasibility of re-identifying "anonymous" DNA donors by matching Y-chromosome data with names posted in popular genealogy Web sites. Similarly, in 2015, Hubert et al. [112] showed that it is possible, by using phenotypic traits, to de-anonymize individuals who posted their genetic information to DNA-sharing Web sites such as OpenSNP [152]. More recently, Lippert et al. [130] even proposed new techniques for identifying "anonymous" individuals by predicting their facial traits from the DNA.

Developing new PETs for medical and genomic data presents unique technical challenges, due to the architecture of the human genome, to the high dimensionality of the data, to the complicated genome-phenome interaction, to the variety of stakeholders involved, to the diversity of the legal frameworks, and to the rapidly evolving knowledge in the medical field. Therefore, PETs that are already being used in other domains (e.g., e-voting, location privacy, secure database management systems, fintech) cannot be simply transferred and applied to the medical context. A sector-specific approach and a thorough understanding of the complex privacy and security requirements, and of the properties of the data, are necessary in order to adapt established PETs and to develop new ones that can have a tangible impact on the way medical and genomic data is managed today.

For example, current PETs based on cryptographic techniques prevent unauthorized users from "viewing" the data, but typically reduce the efficiency of the algorithms by introducing storage and computational overhead that are often unacceptable. Whereas, PETs based on obfuscation techniques do not introduce computational overhead, but they reduce the accuracy (or utility) of the data hence are highly criticized by practitioners. Therefore, it is now more urgent than ever to develop new techniques that can guarantee the security and privacy of medical and genomic data, without significantly degrading the efficiency and accuracy of the use of this data in research and healthcare.

In this thesis, we investigate the problem of developing new privacy-preserving solutions for personalized medicine in order to address these important and impelling challenges. In particular, we study in detail the complex nature and use of medical and genomic data, and we develop new privacy-enhancing solutions for storing, processing and sharing it in two different areas of personalized medicine: clinical care and medical research. Furthermore, we implement these solutions and bring them beyond the status of academic prototypes by also testing and validating them in real-life operational environments. We demonstrate that, with reasonable cost, PETs are practical and scalable enablers for new use-cases that are otherwise impossible due to privacy and security constraints.

Finally, our ultimate goal in this thesis is to propose, throughout a series of concrete examples and contributions, a generic methodology that covers the entire development cycle of new privacy-enhancing technologies for medical and genomic data. This methodology should then guide future researchers in identifying the privacy and security requirements of a dynamic application field, i.e., the medical field, developing new and efficient technical solutions that address such requirements, and finally deploying these solutions to the real world.

Contributions

In this thesis, we investigate the privacy and security problems regarding the management of medical and genomic data, and we **develop new privacy-enhancing technologies for the protection of this data**. In particular, we focus on the aspects of privacy and security related to the confidentiality and control of this data under the semi-honest and malicious-but-covert adversarial models. We provide solutions for two areas of personalized medicine: clinical care and medical research. For clinical care, we first propose a system for securely storing and retrieving raw genomic data that is indispensable for in-depth diagnoses and treatments of genetic complex diseases, such as cancer, and for clinical trials. Then, we focus on genetic variants and propose a new model for privacy-preserving genetic testing in the clinic. We implement the proposed solution in the context of HIV treatment and also collect and evaluate, to the best of our knowledge for the first time, the feedback of practitioners who use our privacy-preserving system.

For medical research, we first devise a method for securely exploring cohorts of genetic data stored on an untrusted central repository in the context of feasibility studies for medical research. Second, we look into genomic and medical data sharing, where the data is distributed across multiple mistrustful institutions. In this context, we start by analyzing the privacy risks that threaten systems for the discovery of genetic variants and propose practical mitigations to the re-identification risk. Then, we look into the protection of medical data confidentiality to mitigate the risk of data breaches and still enable the use of the data. We develop a new system that achieves trust distribution and enables clinical sites to securely share and to process sensitive medical and genetic information under encryption in order to find similar patients to be included in clinical studies. Finally, we design a new framework for systematically reasoning about the risk of unintended disclosure when exact genomic data is shared; this risk is caused by potential inference attacks jointly combined by a malicious adversary.

Our contributions are as follows:

1. Geneticists prefer to store patients' aligned, raw, genomic data, in addition to their variant calls (compact and summarized form of the raw data), mainly because of the

immaturity of bioinformatic algorithms and sequencing platforms and in order to conduct more in-depth analyses that would be impossible with just the information about the variants. Therefore, we propose a new privacy-preserving architecture for protecting the privacy of aligned, raw, genomic data. The raw genomic data of a patient includes millions of short reads, each composed of between 100 and 400 nucleotides. We propose storing these short reads at a biobank in encrypted form. The proposed scheme enables a medical unit (e.g., a pharmaceutical company or a hospital) to privately retrieve a subset of the short reads of the patients (which include a definite range of nucleotides, depending on the type of the genetic test) without revealing the nature of the genetic test to the biobank. Furthermore, the proposed scheme enables the biobank to mask particular parts of the retrieved short reads if (i) some parts of the provided short reads are out of the requested range, and/or (ii) the patient does not give consent to access some parts of the provided short reads (e.g., parts revealing sensitive diseases). We evaluate the proposed scheme to show the amount of unauthorized genomic data leakage it prevents. Finally, we implement the proposed scheme and assess its practicality. This work was done in collaboration with Sophia Genetics, a Swiss-based analytics company that provides clinical genomic services.

2. The implementation of genomic-based medicine is hindered by unresolved questions regarding data privacy and the delivery of interpreted results to healthcare practitioners. Hence, we propose a new privacy-preserving system for genetic tests that uses patients' genomic data encrypted under homomorphic encryption. We used DNA-based predictions of HIV-related outcomes as a use case to explore its acceptance in a clinical operational environment. In particular, we develop a new architecture (between the patient and the medical unit) and propose a "privacy-preserving genetic test with ancestry inference" by using homomorphic encryption, proxy re-encryption and secure two-party protocols. Assuming the whole genome sequencing is done by a certified institution, we propose to store patients' genomic data encrypted under their public keys at a centralized "storage and processing unit" (SPU). Our proposed solution enables the medical unit to securely compute the ancestry information of each patient from the encrypted genomic data and to retrieve the encrypted genomic data from the SPU in order to securely process it for genetic testing and preserve the privacy of patients' genomic data. We also implement the proposed model into a client-server system, and we deploy it at five outpatient clinics of the Swiss HIV Cohort Study (SHCS). We evaluate the feedback from physicians who tested our system on a total of 230 HIV-positive individuals genotyped at 4,149 genetic markers.

3. The re-use of patients' health records can provide tremendous benefits for clinical research. Yet, when researchers need to explore cohorts of patients for inclusion in clinical trials or population health studies, privacy issues represent one of the major obstacles to accessing the data; especially when sensitive/identifying data, such as genomic data, are involved. Hence, we design a new and efficient privacy-preserving explorer for genetic cohorts. To maximize its adoption, our solution is built on top of i2b2 (Informatics for Integrating Biology and the Bedside), the state-of-the-art open-source framework for clinical cohort exploration. Moreover, it uses cutting-edge privacy-enhancing technologies (PETs) such as lattice-based

somewhat-homomorphic encryption and differential privacy. Solutions involving homomorphic encryption are often believed to be costly and still immature for use in operational environments. Here, we show that, contrary to these assumptions, the proposed solution outperforms the state of the art by enabling a researcher to securely explore 3,000 genetic variants over a cohort of 5,000 individuals in less than five seconds with commodity hardware. We successfully deployed and tested our solution in the operational environment of the clinical research data-warehouse of the Lausanne University Hospital (CHUV).

4. Systems for the discovery of genetic variants, such as the Beacon Project of the Global Alliance For Genomics and Health (GA4GH), enable researchers to query participating sites (or "beacons") for the presence of a specified nucleotide at a given position within a chromosome. Recent work has demonstrated that, given a beacon with specific characteristics (relatively small sample size), an adversary who possesses part of the genome sequence of a specific individual can infer the membership of that individual in a beacon, by simply repeating queries for variants present in the individual's genome. Based on this work, we show that the original attack can be significantly improved by considering a smarter adversary that uses public knowledge about allele frequency distribution in order to perform the attack. Furthermore, we propose three practical strategies for mitigating the re-identification risks stemming from such an attack. The first two strategies manipulate the beacon such that the presence of rare alleles is obscured; the third strategy budgets the number of accesses per user for each individual genome. Using a beacon containing data from the 1000 Genomes Project, we demonstrate that the proposed strategies can effectively reduce re-identification risk in beacon-like datasets. These strategies are under evaluation by the European Bioinformatics Institute for deployment in the Elixir Beacon network.

5. Being able to share large amounts of sensitive clinical and genomic data across several institutions is crucial for precision medicine to progress. Unfortunately, because of the increasing number of health-data breaches, clinical sites are uncomfortable exposing their data to external parties if strong security and privacy guarantees are not in place. As a result, currently, only very limited datasets of *non-sensitive* and moderately useful information can be shared. We introduce MedCo, the first operational system that enables clinical sites to protect the confidentiality of their data by means of collective homomorphic encryption and an investigator for securely exploring *sensitive* medical information about patients. MedCo is built on top of established and widespread technology from the biomedical informatics community, such as i2b2 and SHRINE, and relies on state-of-the-art secure protocols for processing encrypted distributed data and complying with regulations. As such, MedCo can be easily adopted by clinical sites. This would pave the way to new unexplored data-sharing use cases. We demonstrate that MedCo scales to several clinical sites with millions of records by testing it on an oncology use-case with real somatic tumor data in a network of three institutions (EPFL, UNIL and CHUV).

6. One major obstacle to developing precision medicine to its full potential is the privacy risk stemming from the inference attacks that can be perpetrated when genomic data is disclosed to untrusted third parties. Even though the academic com-

munity has proposed many solutions to mitigate these attacks, as of this writing, these solutions have not been adopted by practitioners, mainly due to their impact on the data utility. Therefore, we introduce GenoShare, a framework that enables practitioners to systematically reason about the risk of revealing privacy-sensitive attributes (e.g., health status, kinship, physical traits) from disclosed genomic data. GenoShare enables data controllers to take informed decisions about sharing *exact genomic data*. Through a novel mechanism based on synthetic versions of individual genomes (i.e., *avatars*), GenoShare also prevents potential inferences when the decision is not to share. We demonstrate GenoShare's capabilities by instantiating it with three of the most important genomics-oriented inference attacks and showing how it can be used to detect leakage of sensitive attributes by using real data from the 1000 Genomes Project.

Thesis Outline

This thesis is organized as follows. First, in Chapter 2, we discuss the main genomic and cryptographic concepts used throughout this thesis. Then, in Part I, we address the challenge of protecting medical and genomic data in clinical care. In particular, we show in Chapter 3 how raw genomic data can be securely stored and retrieved for in-depth clinical analyses. In Chapter 4, we describe a new privacy-preserving model for genetic testing by using HIV treatment as a real-life use case. Finally, in Part II, we study the protection of medical and genomic data in the context of medical research. More specifically, in Chapter 5 we show a new privacy-preserving system for exploring genetic cohorts stored on a centralized (and potentially untrusted) database. In Chapter 6, we evaluate the privacy risks that affect genetic variants discovery systems and propose a set of practical mitigations to thwart re-identification attacks. In Chapter 7, we describe in detail a new system that enables the collective protection and privacy-conscious sharing of sensitive medical data. We conclude Part II, in Chapter 8, by proposing a new framework for systematically reasoning about inference risks that stem from genomic data disclosures.

Publications

Chapter 3 is an extended version of [32]. Chapter 4 is an extended version of [141] and also contains the findings in [162]. Chapter 5 contains the techniques and findings in [161]. Chapter 6 is an extended version of [163]. Chapter 7 rests on the results of [165]. Finally, Chapter 8 is an extended version of [164].

Chapter 2

Preliminaries

In this preliminary chapter, we briefly summarize the main concepts of genomics and cryptography used throughout this thesis.

2.1 Genomic Background

The human genome is encoded in double-stranded DNA molecules that consist of two complementary polymer chains. Each chain consists of simple units called nucleotides that are represented by letters of the alphabet $\{A, C, G, T\}$. The human genome is composed of over 3 billion nucleotides (letters) distributed along 23 pairs of chromosomes. Most of the DNA sequence is conserved across the whole human population and it is estimated that no more than 0.1% of the DNA differs between any two individuals [64]. It is this difference that influences an individual's health status and other aspects. In recent years, due to the decreasing cost of sequencing technologies, the use of genomic data has increased dramatically. Its applications can be grouped in four main areas: clinical care, genomic research, recreational genomics, and legal and forensic genomics:

- **Clinical Care.** It has been proved that mutations in an individual's genomic makeup can influence his or her health status. In particular, genetic mutations can be associated with change in the susceptibility to a certain disease and in the response to pharmaceutical agents. The latter is used particularly in oncology. Thus, early diagnosis and treatment can be tailored to an individual's genetic makeup [48].

- **Genomic Research.** Research studies that explore the genome function are being conducted worldwide. Researchers are discovering new associations between the genome and a number of disorders and variable responses to treatments almost on a weekly basis. The challenge is to identify and validate associations with *effect sizes*, sensitivity, and specificity that enable counseling about risk beyond what is predicted by traditional clinical factors [54].

- **Recreational Genomics.** In the last few years we have witnessed the rise of direct-to-consumer (DTC) services for medical data in general, and genomic data in particular. These services have made it affordable for individuals to become directly involved in the collection, processing, and even in the analysis of their medical and genomic data. Some well-known examples are 23andMe [19] and ancestry.com [1] that provide their costumers with reports, based on their genotype, about their ethnic mix, their potential kinship and disease susceptibility.

- **Legal and Forensic Genomics.** Given that genomic data has a static nature and uniquely identifies an individual, this information is used also for investigative purposes.

In this thesis, we focus on the protection of genomic data for the first two application areas.

Genetic Variation

As already mentioned, around 99.9% of the whole human genome is identical for any two individuals. The remaining part (∼0.1%) is referred to as *genetic variation.* Most commonly, this genetic variation comes from differences in single nucleotides, called *single nucleotide polymorphisms* (SNPs). For example, in Figure 2.1, two sequenced DNA fragments from two individuals contain a single different nucleotide at a particular genetic position (i.e., locus). Yet, there exist many other types of genetic differences, also involving multiple nucleotides, that are much less frequent such as *insertions/deletions* (INDELs), *duplications* (DUPs), *copy number variants* (CNVs) and more complex *structural variants* (SVs).

In the human population, a given genetic locus can have several possible versions (or alleles) with different genetic variations. Due to the diploid nature of somatic human cells, a human genome comprises two sets of chromosomes – one inherited from each parent. Therefore, each individual either has two copies of the same allele/variant (*homozygous*) or two copies of different alleles/variants (*heterozygous*). Genetic variants in a given individual genome are identified by comparing the genome with the *reference human genome*, a digital sequence of nucleotides considered representative of the human genetic makeup. In the vast majority of cases, a genetic variant is biallelic, i.e., it can take two different alleles: a reference allele, the one appearing on the reference human genome, and an alternate allele, the alternative version occurring in the human population. The presence of the latter is quantified by the *alternate allele frequency.* It is important to note that a variant is such only when it carries at least one alternate allele. Hence, at a given locus, an individual can be homozygous reference (i.e., taking two reference alleles) and

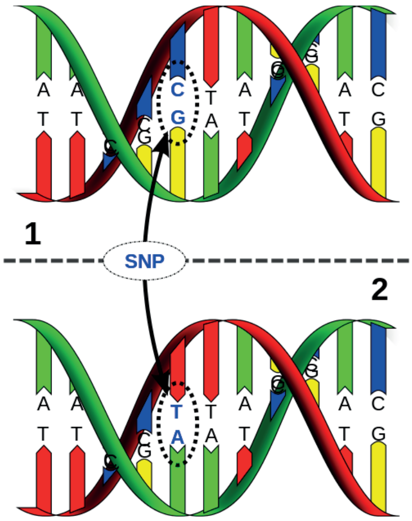


Figure 2.1: Single nucleotide polymorphism (SNP) with alleles C and T (© David Hall, License: Creative Commons).

not have a variant, or have a variant and then be heterozygous (one reference and one alternate alleles) or homozygous alternate (two alternate alleles). For example, assume that '$\mathcal{A}$' is the alternate allele and '$\mathcal{R}$' the reference allele for a potential SNP position (both '$\mathcal{A}$' and '$\mathcal{R}$' are from the set $\{A, C, G, T\}$). Then, in Table 2.1, we illustrate the probable states of this SNP for an offspring, for different combinations of the father's and mother's alleles. In this thesis, we encode the value (or genotype) of a variant with the number of alternate alleles it contains, i.e., 0, 1 or 2.

Genotype-Phenotype Association

Genomic variants can be associated with diseases and traits more generally denoted as phenotypes. For example the presence of some specific variants can either increase the predisposition of an individual to develop a disease at some point in time, or be protective with respect to that disease. These associations are usually studied through genome-wide association studies (GWAS) phenome-wide association studies (PheWAS).

The strength of the genome-phenome association is generally quantified by the *effect size*, denoted as $\omega = log(OR)$, where OR is the *odds ratio*. The *odds* represent the ratio between the probability of disease occurrence in a given group and the probability of non-occurence in the same group. The *odds ratio* is the odds in the group of individuals carrying a genetic variant divided by the odds in the group of those not carrying it. If there are N_{dg} individuals carrying a disease and a variant, N_{hg} healthy individuals carrying the same variant, N_{dn} individuals carrying the disease but not the variant, and N_{hn} healthy individuals not carrying the variant, then the OR is $\frac{N_{dg}/N_{hg}}{N_{dn}/N_{hn}}$.

		MOTHER	
		$\mathcal{R}$	$\mathcal{A}$
FATHER	$\mathcal{R}$	$\mathcal{RR}$ (homozygous reference)	$\mathcal{AR}$ (heterozygous)
	$\mathcal{A}$	$\mathcal{RA}$ (heterozygous)	$\mathcal{AA}$ (homozygous alternate)

Table 2.1: Possible SNP states for an offspring, given different combinations of his parents' alleles for the same SNP position. '$\mathcal{R}$' represents the reference allele and '$\mathcal{A}$' represents the alternate allele.

Genetic Correlations

Because genetic segments (or haplotypes) are inherited in blocks, physically close variants are very often correlated. Such a correlation is called *linkage disequilibrium* (LD). LD is usually measured by the correlation coefficient r^2 and it can be used to infer the value (or genotype) of a given variant from the genotype of other variants[1].

Beyond intra-genome correlations, there exist inter-genome correlations that stem from sexual reproduction. During the reproduction process, at each genetic position, a child inherits one allele from his mother and one from his father. Under the Mendelian inheritance assumption, each allele of a parent is passed to the child with equal probability 0.5, independently of the other positions. Moreover, given both parents' genomes, a child's genome is conditionally independent of all other ancestors' genomes. We illustrate the Mendelian inheritance probabilities in Table 2.2

[1] $r^2 \in [-1, 1]$, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation

		MOTHER								
		$\mathcal{RR}$			$\mathcal{RA}$			$\mathcal{AA}$		
FATHER	$\mathcal{RR}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$
		1	0	0	0.5	0.5	0	0	1	0
	$\mathcal{RA}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$
		0.5	0.5	0	0.25	0.5	0.25	0	0.5	0.5
	$\mathcal{AA}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$	$\mathcal{RR}$	$\mathcal{RA}$	$\mathcal{AA}$
		0	1	0	0	0.5	0.5	0	0	1

Table 2.2: Mendelian inheritance probability for the genotype of an offspring, given different genotypes for the parents. R represents the reference allele and A represents the alternate allele.

2.2 Cryptographic Background

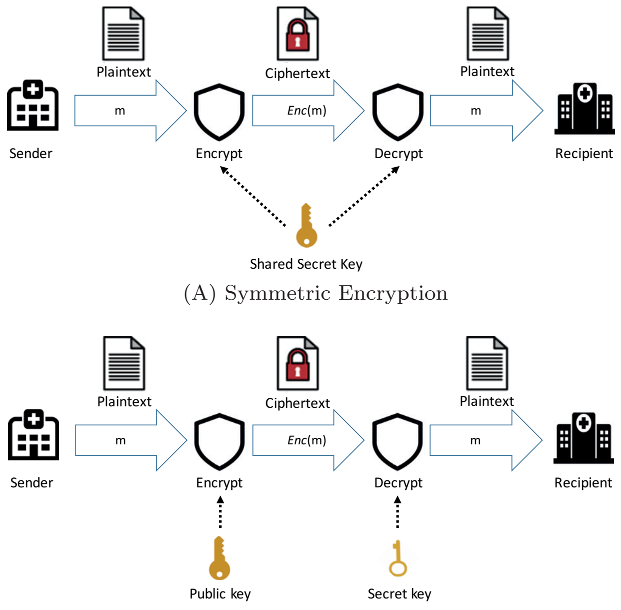
Modern cryptography can be defined as the scientific study of techniques for securing digital information, transactions, and distributed computations [120].

Usually, the distinction is made between symmetric encryption and asymmetric encryption. As illustrated in Figure 2.2A, in symmetric encryption, the same key is shared between the sender and the receiver and it is used to encrypt and decrypt the message. Symmetric encryption is mainly used for message encryption, message authentication codes and hash functions. Instead, in asymmetric encryption, different keys are used to encrypt and decrypt the message, as shown in Figure 2.2B. The receiver is provided with a pair of keys composed of a *public* key and a *secret* key. The sender uses the receiver's public key to encrypt the *plaintext*, and the receiver uses his secret key to decrypt the *ciphertext*. Asymmetric encryption is typically used for key exchange and digital signatures. In general, cryptographic techniques reduce the efficiency of the algorithms, introducing storage and computational overhead, while preventing users from using the data and adversaries not in possession of the secret key from viewing or tampering with the data. Yet, there exist special types of encryption that enable operations of the encrypted data at the expense of decreased security or efficiency. In the following, we provide a high-level explanation of property-preserving encryption and homomorphic encryption.

Property-Preserving Encryption

Property-preserving encryption (PPE) is a special type of encryption that, as opposed to standard probabilistic encryption which makes ciphertext indistinguishable and unusable, preserves some of the properties of the plaintexts. Yet, it also leaks these properties. For example, deterministic encryption encryption (DTE) [42], which always produces the same ciphertext for a given plaintext and key, preserves and reveals the equality property of the plaintext. By encrypting with DTE a set of messages, the resulting ciphertexts reveal the equality of the original messages. More formally, for $A, B \subseteq \mathbb{N}$ with $|A| \leq |B|$, a function $f : A \rightarrow B$ is *equality-preserving* if for all $i, j \in A$, $f(i) = f(j)$ iff $i = j$. We say that an encryption scheme with plaintext and ciphertext-spaces $\mathcal{D}$ and $\mathcal{R}$, respectively, is deterministic if $\mathrm{E}_{\mathrm{DTE}}(K, \cdot)$ is an equality-preserving function from $\mathcal{D}$ to $\mathcal{R}$ for all $K \in \mathcal{K}$ (where $\mathcal{K}$ is the key space).

Similarly, order-preserving encryption (OPE) preserves and reveals the order property of the plaintext. OPE was initially proposed by Agrawal *et al.* [21] and recently re-visited



(A) Symmetric Encryption

(B) Asymmetric Encryption

Figure 2.2: High-level representation of symmetric encryption (A) and asymmetric encryption (B).

by Boldyreva *et al.* [47] and Popa *et al.* [155]. By encrypting a set of messages with OPE, the resulting ciphertexts reveal the order of the messages. More formally, for $A, B \subseteq \mathbb{N}$ with $|A| \leq |B|$, a function $f : A \rightarrow B$ is *order-preserving* if for all $i, j \in A$, $f(i) > f(j)$ iff $i > j$. We say that an encryption scheme with plaintext and ciphertext-spaces $\mathcal{D}$ and $\mathcal{R}$, respectively, is order-preserving if $\mathrm{E}_{\mathrm{OPE}}(K, \cdot)$ is an order-preserving function from $\mathcal{D}$ to $\mathcal{R}$ for all $K \in \mathcal{K}$ (where $\mathcal{K}$ is the key space).

PPE-based schemes have several advantages and are mainly used in the context of encrypted database systems (e.g., CryptDB [156]) as they enable relational database systems to operate on encrypted data in the same way as they would operate on the plaintext data. Yet, they provide less security guarantees than standard probabilistic encryption schemes, as they are vulnerable to inference attacks due to the amount of information they leak. Hence, their application has to be carefully assessed.

Homomorphic Encryption

Homomorphic encryption (HE) is a special type of asymmetric encryption that supports computation on encrypted data as illustrated in Figure 2.3. Homomorphic encryption is probabilistic. It provides semantic security, meaning that no adversary without the secret key can compute any function of the plaintext from the ciphertext. In 2009, Craig Gentry [91] introduced for the first time a special type of HE that enables for *arbitrary computations* on ciphertexts called fully homomorphic encryption (FHE).

More formally, FHE could be described as follows. Let $\mathsf{CS}(K_p, K_s, P, C, \mathcal{E}, \mathcal{D})$ be a cryptosystem with the public key space K_p, the secret key space K_s, the plaintext space P, the ciphertext space C, the encryption function $\mathcal{E} : P \times K_p \rightarrow C$, and the decryption function $\mathcal{D} : C \times K_s \rightarrow P$. We say that the cryptosystem CS is *fully homomorphic* if

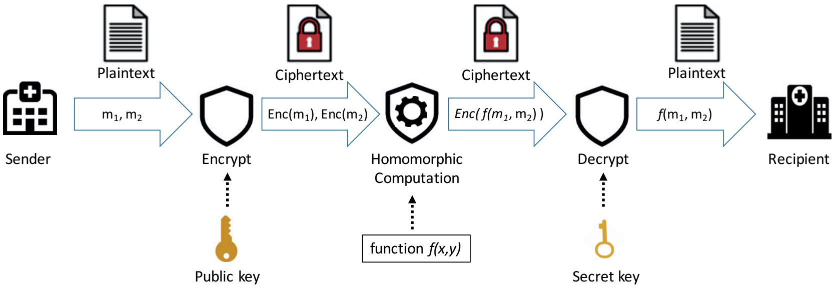


Figure 2.3: High-level representation of homomorphic encryption enabling computations on encrypted data. Different keys are used to encrypt and decrypt messages.

and only if for any function $f : P \times P \to P$ in the plaintext domain, there exists another function $h : C \times C \to C$ in the ciphertext domain such that

$$\mathcal{D}(h(\mathcal{E}(m_1, k_p), \mathcal{E}(m_2, k_p)), k_s) = f(m_1, m_2) \tag{2.1}$$

for any $m_1, m_2 \in P$, $(k_p, k_s) \in K_p \times K_s$.

Yet, despite its complete functionality, FHE is unpractical as it introduces enormous computational and storage overheads that make it unusable for real-world applications. For this reason, many variations of FHE have been proposed in the past few years with the goal to improve efficiency by sacrificing some flexibility. Such cryptosystems are called *practical* homomorphic cryptosystems, and according to their functionality, they can be classified as *additively* homomorphic if they satisfy only the addition of ciphertexts, *multiplicatively* homomorphic if they satisfy only multiplication, or *somewhat* homomorphic if they support addition and a limited number of multiplications.

Part I

Protecting Medical and Genomic Privacy in Clinical Care

Chapter 3

Privacy-Preserving Processing of Raw Genomic Data

The exponentially increasing amount of raw genomic data produced by next-generation sequencing technologies for clinical purposes poses important challenges in terms of secure storage and privacy-preserving management. In this chapter, we propose the first privacy-preserving system for the storage and selective retrieval of encrypted short reads in sequence alignment/map (SAM) files from a centralized biobank.

3.1 Introduction

Genomics holds great promise for better predictive medicine and improved diagnoses. However, genomics also comes with a risk to privacy [31] (e.g., revelation of an individual's genetic properties due to the leakage of his genomic data). An increasing number of medical units (pharmaceutical companies or hospitals) are willing to outsource the storage of genomes generated in clinical trials. Acting as a third party, a biobank could store patients' genomic data that would be used by the medical units for clinical trials. In the meantime, the patient can also benefit from the stored genomic information by interrogating his own genomic data, together with his family doctor, for specific genetic predispositions, susceptibilities and metabolical capacities. The major challenge here is to preserve the privacy of patients' genomic data while allowing the medical units to operate on specific parts of the genome (for which they are authorized).

We can put the research on genomic privacy in three main categories: (i) re-identification of anonymized genomic data [209, 85, 99, 220], (ii) cryptographic algorithms to protect genomic data [200, 114, 35, 29, 30, 28], and (iii) private clinical genomics [59]. To the best of our knowledge, none of the existing works on genomic privacy addresses the issue of private processing of aligned, raw genomic data (i.e., sequence alignment/map files), which is crucial to enable the use of genomic data in clinical trials.

Sequence alignment/map (SAM and its binary version BAM) files are the *de facto* standards used to store the aligned[1], raw genomic data generated by next-generation DNA sequencers and bioinformatic algorithms. There are hundreds of millions of short

[1] Alignment is with respect to the reference genome, which is assembled by the scientists.

reads (each including between 100 and 400 nucleotides) in the SAM file of a patient. Typically, each nucleotide is present in several short reads in order to have sufficiently high coverage of each patient's DNA.

In general, geneticists prefer storing aligned, raw genomic data of the patients (i.e., their SAM files), in addition to their variant calls (which include each nucleotide on the DNA sequence once, hence is much more compact) due to the following reasons: (i) Bioinformatic algorithms and sequencing platforms for variant calling are currently not yet mature, and hence geneticists prefer to observe each nucleotide in several short reads. (ii) If a patient carries a disease, which causes specific variations in the diseased cells (e.g., cancer), his DNA sequence in his healthy cells will be different from those diseased. Such variations can be misclassified as sequencing errors by only looking at the patient's variant calls (rather than his short reads). And (iii) due to the rapid evolution of genomic research, geneticists do not know enough to decide which information should really be kept and what is superfluous, hence they prefer to store all outcome of the sequencing process as SAM files.

In this chapter, we propose a privacy-preserving system for the storage, retrieval and processing of SAM files. In a nutshell, the proposed scheme stores the encrypted SAM files of the patients at a *biobank* and enables a medical unit to retrieve a range of nucleotides on the DNA sequence (for a genetic test) while protecting the patients' genomic privacy. It is important to note that the proposed scheme enables the privacy-preserving processing of the SAM files both for individual treatment (when the medical unit is embodied in a hospital) and for genetic research (when the medical unit is embodied in a pharmaceutical company). The main contributions of this chapter are summarized in the following:

- We develop a privacy-preserving framework for the retrieval of encrypted short reads (in the SAM files) from the biobank without revealing the scope of the request to the biobank.
- We develop an efficient system for obfuscating (i.e., masking) specific parts of the encrypted short reads that are out of the requested range of the medical unit (or that the patient prefers to keep secret) at the biobank before providing them to the medical unit.
- We show the benefit of masking by evaluating the information leak to the medical unit, with and without the masking is in place.
- We implement the proposed privacy-preserving system, evaluate its efficiency, and show its practicality by using real genomic data.

We summarize the notations used in this Chapter in Table 3.1.

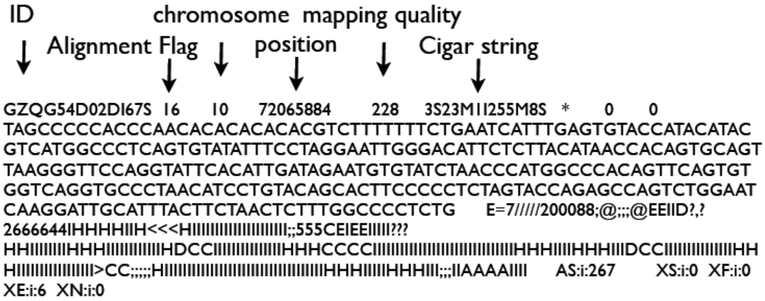
3.2 Sequence Alignement Map (SAM) Format

The DNA sequence data produced by DNA sequencers consists of millions of short reads, each typically including between 100 and 400 nucleotides (A,C,G,T), depending on the type of sequencer (Fig. 3.1). These reads are randomly sampled from a human genome. Each read is then bioinformatically treated and positioned (aligned) to its genetic location

Notation	Description
CS	Cigar string of a short read
OPE	Order-preserving encryption
SC	Stream cipher encryption
SE	Semantically secure symmetric encryption
SAM	Sequence aligned/map
$L_{i,j} = \langle x_i \vert y_j \rangle$	Position of a short read, where x_i represents the chromosome number and y_j represents the position of its first aligned nucleotide on chromosome
$\mathfrak{M}_P(.)$	Mapping function for patient P used before OPE encryption
$K_{i,j}$	Key shared between parties i and j for semantically secure encryption scheme
K_P^O	OPE key for patient P
M_P	Master key of patient P used to generate the keys of the stream cipher
$K_P^{C_{i,j}}$	Stream cipher key used to encrypt the content of the short read at position $L_{i,j}$
$C_{i,j}$	Content of the short read at position $L_{i,j}$
$S_{i,j}$	Random salt to provide different keys for the short reads with the same positions
H	pseudorandom function
$\mathcal{E}(\mathcal{K}_i, m)$	Public-key encryption of message m under public key of i
$\mathrm{E}_{\mathrm{SE}}(K_{i,j}, m)$	Semantically secure symmetric encryption of message m under symmetric key $K_{i,j}$
$\mathrm{E}_{\mathrm{OPE}}(K_P^O, m)$	OPE of message m using the OPE key of P
$\mathrm{E}_{\mathrm{SC}}(K_P^{C_{i,j}}, C_{i,j})$	Stream cipher encryption of short read content
$[R_L, R_U]$	Requested nucleotide range
Γ	Maximum number of nucleotides in a short read
V_m	Binary masking vector indicating the positions to be masked by the biobank
Π_P	Set of positions for which the patient P does not give consent
Δ	Set of encrypted (with OPE) positions of short reads to be retrieved from the biobank

Table 3.1: Notation used throughout the chapter.

to produce a so-called SAM file. There are hundreds of millions of short reads in the SAM file of one patient.



```
GZQG54D02DI67S  16    10    72065884     228    3S23M1I255M8S  *    0    0
TAGCCCCCACCCAACACACACACACGTCTTTTTTTCTGAATCATTTGAGTGTACCATACATAC
GTCATGGCCCTCAGTGTATATTTCCTAGGAATTGGGACATTCTCTTACATAACCACAGTGCAGT
TAAGGGTTCCAGGTATTCACATTGATAGAATGTGTATCTAACCCATGGCCCACAGTTCAGTGT
GGTCAGGTGCCCTAACATCCTGTACAGCACTTCCCCCTCTAGTACCAGAGCCAGTCTGGAAT
CAAGGATTGCATTTACTTCTAACTCTTTGGCCCCTCTG     E=7/////200088;@;;;@EEIID?,?
2666644IHHHHIIH<<<HIIIIIIIIIIIIIIIIIII;;555CEIEEIIIII???
HHIIIIIIIHHHIIIIIIIIIIIHDCCIIIIIIIIIIIIIHHHCCCCIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIHHHIIIIHHHIIIDCCIIIIIIIIIIIIIHH
HIIIIIIIIIIIIIIII>CC;;;;;HIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIHHHIIIIIHHHIII;;;IIAAAAIIII     AS:i:267      XS:i:0  XF:i:0
XE:i:6  XN:i:0
```

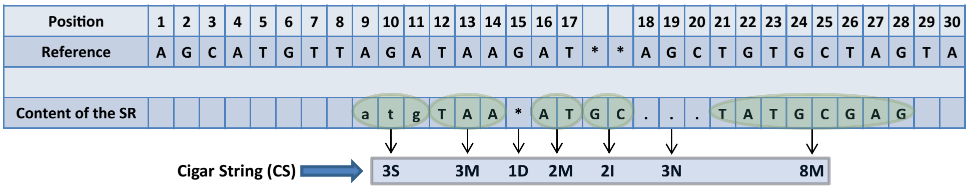
Figure 3.1: Format of a short read in a SAM file.

The privacy-sensitive fields of a short read are (i) its position with respect to the reference genome, (ii) its *cigar string* (CS), and (iii) its content (including the nucleotides from $\{A, T, G, C\}$). For simplicity of the presentation, from here on, we focus on these three fields only. We note that the rest of the short read does not contain privacy sensitive information about the patient, hence the rest of the short read can be encoded as a vector and provided to the medical unit, along with the aforementioned privacy-sensitive fields.

A short read's position denotes the position of the first aligned nucleotide in its content, with respect to the reference genome. The position of a short read is in the form $L_{i,j} = \langle x_i | y_j \rangle$, where x_i represents the chromosome number ($x_i \in [1, 23]$ as there are 23 chromosomes in the human genome) and y_j represents the position of its first aligned nucleotide on chromosome x_i ($y_j \in [1, 240\text{M}]$ as the maximum number of nucleotides on

Operation	Description
M	alignment match (can be a sequence match or mismatch)
I	insertion to the reference
D	deletion from the reference
N	skipped region from the reference
S	soft clipping (misalignment), clipped sequences (i.e., misaligned nucleotides) present in the content
H	hard clipping (misalignment), clipped sequences (i.e., misaligned nucleotides) NOT present in the content
P	padding (silent deletion from padded reference)

Table 3.2: Operations in the Cigar String (CS) of a short read [8].



Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17			18	19	20	21	22	23	24	25	26	27	28	29	30
Reference	A	G	C	A	T	G	T	T	A	G	A	T	A	A	G	A	T	*	*	A	G	C	T	G	T	G	C	T	A	G	T	A
Content of the SR									a	t	g	T	A	A	*	A	T	G	C	.	.	.	T	A	T	G	C	G	A	G		

Figure 3.2: Content of a short read (SR) and its Cigar String (CS) with respect to the reference genome. The position of the short read corresponds to the first aligned nucleotide in its content and it is 12 in this example. The CS of the short read includes 7 pairs, each indicating an operation from Table 3.2 and the number of nucleotides involved in the corresponding operation. The non-aligned nucleotides (the 3 nucleotides represented with the operation "S" in the CS) are represented in lowercase letters (i.e., a). The dots (at positions $18-20$) and star (at position 15) represent a skipped region and a deletion in the SR, respectively, and they are not present in the actual content.

a chromosome is around 240 million). The cigar string (CS) of a short read expresses the variations in the content of the short read. The CS includes *pairs* of nucleotide lengths and the associated operations. The operations in the CS indicate some properties about the content of the short read such as which nucleotides align with the reference, which are deleted from the reference, and which are insertions that are not in the reference (without revealing the content of the short read).We illustrate descriptions of common operations in the CS in Table 3.2. Finally, the content of a short read includes the nucleotides. In Fig. 3.2, we illustrate how the content of a short read looks and how the CS of the corresponding short read is generated. We note that the actual content only includes nucleotides; the dots (at positions $18-20$) and star (at position 15) in Fig. 3.2 are not present in the content, and they are understood from the CS of the short read. In practice, the position of a short read is in the form $L_{i,j} = \langle x_i | y_j \rangle$, where x_i represents the chromosome ($x_i \in [1, 23]$) and y_j represents the position of the short read on chromosome x_i ($y_j \in [1, 240\text{M}]$). For the clarity of the example in Fig. 3.2, we simplified the representation of the position.

3.3 Overview of the Proposed Solution

In this work, we develop a privacy-preserving system for the storage, retrieval and processing of the SAM files (details are in Section 3.6).

We assume that the sequencing and encryption of the genomes are done at a *certified institution* (CI), which is a trusted entity. Short reads are encrypted after the sequencing,

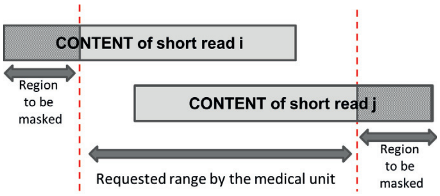


Figure 3.3: Parts to be masked in the short reads for out-of-range content.

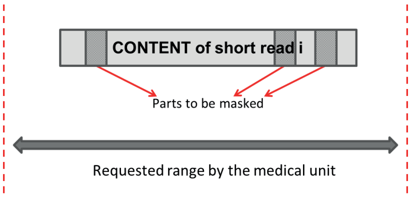


Figure 3.4: Parts to be masked in a short read based on patient's consent. The patient does not give consent to reveal the dark parts of the short read.

and encrypted SAM files of the patients are stored at a biobank (for security, efficiency, and availability). We note that a private company (e.g., cloud storage service) or the government could play the role of the biobank. When a *medical unit* (MU) requests a specific range of nucleotides (on the DNA sequence of one or multiple patients) for a genetic test, the biobank provides all the short reads that include at least one nucleotide from the requested range. We assume that an MU is a broad unit consisting of many subunits (e.g., physicians or specialized clinics) that can potentially request nucleotides from any parts of a patient's genome. To avoid the biobank from associating the conducted genetic tests with the patients, we hide both the real identities of the patients (using pseudonyms) and the types of the conducted tests from the biobank.[2] We hide the types of the conducted tests from the biobank by permuting the positions of the short reads, and then using order-preserving encryption (OPE) on the positions of the short reads. OPE is a deterministic encryption scheme whose encryption function preserves numerical ordering of the plaintexts [21].

As each short read includes between 100 and 400 nucleotides, some short reads that are provided to the MU might include information out of the MU's requested range of genomic data, as in Fig. 3.3. Similarly, some provided short reads might contain privacy-sensitive SNPs of the patient, hence the patient might not give consent to reveal such parts, as in Fig. 3.4. Therefore we mask such parts of the encrypted short reads at the biobank, without decrypting them using an efficient algorithm.

The cryptographic keys of each patient are stored on a *masking and key manager* (MK) by using the patient's pseudonym (hence the participation of the patient is not required in the protocol).[3] The MK can also be embodied by the government or a private company. To avoid the MK from associating the genetic tests with the patients, we do not reveal the identities of the MUs or the patients to the MK.

3.4 Design Constraints and Options

For security, efficiency, and availability, we propose storing the SAM files at a biobank instead of at the MU. Extreme precaution is needed for the storage of genomic data due to its sensitivity. We assume that the biobank is more "security-aware" than an

[2] Knowing the MU (e.g., the name of the hospital) the biobank could de-anonymize an individual using other sources (e.g., by associating the time of the test and the location of the MU with the location patterns of the victim).

[3] Following our discussions with geneticists and medical doctors, we conclude that the patient's involvement in the genetic tests is not desired for the practicality of the protocol (e.g., when a pharmaceutical company conducts genetic research on thousands of patients).

MU, hence it can protect the stored genomic data against a hacker better than an MU (yet, attacks against the biobank cannot be ruled out, as we discuss next). Indeed, this assumption is supported by recent serious medical data breaches from various MUs [7]. Furthermore, by storing the SAM files at one biobank, multiple MUs can reliably access the patients' genomic data from it (instead of each MU individually storing that same large amount of data) at any time.

We propose outsourcing the storage of the cryptographic keys (of the patients) to the MK instead of storing them on a *patient's device* (e.g., a smartphone) due to the following two reasons: (i) It is not realistic to assume that all the patients will take sufficient precautions to protect their cryptographic keys (which will possibly be stored in their smartphones), and (ii) if the keys are stored on a patient's device, operations involving the patient are done on the MU's (e.g., the hospital) computer via the patient's device, hence this approach requires the involvement of the patient in the operation (e.g., physical presence at the hospital). Whereas, following our discussions with geneticists and medical doctors, we conclude that the patient's involvement in the genetic tests is not desired for the practicality of the protocol (e.g., when a pharmaceutical company conducts genetic research on thousands of patients).

In this work, we use OPE instead of private information retrieval (PIR), searchable encryption [183, 117], or oblivious RAM (O-RAM) storage [94] techniques for the privacy-preserving retrieval of the short reads for the following reasons: (i) As we discussed before, the short reads are randomly sampled from the genomes of the patients, and hence the positions of the short reads vary in each patient's genome. The MU typically asks for a particular range of nucleotides on the DNA sequence of one or multiple patients. However, these requested nucleotides reside in different short reads for each patient and the MU does not know which nucleotide is stored in which short reads of each patient (storing the positions of all short reads and the list of nucleotides they accommodate for each patient at every MU requires significant storage overhead). Thus, the MU does not know exactly which short reads to ask for, and hence PIR or searchable encryption techniques would be impractical for our scenario. And (ii) although O-RAM techniques completely hide the data access patterns from the server (biobank), even the most efficient implementations of O-RAM introduce high storage overhead to both the client (MU) and the server (biobank) and introduce about 25 times more overhead with respect to non-oblivious storage [186].

3.5 Threat Model and Security Considerations

We consider the following models for the attacker:

- A curious party at the biobank (or a hacker who breaks into the biobank), who tries (i) to infer the genomic sequence of a patient from his stored genomic data and (ii) to associate the type of the genetic test (e.g., the disease for which the patient is being tested, which can be inferred from the nucleotides requested by the MU) with the patient being tested.
- A curious party at the MK (or a hacker who breaks into the MK), who tries (i) to infer the genomic sequence of a patient from his stored cryptographic keys and the information provided by the biobank and (ii) to associate the type of the genetic test with the patient being tested.
- A curious party at an MU, who can be considered either as an attacker who hacks

into the MU's system or a disgruntled employee who has access to the MU's database. The goal of such an attacker is to obtain the private genomic data of a patient for which it is not authorized.

Apart from (potentially) being curious, we assume that the biobank, the MK, and the MUs are honest organizations. That is, the biobank, the MK, and the MUs honestly follow the protocols and provide correct information to the other parties. In the following, we discuss how we prevent the aforementioned attacks.

SAM files are encrypted (at the CI) and stored at the biobank to prevent the biobank from inferring the genomic data of the patients (details about encryption are in Section 3.6.1). To avoid the biobank from associating the conducted genetic tests with the patients, we hide both the real identities of the patients (using pseudonyms) and the types of the conducted tests (using OPE on the positions of the short reads) from the biobank. Note, however, that the biobank knows the real identity of an MU to make sure that the request comes from a valid source.[4] To avoid the MK from associating the genetic tests with the patients, we do not reveal the identities of the MUs or the patients to the MK. Alternatively (to further increase the security of the scheme), a group signature scheme or anonymous credentials can be integrated for the communication between the MU and the MK. By this way, the MK can also make sure that a request is coming from an authorized MU, without knowing the real identity of the corresponding MU.

Even though we encrypt the positions of the short reads (using OPE) to hide the conducted genetic tests from the biobank, the biobank might still infer the approximate positions of the short reads as a result of using OPE. The biobank does not see the exact bounds of the queries, but it can sort all short reads of the stored genome based on their offsets, which certainly gives it a rough idea which short read contains which nucleotides, and hence which genetic test is being performed. To avoid this, for each patient, we re-define the positions of the short reads before encrypting them using OPE (as discussed in detail in Section 3.6.1).

We also make sure that the MK cannot infer the genomic data of the patients by using the information it receives from the biobank and the cryptographic keys it stores. Indeed, as we will discuss in Section 3.6.2, we only provide the positions and the cigar strings (CSs) of a subset of the short reads (depending on the range of nucleotides requested by the MU) to the MK, which is not enough to infer the nucleotides residing in the contents of corresponding short reads (the contents of the short reads are never transferred to the MK). By only analyzing the CS (without having access to the content), the MK can learn the locations of some insertions and deletions in the patient's genome (but not the contents of these insertions or deletions). However, the MK cannot infer the locations or contents of the patient's privacy-sensitive point mutations (e.g., SNPs), which are typically used to evaluate the predispositions of the patients to various diseases. These privacy-sensitive point mutations can only be inferred when the CS is used together with the content of the short read (which is not revealed to the MK). Furthermore, as we mentioned in Section 3.3, by masking the encrypted short reads before providing them to the MU, we avoid the MU acquiring more genomic data than it requests.

[4] Knowing the MU (e.g., the name of the hospital), the biobank could de-anonymize an individual using other sources (e.g., by associating the time of the test and the location of the MU with the location patterns of the victim). Thus, we hide the types of the conducted tests from the biobank to avoid it associating the conducted genetic test with the individual.

Collusion between the parties (i.e., the biobank, the MK, and an MU) is not allowed in our threat model and we assume that laws could enforce this. Finally, all communication between the parties are encrypted to protect the system from an external attacker.

3.6 Privacy-Preserving Processing of Raw Genomic Data

3.6.1 Cryptographic Keys and Encryption of the Short Reads

We represent the position of a short read ($L_{i,j} = \langle x_i | y_j \rangle$) as a 35-bit number, where the first 5 bits represent the chromosome number (x_i) and the remaining 30 bits represent the position of the short read in the corresponding chromosome (y_j). If the positions of the short reads were encrypted following this representation, the biobank could infer the approximate positions of the short reads as a result of using OPE.

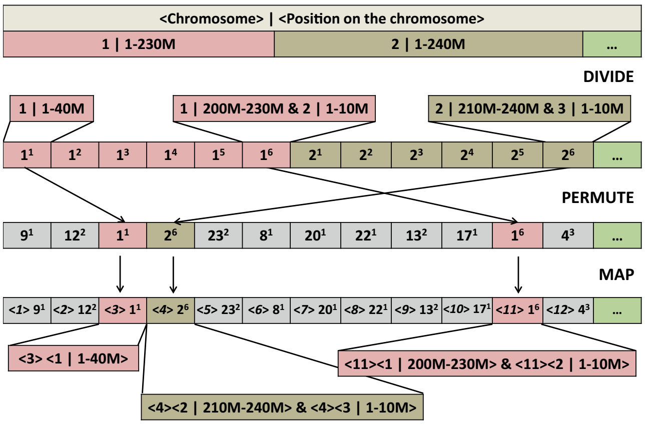
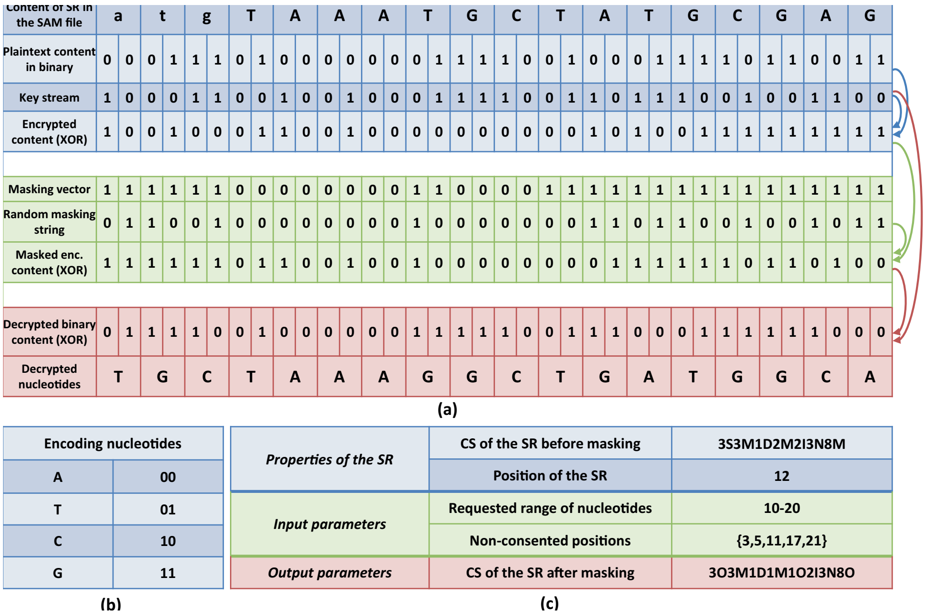


Figure 3.5: Division, permutation and mapping of the positions on the whole genome.

To avoid this, we first divide the positions on the whole genome into parts of equal lengths, permute these parts, and then modify the positions in each part based on the permutation. In Figure 3.5, we show such an example, in which the positions on the genome are divided into parts of length 40 million (totaling 75 parts as there are 3 billion nucleotides in the human genome). For example, chromosome 1 is divided into 6 parts ($1^1, 1^2, \ldots, 1^6$), where the last part includes positions from both the first and second chromosomes. After division, all parts are permuted and mapped to different positions. As a result of the new mapping, the new position of a short read at $L_{i,j} = \langle x_i | y_j \rangle$ becomes $\mathfrak{M}(L_{i,j}) = \langle k \rangle \langle x_i | y_j \rangle$, where $\mathfrak{M}(.)$ is the mapping function for patient P, and k is the mapping of the corresponding part. For example, the position of a short read located in the first part of the first chromosome (part 1^1 in Fig. 3.5) becomes $\mathfrak{M}(L_{i,j}) = \langle 3 \rangle \langle x_i | y_j \rangle$ after the permutation and mapping. Thus, for each patient, we re-define the positions of the short reads based on this new positioning, before encrypting the positions of the short reads using OPE. By doing so, we also change the ordering of the encrypted positions of the short reads. As a consequence, a curious party at the biobank cannot infer which part of the patient's genome is queried by the MU from the stored (encrypted) positions of the short reads. Finally, we assume that the MK keeps the mapping table $\mathfrak{M}_P$ (showing the mapping of each part in each chromosome) for each patient. Note that as the permutation is done differently for each patient, the biobank cannot infer if two different patients are having a similar genetic test.



Content of SR in the SAM file	a		t		g		T		A		A		A		T		G		C		T		A		T		G		C		G		A		G	
Plaintext content in binary	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	1	1	1	1	0	1	1	0	0	1	1
Key stream	1	0	0	0	1	1	0	0	1	0	0	1	0	0	0	1	1	1	1	0	0	1	1	0	1	1	1	0	0	1	0	0	1	1	0	0
Encrypted content (XOR)	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1
Masking vector	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random masking string	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	1	0	0	1	0	0	1	0	1	1
Masked enc. content (XOR)	1	1	1	1	1	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0	1	0	0
Decrypted binary content (XOR)	0	1	1	1	1	0	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
Decrypted nucleotides	T		G		C		T		A		A		A		G		G		C		T		G		A		T		G		G		C		A	

(a)

Encoding nucleotides	
A	00
T	01
C	10
G	11

(b)

Properties of the SR	CS of the SR before masking	3S3M1D2M2I3N8M
	Position of the SR	12
Input parameters	Requested range of nucleotides	10-20
	Non-consented positions	{3,5,11,17,21}
Output parameters	CS of the SR after masking	3O3M1D1M1O2I3N8O

(c)

Figure 3.6: Illustrative example for the encryption, masking and decryption of the content of a short read (SR). (a) Content of the SR (the 2 stars between positions 17 and 21 represent the positions at which the SR has insertions, G and C), its binary representation, the key stream to encrypt the corresponding content, and the format of the encrypted content. Furthermore, following the discussion in Section 3.6.2, we illustrate the masking process considering the range of the requested nucleotides and the patient's consent (in (c)). Finally, we show the format of the decrypted binary content. (b) Encoding format of the nucleotides. (c) Properties of the corresponding short read.

The different parts of each short read are encrypted as follows: (i) The positions of the short reads are encrypted using order-preserving encryption (OPE), (ii) the cigar string (CS) of each short read is encrypted using a semantically secure symmetric encryption function (SE), and (iii) the content of each short read is encrypted using a stream cipher (SC). We note that an SC also provides semantic security, and although we really need an SC for the encryption of the content, one can also use an SC for the encryption of the CS.

We represent the key used for the semantically secure encryption scheme between two parties i and j as $K_{i,j}$. The symmetric OPE key that is used to encrypt the positions of the short reads of patient P is represented as K_P^O. Further, the master key of patient P, which is used to generate the keys of the SC is represented as M_P. We denote $K_P^{C_{i,j}}$ as the SC key used to encrypt the content of the short read whose position is $L_{i,j}$ (where $C_{i,j}$ represents the content of the short read with position $L_{i,j}$). We compute $K_P^{C_{i,j}} = \mathrm{H}(M_P, \mathcal{F}(L_{i,j}, S_{i,j}), L_{i,j})$, where $L_{i,j}$ is the (starting) position of the corresponding short read (on the DNA sequence), $S_{i,j}$ is a random salt to provide different keys for the short reads with the same positions, and H is a pseudorandom function. Moreover, $\mathcal{F}(L_{i,j}, S_{i,j})$ is a function that generates a *nonce* from the position and the random salt of the corresponding short read. We represent the public-key encryption of message m under the public key of i as $\mathcal{E}(\mathcal{K}_i, m)$, the encryption of message

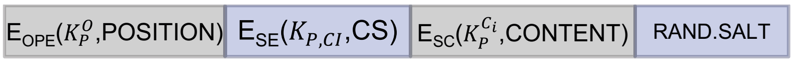

$\text{E}_{\text{OPE}}(K_P^O$,POSITION)
$\text{E}_{\text{SE}}(K_{P,CI}$,CS)
$\text{E}_{\text{SC}}(K_P^{C_i}$,CONTENT)
RAND.SALT

Figure 3.7: Format of an encrypted short read. The size of each field is discussed in Section 3.8.

m via a semantically secure symmetric encryption function (SE) using the symmetric key between i and j as $\text{E}_{\text{SE}}(K_{i,j}, m)$, and the OPE of message m using the OPE key of P as $\text{E}_{\text{OPE}}(K_P^O, m)$. Furthermore, we represent the SC encryption of the content of a short read as $\text{E}_{\text{SC}}(K_P^{C_{i,j}}, C_{i,j})$, where $C_{i,j}$ represents the content of the short read at $L_{i,j}$. In Fig. 3.6(a), we illustrate how the content of a short read is translated to plaintext bits and encrypted using SC (by XOR-ing the content with the key stream). Finally, in Fig. 3.7, we illustrate the format of an encrypted short read.

We assume that the certified institution (CI), where the patient's DNA is sequenced and analyzed, has K_P^O, M_P, and $K_{P,CI}$ ($K_{P,CI}$ is used to encrypt the CSs of the short reads) for the initial encryption of the patient's genomic data. These keys are then deleted from the CI after the sequencing, alignment, and encryption. We also assume that for each patient P, the MK stores K_P^O, M_P, and $K_{P,CI}$ along with the mapping table $\mathfrak{M}_P$ (as discussed before). Finally, the MU only stores the public key of the MK, $\mathcal{K}_{MK}$.

3.6.2 Proposed Protocol

Typically, a specialist at the MU (e.g., a physician at the hospital or a specialized clinic connected to the hospital) requests a range of nucleotides (on the DNA sequence of one or more patients) from the biobank (either for a personal genetic test or for clinical research). For simplicity of the presentation, we assume that the request is for a specific range of nucleotides of patient P. We illustrate the connections between the parties that are involved in the protocol in Fig. 3.8(a). In the following, we describe the steps of the proposed protocol (these steps are also illustrated in Fig. 3.8(b)).

- **Step 1:** The patient (P) provides a sample (e.g., his saliva) along with his permission to the certified institution (CI) for sequencing.
- **Step 2:** The CI does the sequencing and constructs the SAM file of the patient. The short reads of the patient are also encrypted at the CI (as discussed in Section 3.6.1).
- **Step 3:** The CI sends the encrypted SAM file to the biobank along with the corresponding pseudonym of the patient. The CI also sends K_P^O, M_P, $K_{P,CI}$, and the mapping table $\mathfrak{M}_P$ for patient P directly to the MK via a secure channel (we do not illustrate this step in Fig. 3.8). We note that the first 3 steps of the protocol are executed only once.
- **Step 4:** A specialized sub-unit at the MU requests nucleotides from the range $[R_L, R_U]$ (R_L being the lower bound and R_U being the upper bound of the requested range) on the DNA sequence of patient P for a genetic test. We note that an access control unit stores the authorizations (i.e., access rights) of the original request owners (e.g., specialist at a hospital) to different parts of the genomic data. In our setting, the MU checks the access rights of the original request owner before forwarding the request to the biobank. Once, the MU verifies that the original request owner has the sufficient access rights to the requested range of nucleotides, the MU generates a one-time session key $K_{MK,MU}$, which will be used for the secure communication between the MU and the MK. The MU encrypts this session key with the public key of the MK to obtain $\mathcal{E}(\mathcal{K}_{MK}, K_{MK,MU})$.

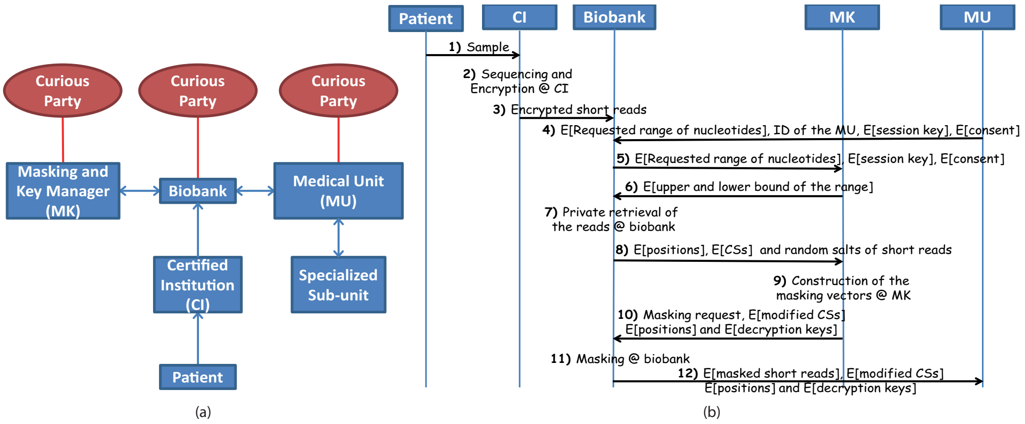

Curious Party
Curious Party
Curious Party
Masking and Key Manager (MK)
Biobank
Medical Unit (MU)
Certified Institution (CI)
Specialized Sub-unit
Patient
(a)
Patient
CI
Biobank
MK
MU
1) Sample
2) Sequencing and Encryption @ CI
3) Encrypted short reads
4) E[Requested range of nucleotides], ID of the MU, E[session key], E[consent]
5) E[Requested range of nucleotides], E[session key], E[consent]
6) E[upper and lower bound of the range]
7) Private retrieval of the reads @ biobank
8) E[positions], E[CSs] and random salts of short reads
9) Construction of the masking vectors @ MK
10) Masking request, E[modified CSs] E[positions] and E[decryption keys]
11) Masking @ biobank
12) E[masked short reads], E[modified CSs] E[positions] and E[decryption keys]
(b)

Figure 3.8: (a) Connections between the parties in the proposed protocol. (b) The operations and message exchanges in the proposed protocol.

The MU encrypts the lower and upper bounds of the requested range with $K_{MK,MU}$ to obtain $\mathrm{E_{SE}}(K_{MK,MU}, R_L||R_U)$ and sends the corresponding request to the biobank along with the pseudonym of the patient P, the identification of the MU[5], $\mathcal{E}(\mathcal{K}_{MK}, K_{MK,MU})$, and $\mathrm{E_{SE}}(K_{MK,MU}, \Omega_P)$, where Ω_P is the pseudonymized consent of the patient.[6] The MK uses this pseudonymized consent Ω_P to generate the masking vectors (as in Step 9).

• **Step 5:** Once the biobank verifies that request comes from a valid source[7], it forwards $\mathrm{E_{SE}}(K_{MK,MU}, R_L||R_U)$, and $\mathrm{E_{SE}}(K_{MK,MU}, \Omega_P)$, along with the pseudonym of the patient, and the encrypted session key $\mathcal{E}(\mathcal{K}_{MK}, K_{MK,MU})$ to the MK.

• **Step 6:** The MK decrypts the session key to obtain $K_{MK,MU}$ and decrypts the request $(\mathrm{E_{SE}}(K_{MK,MU}, R_L||R_U))$ to obtain R_L and R_U. As we discussed before, the position of a short read is the position of the first aligned nucleotide in its content. Let Γ be the maximum number of nucleotides in a short read. Then, the short reads with position in $[R_L - \Gamma, R_L - 1]$ might also include nucleotides from the requested range ($[R_L, R_U]$) in their contents. Thus, the MK re-defines the lower bound of the request as $R_L - \Gamma$ in order to make sure that all the short reads (which include at least one nucleotide from the requested range of nucleotides) are retrieved by the biobank.

Next, the MK determines where $(R_L - \Gamma)$ and R_U are mapped to following the mapping table $\mathfrak{M}_P$ of patient P (as discussed in Section 3.6.1). If both $(R_L - \Gamma)$ and R_U are on the same part (e.g., in Fig. 3.5), then the MK computes the range of short read positions (to be retrieved by the biobank) as $[\mathfrak{M}(R_L - \Gamma), \mathfrak{M}(R_U)]$, where $\mathfrak{M}(.)$ is the mapping function for patient P. Otherwise (if they are not on the same part), due to the permutation of the parts, the MK generates multiple ranges of short read positions to make sure all short reads including at least one nucleotide from $[R_L, R_U]$ are retrieved by the biobank. For simplicity of the presentation, we assume $(R_L - \Gamma)$ and R_U are on the same part. Finally, the MK computes the encrypted range $[\mathrm{E_{OPE}}(K_P^O, \mathfrak{M}(R_L - \Gamma)), \mathrm{E_{OPE}}(K_P^O, \mathfrak{M}(R_U))]$, and sends this encrypted range to the biobank (with pseudonym of P).

[5] We reveal the real identity of the MU to the biobank to make sure that the request comes from a valid source.

[6] Ω_P denotes the positions on the patient's genome for which the patient does not give consent to the original request owner (e.g., specialized sub-unit at the MU).

[7] We assume that the biobank has a list of valid MUs, whose requests it will answer.

- **Step 7:** The biobank retrieves all the short reads (in the SAM file of patient P) whose encrypted positions ($\mathrm{E}_{\mathrm{OPE}}(K_P^O, \mathfrak{M}(L_{i,j}))$) are in the set $\Delta = \{\mathrm{E}_{\mathrm{OPE}}(K_P^O, \mathfrak{M}(L_{i,j})) : \mathrm{E}_{\mathrm{OPE}}(K_P^O, \mathfrak{M}(R_L - \Gamma)) \leq \mathrm{E}_{\mathrm{OPE}}(K_P^O, \mathfrak{M}(L_{i,j})) \leq \mathrm{E}_{\mathrm{OPE}}(K_P^O, \mathfrak{M}(R_U))\}$. As OPE preserves the numerical ordering of the plaintext positions, the biobank constructs the set Δ without accessing the plaintext positions of the short reads.
- **Step 8:** The biobank provides Δ along with the corresponding encrypted CSs and the random salt values of the short reads to the MK.
- **Step 9:** The MK decrypts the corresponding positions and the CSs of the retrieved short reads by using K_P^O and $K_{P,CI}$ in order to construct the masking vectors for the biobank. These masking vectors prevent the leakage of out-of-range content (in Fig. 3.3) and non-consented nucleotides (in Fig. 3.4) to the MU, as we discussed in Section 3.3. We note that from the positions and the CSs of the short reads, the MK cannot infer the locations or contents of the patient's privacy-sensitive point mutations (e.g., SNPs), which are typically used to evaluate the predispositions of the patients for various diseases. These privacy-sensitive point mutations can only be inferred when the CS is used together with the content of the short read (which is not revealed to the MK).

The MK can determine the actual position of a short read from its mapped position as the MK has the mapping table $\mathfrak{M}_P$ for patient P (i.e., it can infer $L_{i,j}$ from $\mathfrak{M}(L_{i,j})$ using $\mathfrak{M}_P$). Using the position and the CS of a short read, the MK can determine the exact positions of the nucleotides in the content of a short read (but not the contents of the nucleotides, because the contents are encrypted and stored at the biobank). Using this information, the MK can determine the parts in the content of the short read that are out of the requested range $[R_L, R_U]$. Furthermore, the MK can also determine whether the short read includes any nucleotide positions for which the patient P does not give consent. Therefore, the MK constructs binary masking vectors indicating the positions in the contents of the short reads that are needed to be masked by the biobank before sending the retrieved short reads to the MU.

Let Π_P be the set of nucleotide positions (on the DNA sequence) for which the patient P does not give consent (e.g., set of positions including privacy-sensitive SNPs of the patient). Then, the set $\Sigma = [R_L, R_U] \setminus \Pi_P$ includes the positions of the nucleotides that can be provided to the MU without masking. The masking vector for a short read (with position $L_{i,j}$) is constructed following Algorithm 1. In Figure 3.6(a), we illustrate how the masking vector is constructed for the corresponding short read, when the requested range of nucleotides is $[10, 20]$ and for a given set of nucleotide positions (on the DNA sequence) for which the patient P does not give consent (as in Fig. 3.6(c)).

The MK also modifies the CS of each short read (if it is marked for masking) according to the nucleotides to be masked. That is, the MK modifies the CS such that the masked nucleotides are represented with a new operation "*O*" in the CS. By doing so, when the MU receives the short reads, it can see which parts of them are masked. In Figure 3.6(c), we illustrate how the CS of the corresponding short read changes as a result of the masking vector in Figure 3.6(a). Then, the MK generates the decryption keys for each short read (whose position is in Δ) by using the master key of the patient (M_P), positions of the shorts read, and the random salt values.[8]

- **Step 10:** The MK encrypts the positions, the (modified) CSs, and the generated decryption keys of the contents of the short reads, using $K_{MK,MU}$. Then, it sends the

[8] The generation of the decryption keys for the SC is the same as the generation of the encryption keys as we discussed in Section 3.6.1.

Algorithm 1: Construct the masking vector V_m for short read with position $L_{i,j} = \langle x_i | y_j \rangle$

```
Input  : $L_{i,j} = \langle x_i | y_j \rangle$, CS of the short read at $L_{i,j}$, positions of authorized nucleotides ($\Sigma$)
Output: $V_m$ // Each nucleotide is represented by 2-bits, initially all bits are set to 0
$N_p \leftarrow$ # pairs in the CS of the short read
$P_0 \leftarrow y_j$ // Assign the position of the short read on chromosome $x_i$ to $P_0$
$I \leftarrow 0$ // Index of the nucleotides in the content of the short read
for $i \leftarrow 1$ to $N_p$ do
    Get the $i^{th}$ pair of the CS with the fields $n_i$ and $\ell_i$
    $\ell_i \leftarrow$ Operation noted in the $i^{th}$ pair of the CS (from Table 3.2)
    $n_i \leftarrow$ # nucleotides following the operation noted in $\ell_i$
    if $\ell_i = H \vee \ell_i = P$ then
        do nothing
    else if $\ell_i = S$ then
        for $j \leftarrow 0$ to $(n_i - 1)$ do
            $V_m(1, 2(I+j)) \leftarrow 1$, $V_m(1, 2(I+j)+1) \leftarrow 1$ // Mark the $(I+j)^{th}$ nucleotide in the content of the short read for masking
        end
        $I \leftarrow I + n_i$
    else if $\ell_i = M$ then
        for $j \leftarrow 0$ to $(n_i - 1)$ do
            if $(P_0) \notin \Sigma$ then
                $V_m(1, 2(I+j)) \leftarrow 1$, $V_m(1, 2(I+j)+1) \leftarrow 1$
            end
            $P_0 \leftarrow P_0 + 1$
        end
        $I \leftarrow I + n_i$
    else if $\ell_i = I$ then
        if $(P_0) \notin \Sigma$ then
            for $j \leftarrow 0$ to $(n_i - 1)$ do
                $V_m(1, 2(I+j)) \leftarrow 1$, $V_m(1, 2(I+j)+1) \leftarrow 1$
            end
        end
        $I \leftarrow I + n_i$
    else if $\ell_i = D \vee \ell_i = N$ then
        $P_0 \leftarrow P_0 + n_i$
end
```

masking vectors along with the encrypted positions, CSs and decryption keys to the biobank. We note that in this step, the MK encrypts the actual positions of the short reads (e.g., $L_{i,j}$ instead of $\mathfrak{M}(L_{i,j})$) as these positions will be eventually decrypted and used by the MU, and the MU does not need to know the mapping table $\mathfrak{M}_P$ of the patient.

- **Step 11:** The biobank conducts the masking by XOR-ing the bits of the encrypted content of each short read (whose position is in Δ) with a random masking string. Each entry (bit) of the random masking string is assigned as follows: (i) If the corresponding entry is set for masking in the masking vector, it is assigned with a random binary value, and (ii) it is assigned with zero, otherwise. We describe this process in Algorithm 2. Furthermore, in Figure 3.6(a), we illustrate how the masked encrypted content for the corresponding short read is constructed by XOR-ing the random masking string with the encrypted content.
- **Step 12:** Finally, the biobank sends the encrypted positions, CSs and decryption keys (generated in Step 10 by the MK) along with the masked contents (generated in Step 11 by the biobank) to the MU. The MU decrypts the received data and obtains the

Algorithm 2: Construct the random masking string V_s and conduct the masking for short read with position $L_{i,j} = \langle x_i | y_j \rangle$

```
Input : $V_m$ // Masking vector for the short read with position $L_{i,j}$
        $\mathrm{E}_{\mathrm{SC}}(K_P^{C_{i,j}}, C_{i,j})$ // Encrypted content with (encrypted) position
        $\mathrm{E}_{\mathrm{OPE}}(K_P^{O}, \mathfrak{M}(L_{i,j}))$ in $\Delta$
Output: $\mathcal{M}\{\mathrm{E}_{\mathrm{SC}}(K_P^{C_{i,j}}, C_{i,j})\}$ // The masked content
$V_s \leftarrow$ zeros$(1, size(V_m, 2))$
for $i \leftarrow 1$ to $size(V_m, 2)$ do
    if $V_m(i) = 1$ then
        $V_s(i) \leftarrow$ Rand // Rand generates a random number from $\{0, 1\}$
    end
end
$\mathcal{M}\{\mathrm{E}_{\mathrm{SC}}(K_P^{C_{i,j}}, C_{i,j})\} \leftarrow \mathrm{E}_{\mathrm{SC}}(K_P^{C_{i,j}}, C_{i,j}) \oplus V_s$
```

requested nucleotides of the patient.

3.7 Evaluation

Focusing on the leakage of genomic data, we evaluate the proposed privacy-preserving system by using real genomic data to show (i) how the leakage of genomic data from the short reads threatens the genomic privacy of a patient, and (ii) how the proposed masking technique helps to prevent this leakage. We assume that the MU requests a specific range of nucleotides of patient P (e.g., for a genetic test) from the biobank. In practice, the requested range can include from one to thousands of nucleotides depending on the type of the genetic test.

First, without the masking in place, we observe the ratio of unauthorized genomic data (i.e., number of nucleotides provided to the MU that are out of the requested range) to the authorized data (i.e., number of nucleotides within the requested range) for various request sizes. For simplicity, we assume that all the nucleotides within the requested range are considered as consented data (i.e., the situation in Figure 3.4 is not considered); and only those that are out of the requested range (but still provided to the MU via the short reads) are considered as the unauthorized data. For the patient's DNA profile (i.e., SAM file), we use a real human DNA profile [6] (with an average coverage of 8, meaning each nucleotide is present, on the average, in 8 short reads in the SAM file, and each short read includes at most 100 nucleotides) and we randomly choose the ranges of requested nucleotides from the entire genome of the patient. We illustrate our results in Fig. 3.9. We observe that for small request sizes, the amount of leakage (of unauthorized data) is very high compared to the size of authorized data. As the leakage vanishes (e.g., the ratio in Figure 3.9 becomes 0) with the proposed masking technique, we do not show the leakage when the proposed masking technique is in place in Figures 3.9 and 3.12.

Using the same DNA profile, we also observe the evolution in the amount of leaked genomic data over time. For simplicity of the presentation, we assume slotted time and that the MU conducts a genetic test on the patient at each time slot (by requesting a particular range of nucleotides from a random part of his genome). In Figure 3.10, we illustrate the amount of genomic data (i.e., number of nucleotides) that is leaked to the MU in 100 time-slots. The jumps in the number of leaked nucleotides (at some time-slots) is due to the fact that some requests might retrieve more short reads comprised of

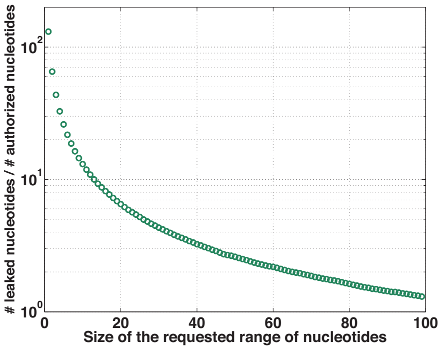


Figure 3.9: Ratio of unauthorized genomic data to the authorized data vs. the size of the requested range of nucleotides, when there is no masking in place.

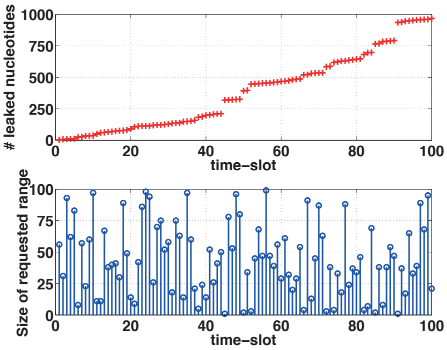


Figure 3.10: Number of leaked nucleotides vs. time for various request sizes, when there is no masking in place.

more out-of-range nucleotides. As before, leakage becomes 0 when masking is in place, which shows the crucial role of the proposed scheme.

We also study the information leakage, focusing on the leaked single nucleotide polymorphisms (SNPs) of the patient as a result of different sizes of requests (from random parts of the patient's genome). In Figure 3.11, we illustrate the number of SNPs leaked to the MU in 100 time-slots. We observe that the number of leaked SNPs is more than twice the number of authorized SNPs (which are within the requested range of nucleotides). When the proposed masking technique is in place, the number of leaked SNPs (outside the requested range) becomes 0 in Figure 3.11.

Finally, we study the genomic data leakage (number of leaked nucleotides and SNPs) when the MU tests the susceptibility of the patient [6] to a particular disease (i.e., when the MU asks for the set of SNPs of the patient that are used to test the corresponding disease). For this study, we use real disease markers [9]. We note that for this type of test, the size of the requested range of nucleotides (by the MU) for a single SNP is typically 1, but the SNPs are from several parts of the patient's genome. In Figure 3.12, we illustrate the genomic data leakage of the patient as a result of various disease susceptibility tests each requiring a different number of SNPs from different parts of the patient's genome (on the x-axis we illustrate the number of SNPs required for each test). We again observe that the leaked SNPs, as a result of different disease susceptibility tests, reveal privacy-sensitive data about the patient. For example, leaked SNPs of the patient as a result of a test for the Alzheimer's disease could leak information about the patient's susceptibility to "smoking behavior" or "diabetes".

In Table 3.3, we list a small subset of the leaked SNPs, along with their clinical nature, as a result of the disease susceptibility tests in Figure 3.12.[9] For the patient's genomics data (i.e., SAM file), we used a real DNA profile [6] including around 300 million short reads with a coverage of 10. We also use real disease markers [9].

It is worth noting that the SNPs in Table 3.3 are not the ones that are used to test the patient's susceptibility for the corresponding disease; they are the leaked SNPs due to the corresponding genetic test (when there is no masking in place). For example, in Table 3.3, the SNP with ID "rs6265" is not required to check the patient's susceptibility

[9] We used the Ensembl database (http://www.ensembl.org/info/docs/variation/index.html) to determine the clinical nature of the leaked SNPs.

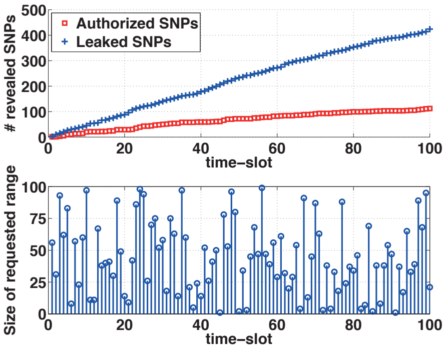


Figure 3.11: Number of leaked SNPs vs. time for various request sizes, when there is no masking in place.

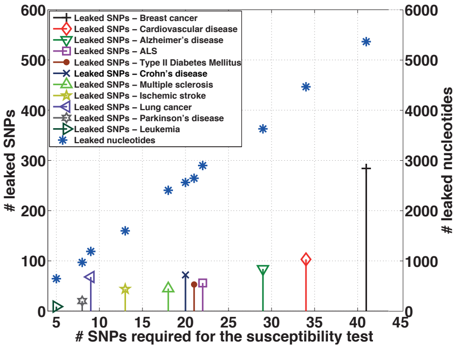


Figure 3.12: Number of leaked SNPs and nucleotides during the susceptibility test to different diseases when there is no masking in place. The values on the right y-axis correspond to the number of leaked nucleotides.

to the Alzheimer's disease. However, it is leaked to the MU as one of the short reads of the patient that include a marker for Alzheimer's, also includes "rs6265" (as each short read includes around 100 nucleotides, a short read could include more than one SNP).

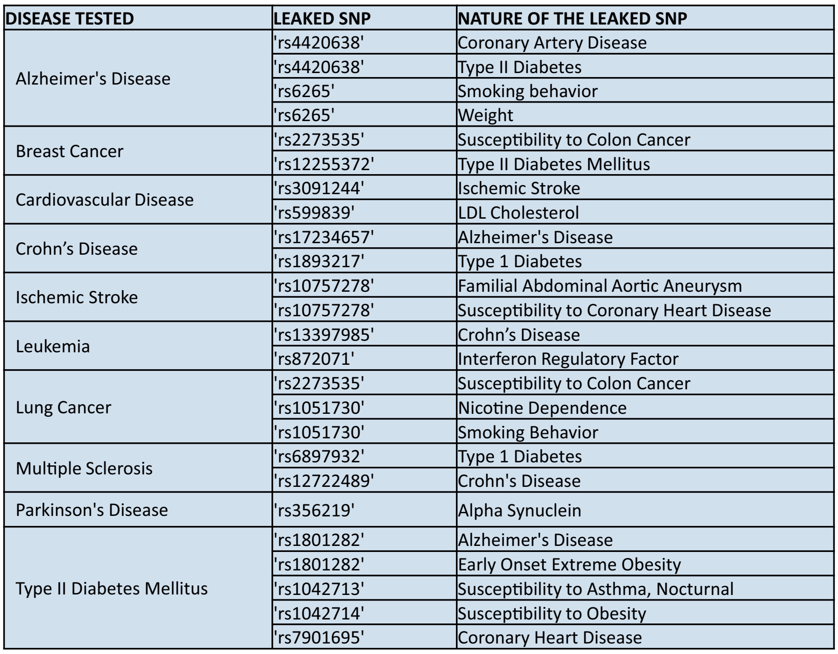
We observe that as a result of this leakage, the patient's (i) susceptibility to certain diseases, and (ii) physical attributes (e.g., body mass index, susceptibility to be overweight, etc.) are revealed. Furthermore, a SNP might reveal more than one attributes (e.g., "rs6265" in Table 3.3). We emphasize that leakage of SNPs (listed in Table 3.3) is avoided when the proposed masking technique (described in Section 3.6.2) is in place (i.e., similar to the previous cases, the number of leaked nucleotides and SNPs is 0 when masking is in place).

3.8 Implementation and Complexity Analysis

We implemented the proposed system and assessed its storage requirement and complexity on an Intel Core i7-2620M CPU with a 2.70 GHz processor under Windows 7, using Java. As before, for the patient's SAM file, we used a real DNA profile [6] including around 300 million short reads (each short read including at most 100 nucleotides).

We used the Salsa20 stream cipher[43] and the implementation of OPE from [156]. We also used CCM mode of AES (with key size of 256-bits) for the secure communication between the MK and the MU, and RSA (with key size of 2048-bits) for the public-key encryption.

We structured the fields in the encrypted short read (in Fig. 3.7) as follows: We reserved the first 8-bytes for the encrypted position of the short read (via OPE). To save storage, we devoted the next 64-bytes of the encrypted short read to the CS and the content of the short read. As the input size of the stream cipher is 64-bytes, we encrypted the CS together with the content and other (header) information of the short read using the stream cipher. That is, out of the 64-byte input of the stream cipher, we allocated the first 20-bytes for the CS, the next 25-bytes for the content (as each short read in the used DNA profile includes at most 100 nucleotides), and the remaining 19-bytes for the



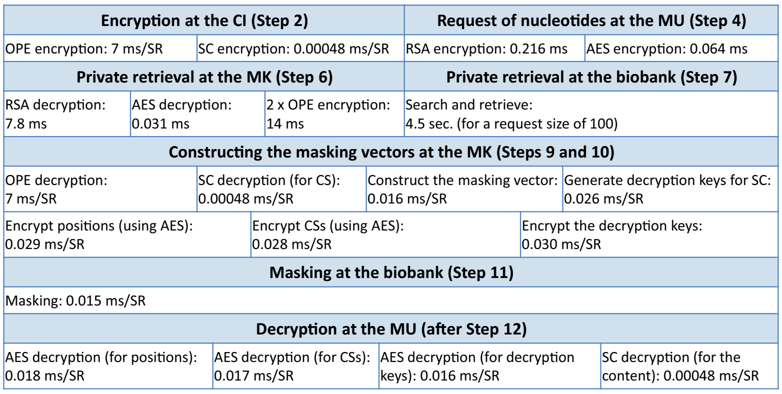
DISEASE TESTED	LEAKED SNP	NATURE OF THE LEAKED SNP
Alzheimer's Disease	'rs4420638'	Coronary Artery Disease
	'rs4420638'	Type II Diabetes
	'rs6265'	Smoking behavior
	'rs6265'	Weight
Breast Cancer	'rs2273535'	Susceptibility to Colon Cancer
	'rs12255372'	Type II Diabetes Mellitus
Cardiovascular Disease	'rs3091244'	Ischemic Stroke
	'rs599839'	LDL Cholesterol
Crohn’s Disease	'rs17234657'	Alzheimer's Disease
	'rs1893217'	Type 1 Diabetes
Ischemic Stroke	'rs10757278'	Familial Abdominal Aortic Aneurysm
	'rs10757278'	Susceptibility to Coronary Heart Disease
Leukemia	'rs13397985'	Crohn’s Disease
	'rs872071'	Interferon Regulatory Factor
Lung Cancer	'rs2273535'	Susceptibility to Colon Cancer
	'rs1051730'	Nicotine Dependence
	'rs1051730'	Smoking Behavior
Multiple Sclerosis	'rs6897932'	Type 1 Diabetes
	'rs12722489'	Crohn's Disease
Parkinson's Disease	'rs356219'	Alpha Synuclein
Type II Diabetes Mellitus	'rs1801282'	Alzheimer's Disease
	'rs1801282'	Early Onset Extreme Obesity
	'rs1042713'	Susceptibility to Asthma, Nocturnal
	'rs1042714'	Susceptibility to Obesity
	'rs7901695'	Coronary Heart Disease

Table 3.3: Nature of the leaked SNPs as a result of various genetic tests for different diseases.

remaining information about the short read (or padding). Finally, the last byte of the short read includes the plaintext random salt. Consequently, we computed the storage cost as 21.6 GB per patient. We note that stream cipher encryption does not increase the size of the data as it is the XOR of the key stream with the plaintext. The storage overhead (due to the proposed privacy-preserving scheme) is due to the encryption of the positions of the short reads by using OPE. A plaintext position is around 40 bits (depending the number of parts in Fig. 3.5) and an encrypted position is 8-bytes using the implementation of OPE in [156].

We also evaluated the computation times for different steps of the proposed scheme (following the operations in Figure 3.8(b)) in Table 3.4. As shown in Table 3.4, the computation time of the whole process is dominated by the retrieval of the reads at the biobank. However, we observe that this operation can be parallelized. We note that encryption of the SAM file at the CI (Step 2) is a one-time operation and the encryption time is dominated by the execution of OPE. We used the implementation in [156] for the OPE. However, the OPE encryption and decryption are shown to be about 80 times faster using the more recent and secure version of the OPE in [155].

Overall, it takes approximately 5 seconds for the MU to receive the requested range of nucleotides of the patient (Steps 4-12) after privacy-preserving retrieval and masking (for a range size of 100, which includes on the average 23 short reads), which shows the efficiency and practicality of the proposed scheme. We note that the computation time of the whole process is dominated by the retrieval of the reads at the biobank (which does not involve any cryptographic operations). Therefore, we can easily claim that the cost of cryptographic operations is not a bottleneck for the proposed protocol.



<table>
<tr><th colspan="6">Encryption at the CI (Step 2)</th><th colspan="6">Request of nucleotides at the MU (Step 4)</th></tr>
<tr><td colspan="3">OPE encryption: 7 ms/SR</td><td colspan="3">SC encryption: 0.00048 ms/SR</td><td colspan="3">RSA encryption: 0.216 ms</td><td colspan="3">AES encryption: 0.064 ms</td></tr>
<tr><th colspan="6">Private retrieval at the MK (Step 6)</th><th colspan="6">Private retrieval at the biobank (Step 7)</th></tr>
<tr><td colspan="2">RSA decryption:
7.8 ms</td><td colspan="2">AES decryption:
0.031 ms</td><td colspan="2">2 x OPE encryption:
14 ms</td><td colspan="6">Search and retrieve:
4.5 sec. (for a request size of 100)</td></tr>
<tr><th colspan="12">Constructing the masking vectors at the MK (Steps 9 and 10)</th></tr>
<tr><td colspan="3">OPE decryption:
7 ms/SR</td><td colspan="3">SC decryption (for CS):
0.00048 ms/SR</td><td colspan="3">Construct the masking vector:
0.016 ms/SR</td><td colspan="3">Generate decryption keys for SC:
0.026 ms/SR</td></tr>
<tr><td colspan="4">Encrypt positions (using AES):
0.029 ms/SR</td><td colspan="4">Encrypt CSs (using AES):
0.028 ms/SR</td><td colspan="4">Encrypt the decryption keys:
0.030 ms/SR</td></tr>
<tr><th colspan="12">Masking at the biobank (Step 11)</th></tr>
<tr><td colspan="12">Masking: 0.015 ms/SR</td></tr>
<tr><th colspan="12">Decryption at the MU (after Step 12)</th></tr>
<tr><td colspan="3">AES decryption (for positions):
0.018 ms/SR</td><td colspan="3">AES decryption (for CSs):
0.017 ms/SR</td><td colspan="3">AES decryption (for decryption keys): 0.016 ms/SR</td><td colspan="3">SC decryption (for the content): 0.00048 ms/SR</td></tr>
</table>

Table 3.4: Computation times at different steps of the proposed scheme (following the steps in Fig. 3.8(b)), where SR stands for the short read.

3.9 Summary

In this chapter, we have introduced the first privacy-preserving system for the storage, retrieval, and processing of aligned, raw genomic data (i.e., SAM files). The proposed scheme stores the SAM files of the patients at a biobank and lets the medical units (hospitals or pharmaceutical companies) privately retrieve the data (they are authorized for) from the biobank for genetic tests. We have shown that the proposed scheme efficiently prevents the leakage of genomic data and preserves the genomic privacy of the patients. We are confident that the proposed scheme will accelerate genomic research, because clinical trial participants will be more willing to consent to the sequencing of their genomes if they are ensured that their genomic privacy is preserved.

3.10 Acknowledgements

We would like to thank Jurgi Camblong, Pierre Hutter, Zhenyu Xu, Wolfgang Huber, and Lars Steinmetz for their useful comments.

Chapter 4

Privacy-Preserving Genomic Testing in the Clinic: A Model Using HIV Treatment

In the previous chapter, we have seen how to securely store and process raw genomic data stored in SAM files that are required for in-depth bioinformatics analyses and clinical trials. Yet, the implementation of routine genomic-based medicine is mostly based on variant call formats (VCF) that store only the information related to the genetic variation. In this chapter, we address unresolved questions regarding the privacy-preserving storage and process of genetic variants and the delivery of interpreted test results to health-care practitioners. We used DNA-based prediction of HIV-related outcomes as a use case for our model and we report its successful implementation in the context of the Swiss HIV Cohort Study.

4.1 Introduction

The clinical use of genomic data has the potential to improve healthcare by allowing for more individualized preventive and therapeutic strategies. However, such use raises critical issues regarding the protection of the data, its predictive power, the interpretation and delivery of results.

Currently, the majority of clinical genetic testing consists of targeted genotyping of one or a few markers. However, it is likely that future testing will involve the in silico selection of relevant markers from a large set of previously genotyped variants (e.g. by whole genome sequencing). Large-scale genetic data will thus be stored and analyzed routinely in a clinical context, still they have specificities that differentiate them from the rest of health-related information: genomic data have the potential to inform on identity, ancestry and risks of multiple diseases in a given patient and their relatives [191]. Additionally, many of the approaches used in research (e.g. anonymization, de-identification) are not applicable to genetic information, as the genome is the ultimate identifier for each individual. Thus, there is a requirement for additional strategies that preserve the privacy of genomic data while not compromising the accuracy of results.

Clinical genetic tests vary in number of informative markers and overall predictive power. Some tests are deterministic (or nearly deterministic), and thus are associated with a clear interpretation (e.g., HLA-B*57 and severe hypersensitivity reaction to abacavir [136, 166]). However, other variants are largely non-deterministic (e.g., multiple variants moderately impact risk of metabolic disorders) and are best summarized by genetic risk scores, and reported as modifying an individual's basal risk. Thus, a real-world framework for genomic testing needs to provide a calculation and reporting infrastructure that incorporates both classes of results.

Another roadblock to implementing genomic-based medicine is the challenge of transmitting clinically useful information to healthcare practitioners. Most clinicians lack both the time and the specialized knowledge that are required for an expert interpretation of genotyping results. Reports of genetic risk should, therefore, be formatted similarly to other common laboratory tests results and include only actionable, interpreted results.

In this study, we have chosen clinical aspects of HIV care as a model setting for an implementation of privacy-preserving genetic testing and results reporting in the clinic. Human genetic variation impacts multiple aspects of HIV disease including rate of disease progression off therapy (recently reviewed in [140]), response to therapy [39] and adverse events [131] and susceptibility to metabolic disorders [97, 171, 172]. Today, decisions for clinical care of HIV are based on guidelines, local preferences, clinical and demographic data, viral resistance analyses, and (increasingly) cost [98]. The fact that there are now multiple alternatives for first and second line treatments sets the stage for more informed treatment decisions.

We surveyed 55 physicians of the Swiss HIV Cohort Study [10] who used our privacy-preserving model for genetic testing. We evaluated their feedback on three different aspects: clinical utility, ability to address privacy concerns and system usability. The purpose was to identify the key aspects of our model that could represent general drivers for a faster adoption of privacy-enhancing solutions in clinical genomics.

The main contribution of this chapter can be summarized as follows:

- We develop the first system for privacy-preserving genetic testing with ancestry inference and delivery of interpreted results to health-care practitioners.

- We deploy our framework in the context of the Swiss HIV Cohort Study and survey physicians that used it with 230 HIV-positive individuals genotyped at 4,149 markers.

- We evaluate physicians feedback and discuss key insights that will improve the design of privacy-preserving systems for personalized medicine in the future.

We summarize the notation used in this Chapter in Table 4.1.

4.2 Preliminaries

In this section, we summarize the main concepts in genomics and cryptography that we use throughout the chapter, as a complement to those we have seen in Chapther 2.

Notation	**Description**
VCF	Variant call format
OR	Odds ratio
PCA	Principal component analysis
SNP	Single nucleotide polymorphism
HBC	Honest-but-curious
DTE	Deterministic encryption
$\mathbf{A}_i$	Set of ancestry information of participant i
$\mathbf{S}_i$	Set of SNPs genotypes of participant i
$\mathbf{N_i}$	Set of values for clinical and environmental factors of participant i
$\mathbf{PC_i}$	Set of principal components of participant i
$\mathbf{W}$	Set of SNP weights obtained as a result of the PCA
$\mathbf{C}$	Set of cluster means
$\mathbb{G}(X)$	Genetic risk score
$\mathbb{R}(X)$	Overall risk score (including clinical and environmental factors)
β_j	Regression coefficient where $OR_j = exp(\beta_j)$ for the j-th covariate
$\mathrm{p}^i_j(X)$	Contribution of the j-th value (or genotype) of the i-th variant to the genetic risk
α	Baseline risk
$[m]$	Paillier encryption of message m
$\langle m \rangle$	Paillier partial decryption fo message m
$[\![m]\!]$	DGK encryption of message m
K_i	Paillier public key for the i-th participant
k_i	Paillier secret key for the i-th participant
k_i^1	Paillier partial secret key for the i-th participant provided to SPU
k_i^2	Paillier partial secret key for the i-th participant provided to MU
$f_C([a],[b])$	Encrypted result of the two-party comparison protocol with input $[a]$ and $[b]$
$[a] \otimes [b] = [a \cdot b]$	Encrypted result of the two-party multiplication protocol with input $[a]$ and $[b]$

Table 4.1: Notation used throughout the chapter.

4.2.1 Genomic Background

Computation of the Genetic Risk

As previously mentioned in Chapter 2, the strength of the association between each variant and a disease is usually expressed by the *odds ratio* (OR), where the *odds* is the ratio of the probability of occurrence of the disease to that of its non-occurrence in a specific group of individuals. Thus, the OR is the ratio of *odds* in the group of individuals carrying a genetic variation (exposed) to that of those who do not carry it (unexposed). In other words, the OR illustrates by how much the risk of disease is multiplied in an individual carrying a genetic variation compared to another individual not carrying the same variation.

When multiple variants are associated with a disease, the overall genetic risk ($\mathbb{G}$) of an individual for the corresponding disease can be computed as a weighted average, based on the OR of each associated variant by using an additive model [184]. In such a model, the OR of a given variant is generally represented in terms of regression coefficient (β), where $OR = exp(\beta)$. Then, assuming $\mathrm{Pr}_g(X)$ is the susceptibility of a given individual to disease X (only considering his genomic data), his overall genetic risk can be computed as below:

$$\mathbb{G}(X) = ln(\frac{\mathrm{Pr}_g(X)}{1 - \mathrm{Pr}_g(X)}) = \alpha + \sum_{i \in \varphi_X} \beta_i \mathrm{p}^j_i(X), \tag{4.1}$$

where $\mathrm{p}^i_j(X)$ is the contribution of the j-th value (or genotype) of the i-th variant to the genetic risk (for disease X), α is the baseline risk and φ_X is the set of variants associated with disease X. Without loss of generality, in the rest of the chapter, we consider $\mathrm{p}_j^i(X)$ to be the number of risk alleles for each variant, hence $\mathrm{p}_j^i(X) \in \{0, 1, 2\}$. For simplicity, we only focus on the most common type of genetic variant, i.e., single nucleotide polymorphism (SNP). Yet, we note that also other types of genetic variants can be used.

Ancestry Inference

Ancestry inference is a necessary step to correct for population stratification before conducting genetic risk tests. Indeed, predictive markers for some genetic risk tests may have been validated only for specific populations, thus necessitating ancestry inference from genetic data to establish clinical relevance.

According to recent studies [159], ancestry information can be accurately inferred by applying principal components analysis (PCA) to genotype data from an admixed population. Intuitively, PCA infers continuous axes (or principal components) of genetic variation; these axes reduce the data to a small number of dimensions, and describe as much variability as possible. In data sets with ancestry differences between samples, these axes often have a geographical interpretation.

4.2.2 Cryptographic Background

In the following, we describe the properties of the two cryptosystems used in our algorithm for privacy-preserving genetic association studies: a modified version of Paillier cryptosystem [50, 26] and the DGK cryptosystem [69].

Modified Paillier Cryptosystem

The modified Paillier cryptosystem [50, 26] is a public-key cryptosystem supporting additively homomorphic operations and providing semantic security. To the best of our knowledge, it is the most efficient additively homomorphic scheme supporting all the requirements of the proposed system. Other additively homomorphic cryptosystems, such as ElGamal on elliptic curves, that have better performance in terms of computation and storage costs, do not support neither Algorithm 3 nor Algorithm 4, described in Section 4.4.3. We use the modified Paillier cryptosystem to encrypt the privacy-sensitive data of the participants. Semantic security is particularly required because the messages to be encrypted, during the proposed protocol described in Section 4.4.4, have low entropy and could otherwise be recovered by statistical attacks.

Let $K = (n, g, h = g^k)$ represent the public key for the modified Paillier cryptosystem. Then, the strong private key is the factorization of $n = pq$ (p, q are safe primes), and the weak private key is $k \in \left[1, n^2/2\right]$. Furthermore, let g be of order $(p-1)(q-1)/2$. Then, by selecting a random $a \in \mathbb{Z}_{n^2}^*$, it can easily be computed as $g = -a^{2n}$.

- *Encryption*: After generating a random $r \in [1, n/4]$, the encryption of a message, $m \in \mathbb{Z}_n$, is defined as:
$$E(m, K) = (T_1, T_2), \tag{4.2}$$
where (T_1, T_2) is the ciphertext pair such that $T_1 = g^r \mod n^2$ and $T_2 = h^r(1+mn)$ mod n^2.

- *Decryption*: The decryption of a ciphertext $E(m, K)$ is performed as follows:

$$D(E(m,K),k) = \Delta(T_2/T_1^k) = m, \tag{4.3}$$

  where $\Delta(u) = \frac{(u-1) \mod n^2}{n}$, for all $u \in \{u < n^2 \mid u = 1 \mod n\}$.

- *Proxy re-encryption*: Let us assume that the secret key is randomly split into two shares k_1 and k_2, such that $k = k_1 + k_2 \mod n^2$. The modified Paillier cryptosystem enables an encrypted message (T_1, T_2) to be partially decrypted into a ciphertext pair $(\tilde{T_1}, \tilde{T_2})$ using k_1 as below:

$$\tilde{T_1} = T_1 \quad \text{and} \quad \tilde{T_2} = T_2/T_1^{k_1} \mod n^2. \tag{4.4}$$

  Then, to recover the original message, $(\tilde{T_1}, \tilde{T_2})$ can be decrypted using k_2, with the aforementioned decryption function.

- *Additive Homomorphic Property*: The modified Paillier is an additively homomorphic cryptosystem and, as such, it supports some computations in the encrypted domain. In particular, let m_1 and m_2 be two messages encrypted with the same key K. Then, the encryption of the sum of m_1 and m_2 can be computed as:

$$E((m_1 + m_2), K) = E(m_1, K) \cdot E(m_2, K). \tag{4.5}$$

  As a consequence of this property, any ciphertext $E(m, K)$ raised to a constant number c is equal to the encryption of the product of the corresponding plaintext and the constant as follows:

$$E((m \cdot c), K) = E(m, K)^c. \tag{4.6}$$

For simplicity, throughout the chapter, we represent the Paillier encryption of a message m as $[m]$ and its partial decryption as $\langle m \rangle$. Operations between squared brackets, [ ], denote homomorphic operations in the ciphertext domain.

DGK Cryptosystem

The DGK cryptosystem [69] is optimized for the secure comparison of integers. Compared to the modified Paillier cryptosystem, it is more efficient in terms of encryption and decryption due to its smaller message space of a few bits. Let z represent the number of bits of the RSA modulus n, t be the size of two small primes v_p and v_q, and l be the message space size in bits such that $z > t > l$. Also let p and q be two distinct primes of equal bit length, such that $p - 1$ is divisible by v_p and $q - 1$ is divisible by v_q. Then, the public key is represented as $K_{DGK} = (n, g, h, u)$, where u is a l-bit prime, $g \in \mathbb{Z}_n^*$ with order uv_pv_q, and h is an integer with order v_pv_q. Furthermore, the private key is represented as $k_{DGK} = (p, q, v_p, v_q)$.

- *Encryption*: The encryption of a message $m \in \mathbb{Z}_u$ is:

$$E(m, r, K_{DGK}) = g^m \cdot r^n \mod n^2, \tag{4.7}$$

  where r is a random number in $\mathbb{Z}_n^*$.

- *Decryption*: The decryption needs a look-up table for all values of $\mathbb{Z}_u$. Because the message space is very small, this can be achieved efficiently. However, DGK has a particular feature: the encryption of a zero can be checked even faster just by raising the ciphertext c to the power of $w = v_p \cdot v_q$ since $c^w \mod n = 1$ if and only if c encrypts 0.

For simplicity, we represent the DGK encryption of a message m as $[\![m]\!]$.

4.3 System and Threat Models

System Model

The proposed system, illustrated in Figure 4.1 involves four parties: (i) the patients (P), (ii) a certified institution (CI) responsible for genotyping, and system initialization, i.e., generation of cryptographic keys and encryption of patients' genetic data, (iii) a storage and processing unit (SPU) where the encrypted genetic variants are stored and (iv) health care practitioners, or medical units (MU), wishing to perform genetic tests on the patients. We note that, since sequencers generating encrypted data do not exist yet, the CI would currently have access to unprotected raw genetic variants and therefore must be a trusted entity. Additionally, we are assuming a model where the encrypted genetic variants are stored in a centralized SPU rather than at the MU which maximizes efficiency and security. This is similar to applications used in business and government where the trust in the server (SPU) is much higher than in the client (MU) and allows access to be provided to several different clients (MUs) from a trusted central resource whose key task is preserving security.

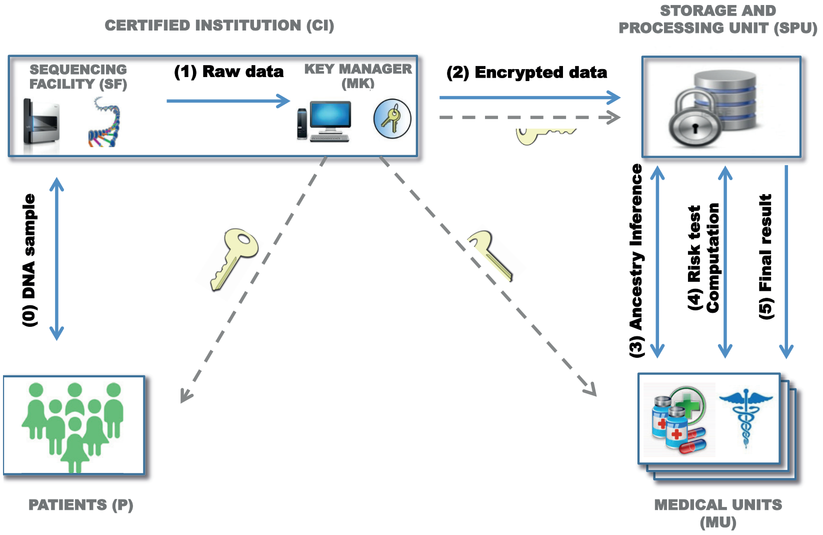
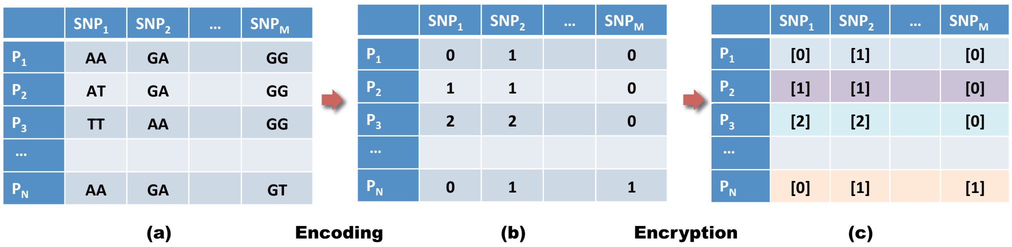


Figure 4.1: Privacy-preserving architecture for genetic testing. Genotype data and encryption keys are generated at a certified institution (CI). The patient is provided the full key, which also is randomly split between the data storage and processing unit (SPU) and the medical unit (MU). The privacy-preserving algorithms for ancestry inference and genetic risk test computation take place between the MU and the SPU.



	SNP_1	SNP_2	...	SNP_M
P_1	AA	GA		GG
P_2	AT	GA		GG
P_3	TT	AA		GG
...				
P_N	AA	GA		GT

	SNP_1	SNP_2	...	SNP_M
P_1	0	1		0
P_2	1	1		0
P_3	2	2		0
...				
P_N	0	1		1

	SNP_1	SNP_2	...	SNP_M
P_1	[0]	[1]		[0]
P_2	[1]	[1]		[0]
P_3	[2]	[2]		[0]
...				
P_N	[0]	[1]		[1]

Figure 4.2: Data encoding and encryption: (a) Genetic information represented in a matrix. (b) Each genotype is encoded by following the additive model. For example, SNP_1 has reference allele A and alternate allele T, hence $AA = 0$, $AT = 1$, and $TT = 2$. (c) The genotypes of a given participant are individually encrypted with his own public key. Different row colors represent encryption under different public keys as each participant owns a different key.

Threat Model

We consider the honest-but-curious (HBC) adversary model where the MU and the SPU are non-colluding parties and are computationally bounded (i.e., with limited computational power). In particular, both the MU and the SPU honestly follow the protocols without altering the data; but they might try to passively infer sensitive information about the patients. The HBC adversary model is a realistic assumption in healthcare where, on a daily basis, different MUs honestly collaborate and share sensitive data about patients based on mutual trust and privacy policies. We recently discussed in [38] how the HBC adversary model can be extended with negligible computational burden to the case of malicious MUs trying to actively infer a patients' sensitive data by tampering the protocols parameters.

4.4 Privacy-Preserving Genetic Risk Test with ancestry Inference

4.4.1 Overview

The main goal of the proposed solution is to compute genetic risk scores in a privacy-preserving way without leaking patients' genetic information to neither HBC adversaries.

We can summarize it as follows. First, the patients provide to CI their biological samples for genotyping. The CI encrypts each patient's variants and sends them to the SPU for secure storage. Then the MU and SPU run a secure protocol to infer the ancestry information of each patient from his encrypted variants. This protocol is run only once, offline. Finally, during the online phase, the MU and SPU securely and jointly compute the genetic risk test computation through a secure two-party protocol without ever decrypting patients' data. In such a protocol, the MU specifies a set of markers to the SPU for each test to be run and obtains only the correspondent final genetic risk scores.

4.4.2 System Initialization and Data Encryption

During the initialization phase, the patients enrolled in the study provide, upon consent, their biological samples for genotyping to the CI that generates and distributes to each of the patients a pair of cryptographic keys for the modified Paillier cryptosystem (described in Section 4.2.2). Each key pair is composed of a public key and a secret key (K_i, k_i). The public key of each participant is also distributed to both the SPU, and the MU. After key distribution, CI encrypts each participant's SNPs individually. SNPs are encoded with the additive model [167], where each copy of the risk allele modifies the association with genetic risk in an additive form (see Figure 4.2(a-b)). We assume the alternate allele to be the risky allele, hence we encode SNPs that are homozygous reference with 0, heterozygous with 1 and homozygous alternate with 2. Let K_i represent the public key for the i-th participant; then $[S_i^j]_{K_i}$ denotes the encrypted genotype of his j-th SNP. For the sake of simplicity, in the rest of the paper we refer to $[S_i^j]_{K_i}$ as $[S_i^j]$, unless specified otherwise.

The secret key is randomly divided into two shares that are distributed to the SPU and the MU, respectively. In particular, let k_i denote the secret key for the i-th participant. Then, k_i is randomly split into k_i^1 and k_i^2, such that $k_i = k_i^1 + k_i^2 \mod n^2$. As a result, k_i^1 is provided to the SPU and k_i^2 to the MU, thus no party, except the participant himself, has the complete secret key.[1] Note that for simplicity, we assume the presence of a single MU. However, in the case of multiple MUs, k_i^2 will be provided to all of them.

The CI also establishes symmetric cryptographic keys to protect the communication between the parties from eavesdroppers. We assume that the CI, as a trusted entity, can also handle the update and the revocation of the cryptographic keys but cannot replace the SPU for the storage of the genetic data.

SNPs' identifiers are encrypted through a deterministic encryption (DTE) scheme by using the symmetric key established between the CI and the MU. Note that this type of encryption prevents the SPU from knowing which SNPs are being used during the genetic test but allows the MU to still select them as the equality property of the plaintext identifiers is preserved.

Finally, after encryption, the CI sends the encrypted SNP genotypes to the SPU for storing them in the data model illustrated in Figure 4.2(c)). Participants' data is stored using pseudonyms (without revealing the identities of the participants) to prevent the SPU from associating a SNP to a specific individual.

4.4.3 Privacy-Preserving Ancestry Inference

As mentioned in Section 4.2.1, ancestry inference is a necessary step to correct for population stratification before conducting genetic risk tests. Our main goal is to infer the participants' ancestry groups without revealing any sensitive information, neither to the SPU nor to the MU. The proposed algorithm for secure ancestry inference consists of a secure two-party protocol that takes place only once between the MU and the SPU during the "offline" phase. The protocol takes in input the encrypted SNP genotypes stored at the SPU and outputs the encrypted ancestry information for each patient, without leaking any private genetic information of the patients. Such ancestry information is encoded as a binary vector $(\mathbf{A}_i)$ per participant of length equal to the number of ancestry groups considered in the study. Each element A_i^g of the vector contains a binary value (either 0 or 1) indicating whether the patient belongs to the g-th ancestry group or not. Below, we describe the protocol in detail. The main operations are also illustrated in Figure 4.3.

[1] We assume in our solution that only the participant, who is the owner of the data, has the full control on his genetic information.

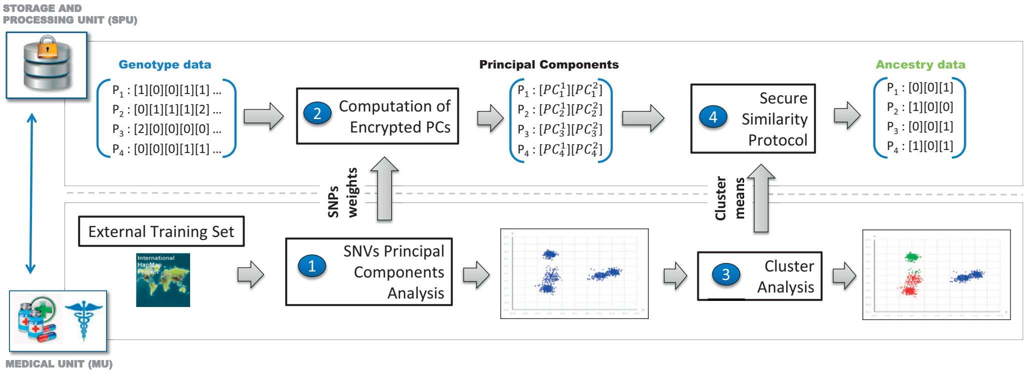


Figure 4.3: Main steps of the protocol for privacy-preserving ancestry inference.

- Step 1: SNPs Principal Components Analysis.

The first step of the proposed privacy-preserving ancestry-inference algorithm consists in the MU performing a PCA on an external reference panel (or training set) of plaintext SNP genotypes of its choice. Such reference panel can be retrieved from international genomics-related projects like the *HapMap* project [196] or the *1000Genomes* project [194], where admixed populations have been extensively studied.

As a result of such a PCA, the MU obtains a set of SNP weights ($\mathbf{W}$) that are then sent to the SPU to compute the encrypted principal components (PCs) for each patient.

- Step 2: Computation of the Encrypted SNP Principal Components. Once the SPU receives the SNP weights $\mathbf{W}$, it homomorphically computes for each patient the encrypted principal components $[\mathbf{PC_i}]$ as

$$[PC_i^l] = [\sum_{v=1}^{V} S_i^v \cdot W_v^l] = \prod_{v=1}^{V} [S_i^v]^{W_v^l}, \qquad l = 1, ..., L \tag{4.8}$$

where V is the number of SNPs and L is the number of principal components. We note that $L = 2$ has been proved to be a reasonable value for identifying continental ancestry groups in admixed populations [159].

- Step 3: Cluster Analysis. While the SPU computes the encrypted principal components from the encrypted SNPs of the study participants, the MU performs a cluster analysis on the principal components of the individuals in the reference panel in order to identify the reference ancestry groups. As such, the MU keeps the L plaintext principal components obtained from the reference panel computed at Step 1 and performs a *k-means* clustering in order to partition the N individuals of the reference panel into k clusters or ancestry groups.[2] Each individual belongs to the ancestry group with the nearest *mean*, that serves as an identifier of the ancestry group itself. The set of cluster means $\mathbf{C}$ is sent by the MU to the SPU so that they can be securely compared with the encrypted PCs of each participant in order to obtain his encrypted ancestry information.

- Step 4: Secure Similarity Protocol Given the patients' encrypted principal components $[\mathbf{PC_i}]$ and the plaintext vector of cluster means $\mathbf{C}$, the SPU infers the

[2] The value of k depends on the reference panel selected for the PCA. Note that different MUs can choose different reference panels.

Algorithm 3: Secure Comparison $f_C([a],[b])$

```
Input  : @SPU: $[a]$, $[b]$ and $prk^1$. @MU: $prk^2$.
Output: @SPU: $f_C([a],[b]) = [(a \leq b)]$. @MU: $\perp$. // $a$ and $b$ are two $l$-bit integers
SPU computes $[z] \leftarrow [a] \cdot [b]^{-1} \cdot [2^l] = [a - b + 2^l]$
SPU generates a random number $r$, $0 \leq r < n^2$, and blinds $[z]$ $[\hat{z}] \leftarrow [z] \cdot [r] = [z + r]$.
SPU partially decrypts $[\hat{z}]$, $\langle \hat{z} \rangle \leftarrow D([\hat{z}], prk^1)$, and sends $\langle \hat{z} \rangle$ to MU.
MU decrypts $\langle \hat{z} \rangle$ with $prk^2$, $\hat{z} \leftarrow D(\langle \hat{z} \rangle, prk^2)$
MU computes $\beta \leftarrow \hat{z} \mod 2^l$.
SPU computes $\alpha \leftarrow r \mod 2^l$.
SPU and MU run a modified DGK comparison protocol [205] with private inputs $\alpha$ and $\beta$ and
  obtain $\delta_{SPU}$ (@SPU) and $\delta_{MU}$ (@MU).
MU computes $\frac{\hat{z}}{2^l}$ and sends $\left[\frac{\hat{z}}{2^l}\right]$ and $[\delta_{MU}]$ to SPU.
SPU computes $[(\beta < \alpha)]$:
if $\delta_{SPU} = 1$ then
    $[(\beta < \alpha)] \leftarrow [\delta_{MU}]$
else
    $[(\beta < \alpha)] \leftarrow [1] \cdot [\delta_{MU}]^{-1}$.
end
SPU computes $[(a \leq b)] \leftarrow \left[\frac{\hat{z}}{2^l}\right] \cdot (\left[\frac{r}{2^l}\right] \cdot [(\beta < \alpha)])^{-1}$
```

encrypted ancestry group of each participant through a *secure similarity protocol.* Intuitively, without revealing any sensitive information, the SPU assigns each participant to one of the G ancestry groups based on the maximum similarity between his encrypted PCs and the cluster means. In summary, for each participant, the protocol consists in (i) securely computing the similarity between his encrypted PCs and each cluster's mean, (ii) finding the maximum encrypted similarity, and (iii) computing the encrypted binary values that indicate the ancestry group he belongs to.

To design such a protocol, we rely on two secure subprotocols adapted from [78]. The first one, described in Algorithm 3, consists of a secure two-party comparison protocol that, given two ciphertexts $[a]$ and $[b]$ encrypted under the same public key, outputs the encrypted result of their comparison. Let $f_C([a],[b])$ represent the encrypted result of the comparison protocol with inputs $[a]$ and $[b]$, where a and b are l-bit integers. Then, $f_C([a],[b])$ outputs the encryption of 1 when $a \leqslant b$ and, otherwise, the encryption of 0. Note that homomorphic encryption does not preserve any order in the ciphertext domain, hence Algorithm 3 is needed to let a party compare two ciphertexts in a privacy-preserving way.

The second protocol, described in Algorithm 4, is a secure two-party multiplication protocol that, given two ciphertexts $[a]$ and $[b]$ encrypted under the same public key, provides the multiplication of their corresponding plaintexts. We denote by $\otimes$ the secure multiplication protocol, such that $[a] \otimes [b] = [a \cdot b]$. Note that the modified Paillier cryptosystem used in the proposed solution is only additively homomorphic and does not support multiplication between ciphertexts. Hence, Algorithm 4 is needed to obtain the encrypted product of two plaintext messages, given only their corresponding ciphertexts.

The *secure similarity protocol* requires as input parameters the set of encrypted principal components $[\mathbf{PC_i}]$ along with the vector of clusters' means $\mathbf{C}$; it outputs the encrypted ancestry information $[\mathbf{A}]$. The details of the protocol are described in Algorithm 5.

Once $[\mathbf{A_i}]$ is obtained, this can be used to establish clinical relevance for genetic risk testing.

Algorithm 4: Secure Multiplication $[a] \otimes [b] = [a \times b]$

```
Input  : @SPU: $[a]$, $[b]$ and $prk^1$. @MU: $prk^2$.
Output: @SPU: $[a \cdot b]$. @MU: $\perp$.
SPU generates two random numbers $r_1$ and $r_2$
SPU blinds $[a]$ and $[b]$: $[\hat{a}] \leftarrow [a] \cdot [-r_1] = [a - r_1]$, $[\hat{b}] \leftarrow [b] \cdot [-r_2] = [b - r_2]$
SPU partially decrypts $[\hat{a}]$ and $[\hat{b}]$ with $prk^1$: $\langle \hat{a} \rangle \leftarrow D([\hat{a}], prk^1)$, $\langle \hat{b} \rangle \leftarrow D([\hat{b}], prk^1)$
SPU sends $\langle \hat{a} \rangle$ and $\langle \hat{b} \rangle$ to MU
MU decrypts $\langle \hat{a} \rangle$ and $\langle \hat{a} \rangle$ with $prk^2$: $\hat{a} \leftarrow D(\langle \hat{a} \rangle, prk^2)$, $\hat{b} \leftarrow D(\langle \hat{b} \rangle, prk^2)$
MU computes $[\hat{a} \cdot \hat{b}]$ and sends it to SPU
SPU computes $[a \cdot b] \leftarrow [\hat{a} \cdot \hat{b}] \cdot [a]^{r_2} \cdot [b]^{r_1} \cdot [-r_1 \cdot r_2] = [\hat{a} \cdot \hat{b} + r_2 \cdot a + r_1 \cdot b - r_1 \cdot r_2]$
```

Algorithm 5: Secure Similarity Protocol

```
Input  : @SPU: $[\mathbf{PC_i}]$ and $\mathbf{C}$; @MU: $\perp$
Output: @SPU: $[\mathbf{A_i}]$. @MU: $\perp$
// Let $I$ be # of participants, $G$ # of ancestry groups (or clusters), and $L$ # of
   selected top PCs.
// SPU computes the encrypted similarities between encrypted PCs and cluster means:
foreach $g : 0 < g \leq G$ do
    $[Sim_i^g] \leftarrow [\sum_{l=1}^{L} (PC_i^l - C_l^g)^2] = \prod_{l=1}^{L} ([PC_i^l] \cdot [-C_l^g]) \otimes ([PC_i^l] \cdot [-C_l^g])$
end
// SPU computes the maximum similarity:
$[M] \leftarrow [Sim_i^1]$
foreach $g : 1 < g \leq G$ do
    $[M] \leftarrow [M \cdot (Sim_i^g \leq M) + Sim_i^g \cdot (M \leq Sim_i^g)] =$
    $\{[M] \otimes f_C([Sim_i^g], [M])\} \cdot \{[Sim_i^g] \otimes f_C([M], [Sim_i^g])\}$
end
// SPU computes the encrypted value of each ancestry group for each participant:
foreach $g : 0 < g \leq G$ do
    $[A_i^g] \leftarrow f_C([M], [Sim_i^g])$
end
```

4.4.4 Privacy-Preserving Genetic Risk Test Computation

The privacy-preserving computation of the risk test is performed as follows. Once a clinician at the MU wants to compute the genetic risk of given patient i for condition X, the MU sends to the SPU the set of encrypted identifiers of the SNPs correlated with X, φ. The SPU retrieves from its database the corresponding set of encrypted SNPs of that patient, $\{[\mathbf{S_i}(X)]\}$, and sends them back to the MU, along with the relevant encrypted ancestry information $[A_i^g]$. We assume that the genetic risk score, $\mathbb{G}(X)$, is computed with an additive model as proposed in [170]. This means that each copy of the risk allele modifies the association with genetic risk in an additive form. The computation of its encrypted version, $[\mathbb{G}(X)]$, is based on the homomorphic properties of the Paillier cryptosystem, as shown below:

$$[\mathbb{G}(X)] = \Big[A_i^g \times \Big(\alpha + \sum_{j \in \varphi} \beta_j S_i^j\Big)\Big] = [A_i^g] \otimes \Big([\alpha] \times \prod_{j \in \varphi} [S_i^j]^{\beta_j}\Big), \tag{4.9}$$

where β_j represents the contribution of S^j to condition X, α represents the baseline risk, and $\otimes$ represents the secure multiplication protocol described in Algorithm 4.

Finally, the encrypted genetic risk score is sent back to the SPU, where it is partially decrypted by using the first part of the patient's secret key k^1 to obtain $[\hat{\mathbb{G}}(X)]$. The SPU sends $[\hat{\mathbb{G}}(X)]$ back to the MU, where it is finally decrypted with the second part of the patient's secret key k_P^2 to obtain the final plaintext risk score $\mathbb{G}(X)$.

4.4.5 Privacy-Preserving Integration of Clinical and Environmental Factors

The proposed solution also enables for the integration of non-genomic (clinical and environmental) factors that are usually required in the computation of the risk when there are strong known influences of environment on a particular trait, such as metabolic disorders, in order to increase the accuracy of the test.

During the initialization phase, along with their biological samples, patients can also provide the CI with their clinical and environmental information such as age, disease conditions, smoking behavior, etc.. This information is encoded as a set of categorical variables and encrypted for secure storage at the SPU.

At the time of the risk test computation, the MU asks the SPU to also provide the encrypted clinical environmental factors of the patient that are correlated with the condition being tested X along with the encrypted genetic markers. Let $[\mathbf{N_i}]$ be the set of encrypted values of the clinical and environmental attributes of the i-th patient and Δ be the set of encrypted identifiers the clinical, where $N_i^j \in \{0,1\}$ for the simplicity of the presentation, and environmental attributes that are required for the computation of the risk for disease X. That is, $N_i^j = 1$ if the patient has the corresponding clinical or environmental attribute, and $N_i^j = 0$ otherwise. Then, the (final) risk score of the patient (for disease X) is computed as below:

$$[\mathbb{R}(X)] = \Big[A_i^g \times \Big(\alpha + \sum_{j\in\varphi} \beta_j S_i^j + \sum_{t\in\Delta} \bar{\beta}_t N_i^t\Big)\Big] = [A_i^g] \otimes \Big([\alpha] \times \prod_{j\in\varphi} [S_i^j]^{\beta_j} \times \prod_{t\in\Delta} [N_i^t]^{\bar{\beta}_t}\Big), \quad (4.10)$$

where $\bar{\beta}_t$ is the regression coefficient of the t-th clinical or environmental attribute. As for the encrypted genetic risk score $[\mathbb{G}(X)]$, the encrypted final risk score $[\mathbb{R}(X)]$ can be partially decrypted at the SPU with k^1 and finally decrypted at the MU with k^2. Finally, the MU computes the final risk of the patient for condition X as $\frac{e^{\mathbb{R}(X)}}{1+e^{\mathbb{R}(X)}}$.

4.5 Use Case: Pharmacogenetics Tests for HIV Treatment

We deployed and evaluated the proposed model in five of the seven outpatient clinics of the Swiss HIV Cohort Study (SHCS) [178] for genetic testing in HIV treatment. The SHCS central data center was used as SPU and the EPFL played the role of the CI.

4.5.1 Patient Characteristics and Genetic Variant Selection

A total of 230 HIV infected individuals initiating Antiretroviral Therapy (ART) enrolled in the SHCS were included in this study and were genotyped for 4,149 variants. All patients signed consent for genetic testing and the institutional review boards of the SHCS centers approved the study.

We identified 71 markers from the literature that were informative for 17 traits relevant to HIV outcomes. Clinically informative markers fell into 3 categories 1) HIV/HCV progression [71, 83, 188, 198, 72] and response to therapy [136, 131, 103, 63, 173, 90, 84], 2) pharmacokinetics of efavirenz (EFV) [168, 23, 100, 73, 44], nevirapine (NVP) [168, 23], etravirine ETV35 or lopinavir (LPV) [133], 3) metabolic traits including vitamin D deficiency [97], coronary artery disease [171, 179], cholesterol and triglyceride levels [169] and type 2 diabetes [98]. Testing included single and multi-marker deterministic tests, where the presence of a risk variant or combination of variants is highly likely to cause the associated trait, and multi-marker risk scores, where several variants combine together to moderately impact trait risk. The prediction scheme for each reported results is provided in [141]. Additional markers predictive of patient ancestry (n=111) [126], or HLA type (n=250) were also incorporated. Markers capturing variation across a set of absorption metabolism distribution and excretion (ADME) genes (n=3,717) were also included [132]. Though these variants were not used for clinical prediction they were used to improve the precision of ancestry inference.

The majority of variants were genotyped using a custom array on the Illumina Infinium platform. Informative variants that could not be included on the genotyping array due to technical issues, (CCR5Δ32 (rs333), CYP2B6 *6 (rs3745274), UGT1A1 *28 (rs8175347)), were genotyped by TaqMan allelic discrimination from Applied Biosystems or fragment size-based analysis [41]. Samples and variants were filtered out if they did not pass quality thresholds for genotyping rate and Hardy-Weinberg equilibrium. Four-digit classical class I HLA allele genotypes were imputed for individuals with inferred European ancestry using the SNP2HLA pipeline [115]. This operation was performed during the initialization phase after genotyping at the CI. Only alleles with an imputation quality above 0.98 were kept.

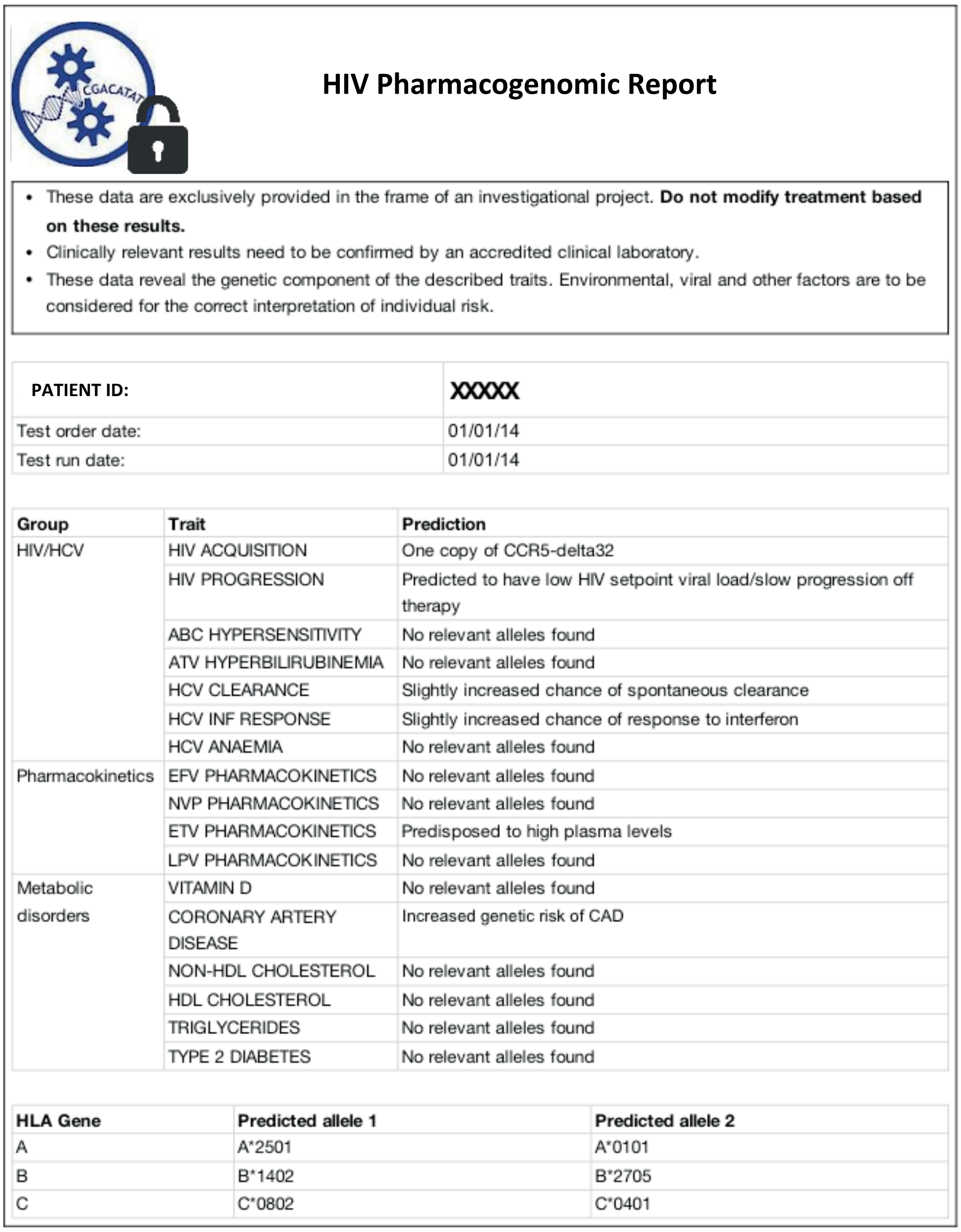
4.5.2 Interpretation and Result Reporting

To maximize clinical utility, clinicians were provided with interpreted test results for each trait, rather than the raw patient genotypes through a dedicated client application implemented in Java. Semantics were adapted to indicate the confidence of each test individually, thus, when significant genetic markers were observed, an alert specific to the test was returned, otherwise a result of "no significant alleles found" was given. An example report returned to physicians is shown in Figure 4.4. Each report included a disclaimer indicating the investigational nature of the study. Importantly, the release of genetic data to the clinics was delayed and, by design, not meant to modify choice of treatment.

4.6 Performance Evaluation

To assess the feasibility of applying genetic testing in the clinical setting, we implemented the proposed privacy-preserving solution as a client-server Java application (see Figure 4.5 for a screen shot of the front-end graphical user interface).

We performed all operations on encrypted data stored in a MySQL server at the SHCS data center (playing the role of the SPU) based on the secure protocol outlined in Figure 4.1 and described in detail in Section 4.4 (with the exception of HLA allele imputation that was directly performed on the plaintext data at the CI during the initialization). We used a modified version of the Paillier cryptosystem [50] supporting both additively homomorphic encryption and proxy re-encryption to encrypt the genetic variants of each patient, and the CCM mode [76] of the advanced encryption standard [185] to deterministically encrypt their identifiers. We tested the performance on off-the-shelf hardware (an Intel Core i7-2620M CPU with 2.70 GHz processor running the Windows 7 Operating System) by using a Paillier's security parameter of 4096 bits size. The encryp-



HIV Pharmacogenomic Report

- These data are exclusively provided in the frame of an investigational project. **Do not modify treatment based on these results.**
- Clinically relevant results need to be confirmed by an accredited clinical laboratory.
- These data reveal the genetic component of the described traits. Environmental, viral and other factors are to be considered for the correct interpretation of individual risk.

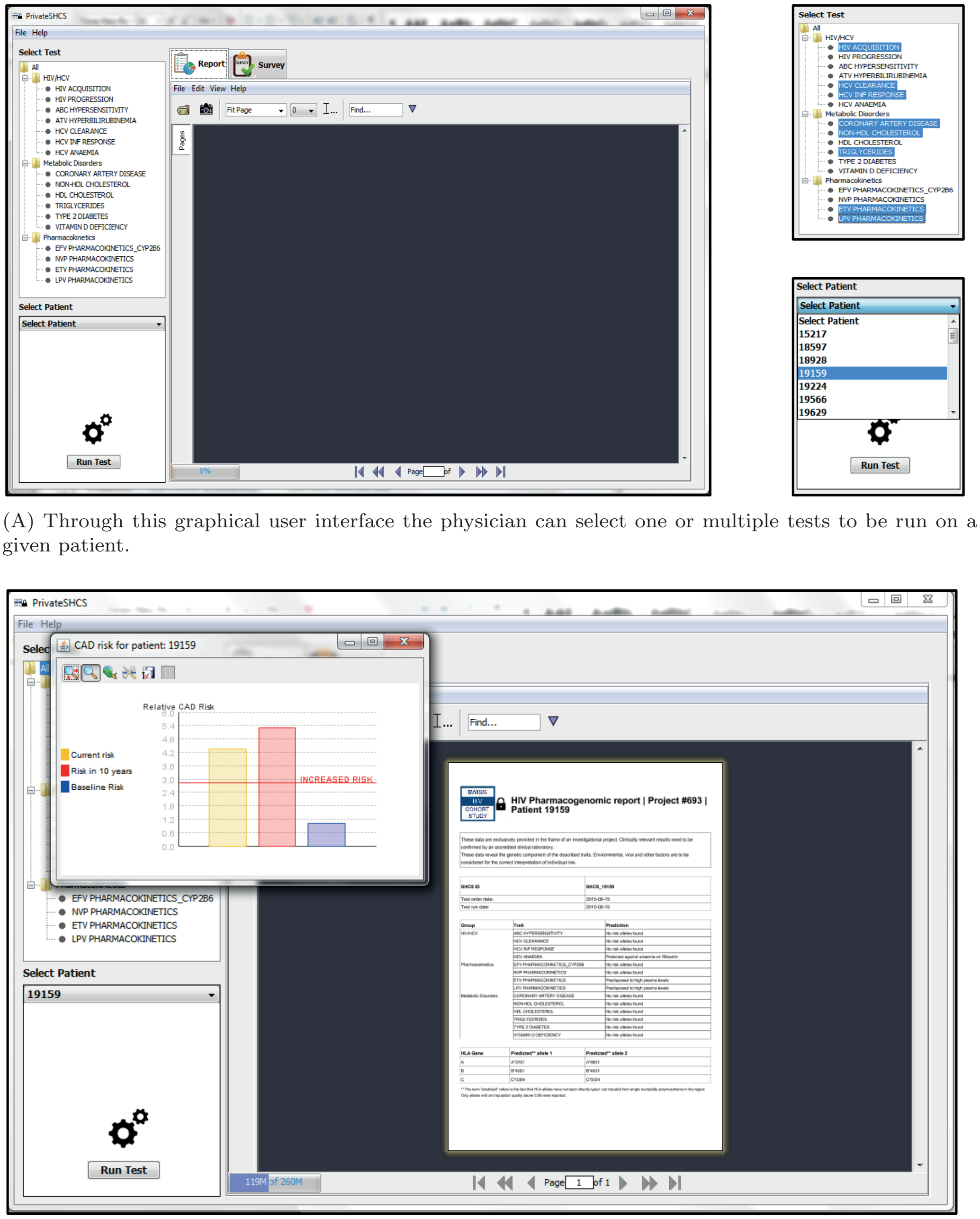
PATIENT ID:	**XXXXX**
Test order date:	01/01/14
Test run date:	01/01/14

Group	Trait	Prediction
HIV/HCV	HIV ACQUISITION	One copy of CCR5-delta32
	HIV PROGRESSION	Predicted to have low HIV setpoint viral load/slow progression off therapy
	ABC HYPERSENSITIVITY	No relevant alleles found
	ATV HYPERBILIRUBINEMIA	No relevant alleles found
	HCV CLEARANCE	Slightly increased chance of spontaneous clearance
	HCV INF RESPONSE	Slightly increased chance of response to interferon
	HCV ANAEMIA	No relevant alleles found
Pharmacokinetics	EFV PHARMACOKINETICS	No relevant alleles found
	NVP PHARMACOKINETICS	No relevant alleles found
	ETV PHARMACOKINETICS	Predisposed to high plasma levels
	LPV PHARMACOKINETICS	No relevant alleles found
Metabolic disorders	VITAMIN D	No relevant alleles found
	CORONARY ARTERY DISEASE	Increased genetic risk of CAD
	NON-HDL CHOLESTEROL	No relevant alleles found
	HDL CHOLESTEROL	No relevant alleles found
	TRIGLYCERIDES	No relevant alleles found
	TYPE 2 DIABETES	No relevant alleles found

HLA Gene	Predicted allele 1	Predicted allele 2
A	A*2501	A*0101
B	B*1402	B*2705
C	C*0802	C*0401

Figure 4.4: Example report returned to clinicians. Interpreted test results are shown for each trait. An alert specific to the test was returned when a significant test score or genetic marker was observed. Otherwise, a result of “no significant alleles found” was displayed.

tion time of the genotype scales linearly with the number of markers and takes 171ms for a single marker with a storage size of 1KB. Thus, total encryption time per patient for all genotypes took 12 minutes generating a ciphertext size of 4MB. We note, the encryption of 4 million markers (the approximate number of variant genotypes carried by a given individual) would take approximately 200 hours. Yet, importantly, this initial encryption is only required once (in the initialization phase of the system) and could be expedited by pre-computation of certain exponents required for encryption ((g^r, h^r) in



(A) Through this graphical user interface the physician can select one or multiple tests to be run on a given patient.

(B) Encrypted results are decrypted, interpreted and presented as a standardized text report.

Figure 4.5: Front-end graphical user interface.

Equation 4.2) and parallel computation, resulting in an encryption time on the order of minutes.

By design, this study incorporated genetic tests where the predictive markers have only been validated in European populations, necessitating ancestry inference from genetic data to establish clinical relevance. We used the HapMap reference panel [65] as a training dataset for the privacy-preserving ancestry inference algorithm. The total time to privately compute the ancestry information was 11.6s per individual, which could be reduced to 3.8s by also pre-computing pairs of (g^r, h^r) used in the encryption algorithm in Equation 4.2.

After ancestry inference, a set of 169 individuals predicted to be of European ancestry was identified (Figure 4.6) for whom full results and HLA alleles could be reported. For the remaining individuals, a result of "Prediction not available for this population" was reported for the ancestry-limited tests. Predicted ancestry was highly similar to self-reported ancestry (94%) and was incorporated solely to determine which test results were valid on an individual basis, and not reported to the clinician or patient.

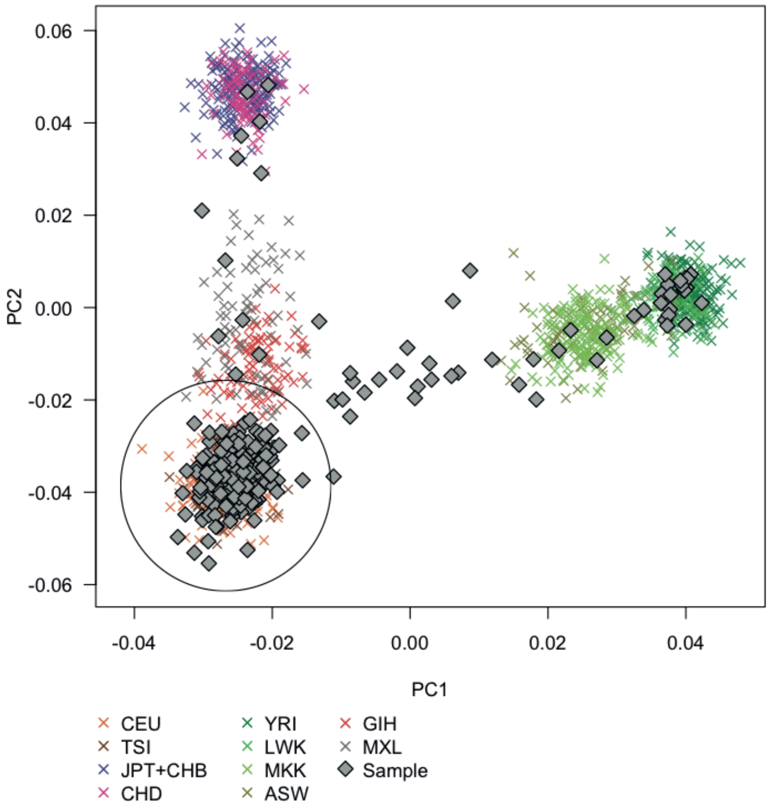


Figure 4.6: Clustering of patient samples (grey diamonds) with populations of different ancestries. Principal component (PC) analysis and ancestry inference were performed in a privacy-preserving way through a secure two-party protocol between the storage and processing unit (SPU) and the medical unit. Encrypted ancestry information was generated and stored at the SPU. Sample clustering with the HapMap CEU (Utah residents with Northern and Western European ancestry collected by the Centre d'Etude du Polymorphisme Humain) and TSI (Tuscans in Italy) populations (i.e., those within the circle) were considered European for the purposes of report generation. ASW, African ancestry in southwest United States; CHB, Han Chinese in Beijing, China; CHD, Chinese in Metropolitan Denver, Colorado; GIH, Gujarati Indians in Houston, Texas; JPT, Japanese from Tokyo, Japan; LWK, Luhya in Webuye, Kenya; MKK, Massai in Kinyawa, Kenya; MXL, Mexican ancestry in Los Angeles, California; YRI, Yoruba in Ibidan, Nigeria.

For risk test computation, we observed an average time of 865ms for a theoretical test using 50 markers. Thus, after encryption and ancestry inference, all tests in the current study could be performed and reported in less than 1 second.

4.7 Security Analysis

The proposed system preserves the privacy of patients' genomic (and potentially clinical and environmental) data in the honest-but-curious adversarial model by relying on the security guarantees provided by the modified Paillier cryptosystem and the DGK cryptosystem. Provided that both the MU and the SPU do not collude, none of the two parties can reconstruct a patient's secret key and individually decrypt his genetic information. Only the MU can eventually obtain the result of the test. For the extensive security evaluation of the modified Paillier and DGK cryptosystems we refer the reader to [50] and [68], respectively.

Yet, we note that the proposed system may be susceptible to a brute force attack (i.e. systematically checking all possible keys until the correct one is found). The feasibility of this type of attack depends on the length of the key used (a cipher with a key length of N bits can be broken in an average time of 2N-1). In our implementation, we used a key of 4096 bits, a key size that is compliant with the recommendations of the National Institute of Standards and Technology and will provide security for the next 30+ years based on the envisioned improvement in computing power [37]. However, for some test results reported, there is a strong linkage between the results and the underlying causal genotype (e.g. HLA-B*57:01 and Abacavir hypersensitivity). Thus, the inclusion of a large number of such tests may present it's own risk to patient privacy. In the case where many such conditions are to be included in the same report, other techniques, such as result obfuscation, may be desirable [33].

Finally, we also note that to prevent the SPU from inferring the nature of the conducted test based on the number of requested SNPs, the MU can include an arbitrary number of "dummy" SNPs with null contribution to the condition being tested. We leave for future work the study of the dummy addition strategy that ensures the best privacy vs. efficiency trade-off.

4.8 Acceptability Questionnaire

Physicians were asked to complete a survey aimed at gauging their acceptance and interest in the proposed privacy-preserving system for pharmacogenetics tests. The questionnaire was directly embedded into the front-end Java application used to run the tests and obtain the interpreted reports. Prior to filling the questionnaire, physicians had to read a short description of the project illustrating the key elements of the privacy-preserving system they were testing. Upon completion, surveys were automatically sent to the SHCS Data Center for incorporation into an anonymized database.

4.8.1 Questionnaire Structure

The questionnaire consisted of two sections. The first section included four questions covering physicians' basic demographics such as age, gender, grade (options: Resident, Attending and Head of Department) and SHCS center. This information was necessary to verify that our sample was representative of the general physician population.

The second section of the questionnaire aimed at obtaining insights on physicians' acceptance level regarding our privacy-preserving system for genetic testing, and more generally, on their attitude toward the adoption of privacy-enhancing technologies for the protection of genetic data in the clinical context. As such, we studied physicians' acceptance level under three different angles represented by the following three observable variables: clinical utility, privacy concerns and system usability. For each of these variables we proposed a set of statements, representative of the concept, about which participants had to indicate their level of agreement with a score from 0 to 4, where 0 means "strongly disagree", 1 means "disagree", 2 means "neutral", 3 means "agree" and 4 means "strongly agree". For the three above-mentioned variables we designed four, six and four statements, respectively.

Prior to this study, we pilot-tested the questionnaire with two attending doctors working on infectious diseases at a clinic not involved in our study. The goal was to assess if the statements were fully understandable and without ambiguity. Both doctors completed all sections without reporting any ambiguity. Hence, the questionnaire was used for the study without further validation.

4.8.2 Quantitative Analysis

We analyzed survey responses with Python Data Analysis Library and with STATA^{TM} software version 14.2. For each statement, we first computed the proportions of physicians per agreement level. Then, for each participant we aggregated his/her scores grouped by variable (i.e., clinical utility, privacy concerns, system usability) in order to obtain a triplet score where each element represented the sum of the scores for the statements related to a given concept. We hypothesize that the three observed variables are the collective expression of the general acceptance level of our system. This is an unmeasured (latent) variable whose relationship with the 3 indicators can be quantified via Confirmatory Factor Analysis (CFA) [124] in the framework of structural equation models (SEM). We derived the relative importance of each indicator from SEM, by estimating standardized coefficients. We assessed the goodness of fit of the model using the following indices: *Comparative Fit Index* (> 0.95), *Tucker Lewis Index* (> 0.95), and root mean square error of approximation (< 0.06).

4.8.3 Results

Of 55 SHCS-affiliated doctors invited to the study, 38/55 (69%) tested the proposed system at least once and completed the survey. There were 8 (21%) females and 30 (79%) males with mean age of 41 years and a maximum of 65. Of the 38 participants, 12 (31%) were attending physicians, 11 (29%) were heads of department and 15 (40%) were residents. Our sample is marginally younger and with some over-representation of male than the overall physician population as compared to the 2016 Report of the Swiss Doctors' Federation (FMH), which reported that the mean age of physicians is 46 and that 60% of physicians in Switzerland are male [108].

We report the distributions of physicians' scores for the different statements in the survey in Figure 4.7. We observed that statements for all three variables received high consensus (i.e., agreement level of 3 or 4) from a large majority of the physicians. Concerning the clinical utility of our system, 71% of participants considered the information provided useful and 68% that it was worth a therapy modification when actionable. Only

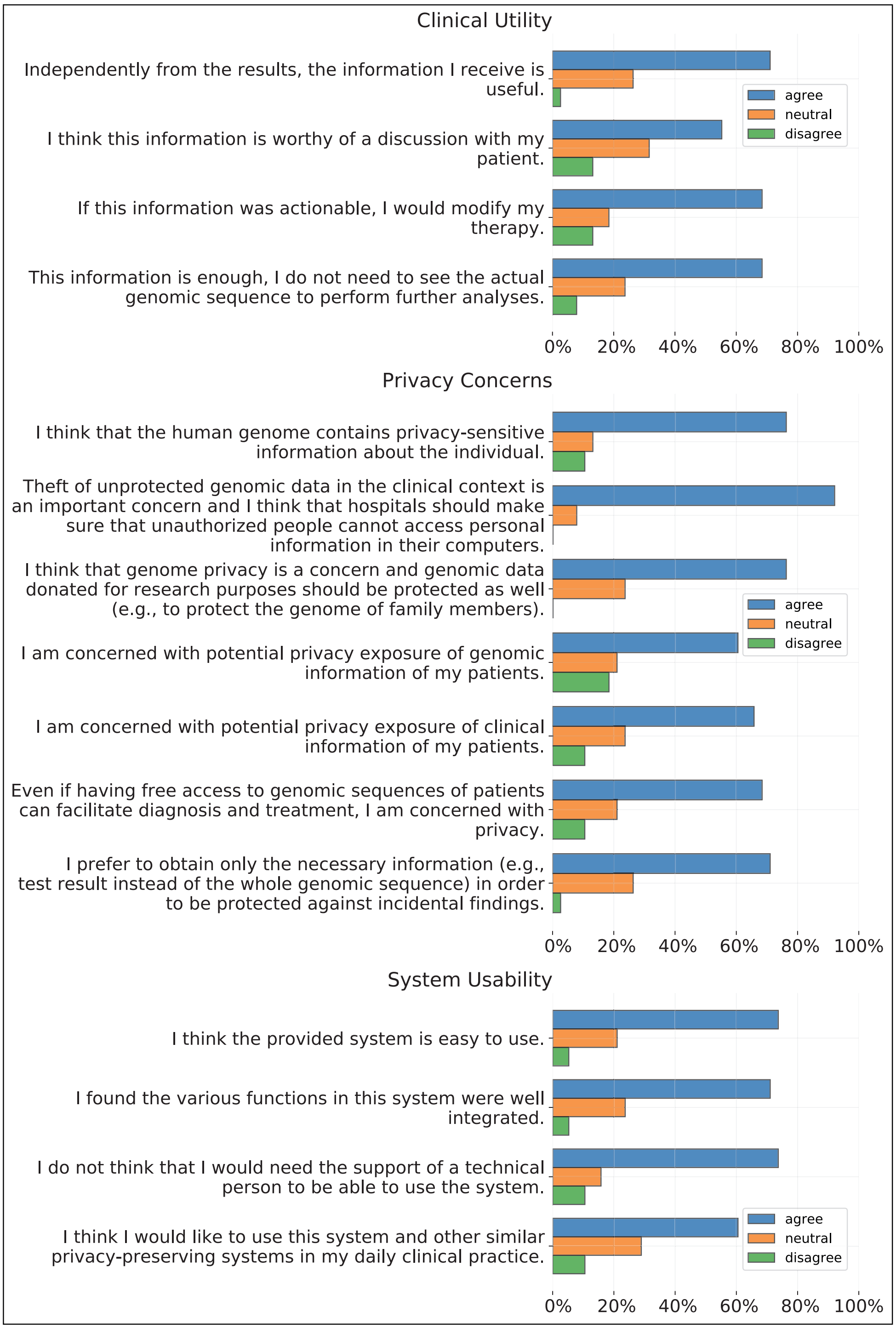


Figure 4.7: Distributions of scores per claim in the questionnaire grouped by observed variable. The category “agree” includes scores > 2, while the category “disagree” includes scores < 2.

8% of physicians were unsatisfied with the level of information and wanted to have access to the patient’s actual DNA sequence.

Regarding the magnitude of participants' privacy concerns, 76% agreed or strongly agreed that the human genome actually contains privacy-sensitive information. Physicians showed similar concerns for potential exposure of their patients' clinical (65%) and genomic (60%) data. Moreover, almost all (92%) thought that it is the responsibility of the healthcare institution storing the genetic data to make sure that only authorized people can access them. Finally, we observed that for a majority of physicians (68%), a better diagnosis and treatment should not be at the expense of privacy and that in general (71%), physicians prefer to obtain only the information necessary for the diagnosis in order to avoid the risk of incidental findings.

Concerning the system usability, participants had a user-friendly experience with our system. Physicians had no perception of the sophisticated privacy-preserving techniques running behind the scenes as the encryption, result decryption and cryptographic key management were performed automatically by the system. Only 10% of those surveyed required technical support and were skeptical about the use of similar privacy-preserving systems in their daily clinical practice.

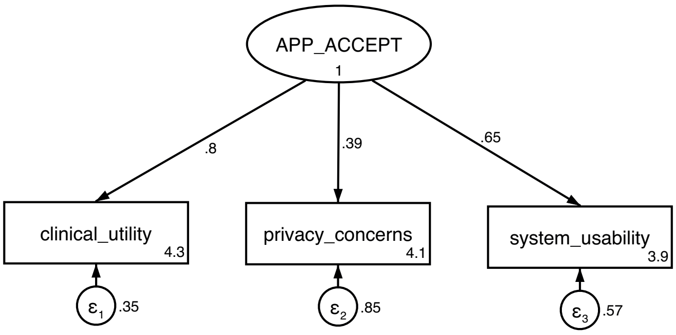


Figure 4.8: Confirmatory factor model where the oval indicates the latent (unmeasured) overall acceptance of the system or factor, (APP_ACCEPT), rectangles indicate the observed variables and single-headed arrows show causality from the factor to the indicators. Numbers along the arrows are the SEM standardized weights that indicate the relative importance of the observed variables as measures of general acceptance of the system. Numbers at the epsilon circles are the error term coefficients of the SEM. Goodness of fit criteria of CFA were satisfied.

Figure 4.8 shows the results of CFA. The reported standardized weights measure the relative importance of each of clinical utility, privacy concerns and system usability for participants' overall acceptance of the system. We can observe that the impact on the acceptance level of clinical utility (0.8) is twice the impact of privacy concerns (0.39).

4.9 Discussion

In this chapter, we assessed the steps required for deployment of privacy-preserving genetic testing in clinical care. We proposed a new privacy-preserving model for genetic testing with ancestry inference and delivery of interpreted information to clinicians and tested its applicability in a real-life operational setting based on HIV treatment. Our model included the protection of patients' genetic data against honest-but-curious adversaries and the possibility to conduct various operations (ancestry analysis, generation of genetic scores and report generation) within an encrypted environment without significant additional cost in terms of computation and storage overhead.

A central outcome of this study is the delivery of interpreted genetic data. Processes such as correction for ancestry, retrieval of imputed HLA alleles, and calculation of genetic scores were securely performed in the background and clinicians only received a fully interpreted report rather than raw genetic data.

The design of the use-case study included the genotyping of several thousand markers and the reporting of a number of HIV-related, potentially actionable variants. Specifically, the panel of genetic tests addresses some recognized issues in HIV care: abnormal drug levels and toxicity, HIV and treatment associated metabolic disorders, co-infection HIV-HCV and prediction of disease progression. For example, tests included deterministic information (e.g., HLA-B*57 and abacavir hypersensitivity), as well as informative results on particular predispositions (e.g., metabolic risk). The language was controlled to indicate this difference. By design, this was not a randomized analysis of the impact of the specific genetic tests on clinical care. In particular, there was no opportunity to modify treatment choice and no intervention based on the report. Instead, this study defined procedures and strategies that are informative of the steps leading to the secure use of genetic information in clinical practice. It provides the basis for future randomized studies aimed at delivering actionable genetic results in real-time. Importantly, the proposed framework could be easily used to incorporate also demographic, behavioral, and laboratory parameters to more precisely estimate a patient's risk of a particular outcome (e.g. cardiovascular risk; a significant issue in the care of HIV infected individuals [171]) as described in Section 4.4.5.

Upon receipt of the interpreted report, physicians were queried as to their impressions of the information content of the report and their attitude toward the proposed privacy-preserving model for genetic testing. The overall response helped us to derive a more general understanding about the key requirements for the adoption of privacy-enhancing technologies in a routine clinical context.

The first important take-home message is that new systems based on sophisticated privacy-enhancing technologies, such as homomorphic encryption, that can perform analytics on genomic data and provide results without revealing sensitive information have the potential to be useful. Despite their inherent complexity, tools as the one evaluated in our survey are seen by a large majority of the physicians responding to the questionnaire as important enablers for the implementation of genomic-based medicine.

As expected, our results show that more than two thirds of the physicians testing the system acknowledged that protecting patient's genetic privacy is crucial when dealing with genetic data both in a clinical and research environment. Privacy-preserving systems such as ours have the potential to improve the management of patients' sensitive data, not only because they provide strong privacy and security guarantees, but also because they can protect physicians from undesirable liability issues (e.g., in the case of incidental findings not reported to patients) by limiting their access only to the necessary information.

Our confirmatory factor analysis indicates that, in addition to the ability of ensuring privacy and security of genetic data, the key requirement for the success and deployment of this new kind of medical information systems resides in the ability to make them usable and understandable by end-users and, most importantly, in the clinical value of the information made accessible to physicians. Our study shows that, despite the fundamental tension between data privacy and data access, an acceptable trade-off can often be found. Indeed, providing only the necessary information according the principle of data minimization, as described in the international good practices for genomic data management and in the data protection laws of Switzerland and Europe [3, 80, 92] , is largely accepted by physicians for this kind of medical applications (e.g., genetic testing) if such information can be clinically useful and actionable.

4.10 Summary

Privacy protection of sensitive medical data and particularly genetic data is an important concern in clinical care as the healthcare sector is increasingly suffering from data breaches. In this chapter, we have described the first privacy-preserving system based on homomorphic encryption that was successfully deployed and tested in a real-life clinical setting. The proposed system enables the secured storage and analysis of large-scale genetic data at an unstrusted storage and processing unit, as well as the targeted delivery of specific subsets of test results to the clinic [144] in full compliance with the principles of "data minimization" and "privacy by design" imposed by international data protection regulations (i.e., US HIPAA [202] and EU GDPR [80]). This will become increasingly important as many large-scale sequencing efforts are initiated with the goal of incorporating the resulting genomic data in clinical care.

Moreover, we have studied for the first time physicians' attitude toward the adoption of this new type of privacy-preserving models for using genetic data in the clinic, by evaluating the feedback of 38 HIV specialists who tested the proposed system. We have derived unique insights that will guide the design and facilitate the adoption of these systems in the future, by better addressing physicians' requirements. In particular, we have seen that, despite the general skepticisms around the apparent unpracticality and complexity of these technologies, systems based on cutting-edge privacy-enhancing techniques can be made efficient, deployable and usable in real operation settings. As physicians' primary scope is patients' health, the inherent complexity of these tools should be concealed as much as possible in order to facilitate physicians' work and avoid undesirable overheads.

Finally, we believe that this study, although limited to a small sample and specific use case, represents a first step towards the full awareness among physicians, policy makers, hospital administrators and the general public about the existence of sophisticated PETs that can play a significant role in mitigating the incidence of data breaches without preventing the use of the data.

Part II

Protecting Medical and Genomic Privacy in Clinical Research

Chapter 5

Privacy-Preserving Exploration of Genetic Cohorts in Untrusted Environments

In the first part of this thesis, we have addressed the problem of using genomic and clinical data for clinical care in a privacy-preserving way. Yet, the ability to re-use these data also for medical research is crucial in order to fully realize the promise of personalized medicine. In this chapter, we show how a medical institution can enable researchers to use, in a privacy-preserving way, its clinical and genomic data for feasibility studies by securely outsourcing the storage of the data to a centralized (and potentially untrusted) repository as, e.g., a public cloud.

5.1 Introduction

Re-use of patients' electronic health records (EHRs) can provide tremendous benefits for clinical research. One of the first essential steps for many research studies, such as clinical trials or population health studies, is to effectively identify, from EHR systems, groups of well-characterized patients who meet specific inclusion and exclusion criteria. This procedure is called cohort exploration or feasibility study. Yet, because nowadays clinical and omics data are stored across a variety of systems within the same medical institution, getting the information that is needed into the hands of researchers often requires substantial time and resources.

To address this problem and foster clinical research, many institutions have started to integrate clinical and genomic information from multiple sources into centralized data warehouses. These data warehouses store de-identified information on patients, such as EHR data, lab results, genetic data, demographic information and, because of security and privacy reasons, they are usually located in "militarized" (or trusted) environments behind the institution's firewall. Indeed, in these environments, all incoming connections are blocked and only a very limited number of employees have the right to access the data stored. This prevents researchers from easily accessing the data necessary for identifying new predictive biomarkers and rapidly finding subjects with similar clinical and omics characteristics. Therefore, in order to facilitate the use of this vast amount of data, it is common practice to create domain-specific "data marts" (i.e., subsets of the data warehouse) and outsource them to less protected environments, outside the militarized zone of the institution (e.g., a server within the research network of the institution or a server on a public cloud), that allow for incoming connections and can be easily accessed by researchers.

Yet, when it comes to outsource data marts of sensitive genomic and clinical data to unprotected environments, privacy and security concerns represent major obstacles that make the process extremely lengthy if not impossible.

For this reason, in the last few years, researchers from both the computer science and medical fields have started collaborating to design new advanced solutions that could enable the outsource of sensitive data while protecting individuals' medical privacy and, in particular, genomic privacy [79, 148]. However, to obtain acceptance and to be adopted in the real world, these solutions need to be deployed and assessed in concrete operational scenarios.

In this chapter, we describe how we effectively addressed this challenge. In particular, in collaboration with the Lausanne University Hospital (CHUV), we developed a new privacy-preserving solution that makes use of advanced privacy-enhancing technologies such as differential privacy and homomorphic encryption (so far believed unpractical) to enable the outsource and exploration of large amounts of genomic and clinical data. The proposed solution is – to the best of our knowledge – the first of its kind to be deployed and tested in a real operational environment.

Indeed, in order to be used in the real world, we built our system on top of the most widespread open-source framework for exploring clinical research data-warehouses, namely *Informatics for Integrating Biology and the Bedside* (i2b2) [146]. The i2b2 framework was jointly developed by the Harvard Medical School and Massachusetts Institute of Technology to enable clinical researchers to use existing de-identified clinical data and only IRB-approved genomic data for discovery research and design of target therapies. i2b2 is currently used by more than 200 medical institutions worldwide for translational research or academic purposes [4] and its software is maintained and constantly upgraded by the i2b2 Foundation. Yet, as most cohort explorers, i2b2 is designed to be primarily used in trusted environments as it does not provide any specific protection mechanism for genomic data, or other types of identifiable information, apart from standard access control and data de-identification [145], both proven to be ineffective [107, 181, 99, 112, 135, 210, 220]. This limitation substantially restricts the scope of potential feasibility studies that could be conducted in less controlled and protected environments for accelerating medical research.

In our solution, patients' genetic data are homomorphically encrypted and stored in a centralized i2b2 server along with pseudonymized and de-identified clinical data. Thanks to homomorphic encryption and as long as the decryption key is secret, such a server can be located in an untrusted environment as the confidentiality of the data is always ensured. Moreover, the use of homomorphic encryption not only guarantees confidentiality at rest but also during computation, hence enabling for privacy-preserving queries on the data (i.e., without ever decrypting the original genomic data) through the standard i2b2 Web client. With our solution, researchers can obtain the aggregate total number of patients (or other summary statistics such as allele frequency) who meet a given set of inclusion and exclusion criteria that also include genomic or other identifiable

Notation	Description
VCF	Variant call format
DWH-RC	Data warehouse for clinical research
REF	Reference allele
ALT	Alternate allele
CHR	Chromosome
POS	Position
$\|A\|$	Cardinality of set A
'|'	Separator used in VCF files for phased genotypes
'/'	Separator used in VCF files for unphased genotypes
$g1_j$	j-th polynomial storing the first value of the unphased genotype encoding for a set of variants
$g2_j$	j-th polynomial storing the second value of the unphased genotype encoding for a set of variants
nc_j	j-th polynomial storing the no-calls value for a set of variants
p	Public key for the FV encryption scheme
s	Secret key for the FV encryption scheme
ϵ_{tot}	Total privacy budget per user assigned by i2b2 administrators
ϵ_i	Privacy budget for query i
n	Number of patients in the database
$\oplus$	Symbol denoting the homomorphic addition operation
P	Set of patients satisfying the clinical predicate specified in the query
V	Set of variants satisfying the range specified in the query
G_P	Set of encrypted genotypes for patients in P and variants in V
N_P	Set of the no-calls values for patients in P and variants in V

Table 5.1: Notation used throughout the chapter.

features. According to their different access rights, researchers can receive either slightly perturbed (with noise satisfying the notion of differential privacy) – but still useful – or unperturbed query results.

We extensively tested the performance of our solution in a real operational setting for different cohort sizes, and we found the overhead introduced by privacy-preserving techniques to be entirely acceptable thanks to the ciphertext packing technique enabled by the *somewhat* homomorphic encryption scheme on which our solution is based. The response time is linear in the number of selected patients and always in the order of a few seconds for standard queries and cohort sizes, which outperforms the state of the art [119, 56, 118, 216, 212].

5.2 Preliminaries

In this section, we briefly summarize the main concepts in cryptography and genomics that are used in the chapter, as a complement to those we have seen in Chapter 2. We summarize the notation used in this chapter in Table 6.1. We denote x uniformly chosen from the set X as $x \overset{U}{\leftarrow} X$. Moreover, we use boldface letters to represent vectors, regular letters for polynomials and capitalized letters for sets.

5.2.1 Cryptographic Background

In this section, we briefly introduce the additional cryptographic concepts used in this chapter.

Fan and Vercauteren Cryptosystem. The proposed solution is based on the Fan and Vercauteren (FV) cryptosystem [82] which is the state-of-the-art lattice-based leveled homomorphic encryption scheme based on the *Ring Learning With Errors* (RLWE) problem. The FV scheme ensures indistinguishability against chosen plaintext attacks if the standard RLWE problem is hard. Moreover, as other lattice-based cryptosystems, it is supposed to be quantum-resistant. Let ℓ be a power of 2 and a polynomial degree, q be a coefficient modulus, t be a plaintext modulus, $\mathcal{X}$ be a noise distribution over a polynomial ring $\mathbb{Z}_q[x]/(x^\ell+1)$, and $m \in \mathbb{Z}_t[x]/(x^\ell+1)$ be a plaintext polynomial. Let $s \overset{U}{\leftarrow} \mathbb{Z}_q[x]/(x^\ell+1)$ be the secret key and $p = (p_0, p_1) = (-(a \cdot s + e) \bmod q, a)$, be the public key where $e \leftarrow \mathcal{X}$ and $a \overset{U}{\leftarrow} \mathbb{Z}_q[x]/(x^\ell+1)$. Then the FV scheme works as follows:

- *Encryption* (with $u, e_1, e_2 \leftarrow \mathcal{X}$):

$$\begin{aligned} &Enc(m,p) = \boldsymbol{c} = (c_0, c_1) = \\ &((p_0 \cdot u + e_1 + \lfloor \frac{q}{t} \rfloor \cdot m) \bmod q, (p_1 \cdot u + e_2) \bmod q), \end{aligned} \tag{5.1}$$

- *Decryption:*

$$Dec(\boldsymbol{c}, s) = \left\lfloor \frac{t}{q} \cdot ((c_0 + c_1 \cdot s) \bmod q) \right\rceil \bmod t, \tag{5.2}$$

- *Homomorphic addition:*

$$Add(\boldsymbol{c}, \boldsymbol{c'}) = ((c_0 + c_0') \bmod q, (c_1 + c_1') \bmod q). \tag{5.3}$$

We do not report the definition of homomorphic multiplication as it is not used in our solution. For further details we refer the reader to the original paper [82]. Note that we chose the FV scheme because, to the best of our knowledge, it provides the best performance in terms of homomorphic computations and storage overhead for the operations required in the proposed solution.

Ciphertext Packing. Ciphertext packing [49] is a technique that can be used to reduce the overall size of the ciphertext and improve the efficiency of homomorphic operations. Despite recent advances, practical HE is still quite expensive. This is because security considerations require ciphertexts to be large, thus slowing down homomorphic computations. Ciphertext packing represents the main technique for dealing with this problem as a vector of plaintext values, and not a single value, can be encrypted in only one ciphertext. Homomorphic operations are applied to these vectors component-wise.

More formally, let $\mathsf{CS}(K_p, K_s, P, C, \mathcal{E}, \mathcal{D})$ be a cryptosystem, and m_0, m_1, ..., be the messages to be encrypted where $m_i \in M$, $\forall i$. Let also $n = \left\lfloor \frac{|P|}{|M|} \right\rfloor$ and P_1, P_2, ..., P_n be n independent subspaces of P where $|P_j| \geq |M|$, $\forall j$. When $|P| \geq 2 \cdot |M|$, we can encrypt at most n messages into one ciphertext by encrypting $m' = m_1p_1 + m_2p_2 + \cdots + m_np_n$, where p_j is the basis of the subspace P_j.

For example, when $P = \mathbb{Z}_q[x]/(x^\ell+1)$ and $M = \mathbb{Z}_t$, we can encrypt at most ℓ messages into one ciphertext with $m' = m_0 + m_1x + \cdots + m_{\ell-1}x^{\ell-1}$.

Differential Privacy. Differential privacy is an approach to privacy-preserving reporting of results, introduced by Cynthia Dwork [74], that guarantees that a given randomized statistic, $f(D) = R$, computed on a dataset D_1 behaves almost the same when computed on the neighbor dataset D_2 that differs from D_1 in exactly one element. More formally we have that

$$\Pr\left[f(D_1) = R_0\right] \leq \exp(\epsilon) \cdot \Pr\left[f(D_2) = R_0\right], \tag{5.4}$$

where the parameter ϵ is a privacy parameter: the closer it is to 0 the more privacy is ensured. The most straightforward method [75] for achieving ϵ-differential privacy consists in perturbing the output of the statistic with noise drawn from the Laplace distribution with mean 0 and scale $\frac{\Delta f}{\epsilon}$, where Δf is known as the *sensitivity* of f:

$$\Delta f = \max_{D_1, D_2} ||f(D_1) - f(D_2)||_1. \tag{5.5}$$

Differently from *k-anonymity*, differential privacy guarantees privacy against an adversary regardless of his prior knowledge.

5.2.2 Genomic Background

In this section, we briefly introduce the additional genomic concepts used in this chapter.

Variant Call Format (VCF). The Variant Call Format (VCF) [11] is the main format for storing genetic variants of one or more individuals with respect to the reference genome. The VCF consists of two parts: header and content. The header contains the meta-information about the file and data along with the definition of file variables. The content holds the information about the genetic variants for a set of individuals. Each variant is uniquely identified by (i) its chromosomal position (CHR, POS), (ii) the reference allele (REF), and (iii) the alternate allele (ALT). Each line of the content corresponds to the information about a single variant and the genotype (i.e., the value) of this variant for each individual in the VCF. We call the information about the variant, such as CHR, POS, REF, ALT, meta-data. Meta-data is not privacy-sensitive as it is public information, as opposed to genotype information that is sensitive and must be protected.

In the VCF file, a genotype is represented by two numbers separated by either '|' or '/'. When it is separated by '|', the genotype is phased (i.e, we know which of the two chromosomes holds which allele). Whereas, when it is separated by /, the genotype is unphased (i.e., there is no information on which chromosome holds which allele). Each number represents the allele value. When it is 0, it means that the allele value is equal to the reference allele. When it is 1, it means the allele value is equal to the alternate allele. When the allele has not been genotyped correctly and there is no information about its value, we put '.' instead of any number. Such an event is named *no-call*.

In our solution, we assume that the VCF file was processed in such a way that entries with multiple alternate alleles were separated in several lines, with one allele per line.

5.3 System and Threat Models

In this section, we introduce the system and threat models inspired by CHUV's infrastructure. We then outline the functional requirements that our solution should satisfy and finally we describe our proposed privacy-preserving solution in detail. Note that these system and threat models can be easily adapted to other similar healthcare providers willing to outsource the storage of their sensitive clinical and genomic data to an untrusted party (e.g., a public cloud).

5.3.1 System Model

The CHUV's information system consists of two physically separated networks or environments, as depicted in Figure 5.1: (i) the main network of the hospital, also called *clinical network* and (ii) a *research network* that is shared with the University of Lausanne (UNIL). Each of these two networks hosts different services.

- *Clinical network.* The clinical network is a trusted environment and is used for hospital's clinical daily activities. It hosts all services used for daily healthcare and administration purposes along with the clinical research data-warehouse[1] (DWH-RC) that contains unencrypted pseudonymized clinical and genomic data of patients. This network is very controlled and protected by a firewall that blocks **all** incoming network traffic. Authorized users are authenticated and their activities are constantly logged.

- *Research network.* The research network is also protected by a firewall that blocks unauthorized incoming network traffic (e.g., the firewall does not block the traffic coming from the clinical network) but the level of control is weaker with respect to the clinical network as users' activities are not logged. This network hosts multiple isolated data marts (i.e., specialized subset of the data extracted from the DWH-RC) used by clinical or academic researchers in their research activities; i2b2 is one of these services. The i2b2 service consists of an i2b2 server and a database to which pseudonymized and de-identified clinical and genomic data are pushed from the DWH-RC so that authorized researchers can efficiently explore it to conduct feasibility studies. Researchers already in the network can access the i2b2 service after authentication through an internal Web-client.

The main purpose of this type of IT architecture is to isolate data that is used for clinical care and that is accessible only to a few trusted and authorized individuals from data used for research activities that can be accessed by several researchers through less restrictive authorization and authentication procedures. We note that all communications are protected through encryption.

5.3.2 Threat Model

In the above-mentioned system model, we consider two types of potential attackers: (i) a *honest-but-curious* (or *semi-honest*) adversary at the i2b2 service who honestly follows the protocol but tries to passively infer some private information about the patients, and (ii) a *malicious-but-covert* adversary who wants to re-identify a patient by performing multiple malicious, but legitimate, queries to the i2b2 service. We consider the DWH-RC as a trusted party as it is the initial source of the data.

[1] The detailed description of the clinical network is out of the scope of this work.

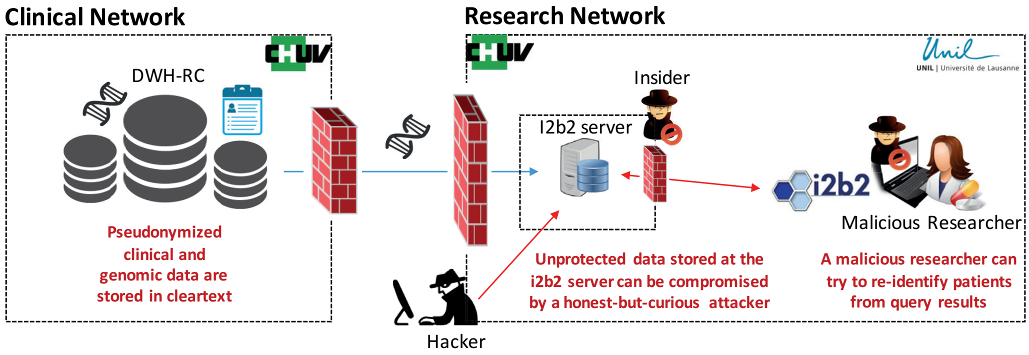

Clinical Network
Research Network
CHUV
CHUV
UNIL | Université de Lausanne
DWH-RC
Insider
I2b2 server
i2b2
Malicious Researcher
Pseudonymized clinical and genomic data are stored in cleartext
Unprotected data stored at the i2b2 server can be compromised by a honest-but-curious attacker
A malicious researcher can try to re-identify patients from query results
Hacker

Figure 5.1: System and threat models.

- The honest-but-curious attacker can be represented by a careless or disgruntled employee of the hospital (i.e., an insider) or a hacker who has illegitimate access to the i2b2 service and tries to obtain patients' private genomic and clinical information without altering the protocol. Note that although this information is pseudonymized and there is no direct link with patients' identities, re-identification would still be possible due to the identifying nature of the genome and to some auxiliary (and often publicly available) information (e.g., public genomic databases, recreational websites, online social networks, etc.) that the attacker might exploit [107, 181, 99, 112, 135, 210, 220]. As a consequence, a potential loss of clinical and genomic data could be extremely dangerous, not only for the patients but also for the reputation of the medical institution itself. Re-identified health-care records are nowadays extremely valuable for hackers as, according to a recent report by IBM [177], their value on the black market is as much as 60 times more than that of stolen credit cards.

- The malicious-but-covert adversary can be represented by a malicious but legitimate user of i2b2 (e.g., a malicious researcher) or hacker who breaks into the research network and uses the i2b2 service to re-identify an individual in a subset of patients with specific clinical characteristics. In particular, an attacker with already some genomic information about the victim (e.g., the value of some of her genetic variants) might repeatedly query the i2b2 service with this genomic information and use the system as an oracle. As such, he could re-identify the presence of the victim in a sensitive subset of individuals (e.g., all cancer patients or all HIV-positive patients, etc.) and infer his/her health status. For example, the attacker could exploit the aggregate information obtained from the cohort explorer as described in well-known attacks such as Homer's attack [107] and the Beacon attack [181].

Therefore, with these adversarial models, there are two potential privacy threats that we need to address with our proposed solution: (i) loss of patients' health data confidentiality due to illegitimate data access and (ii) patients' re-identification and resulting sensitive attribute disclosure from legitimate data access. Data confidentiality can be protected at rest and during processing by using homomorphic encryption, whereas the re-identification risk can be mitigated by perturbing the query result in order to satisfy the notion of differential privacy.

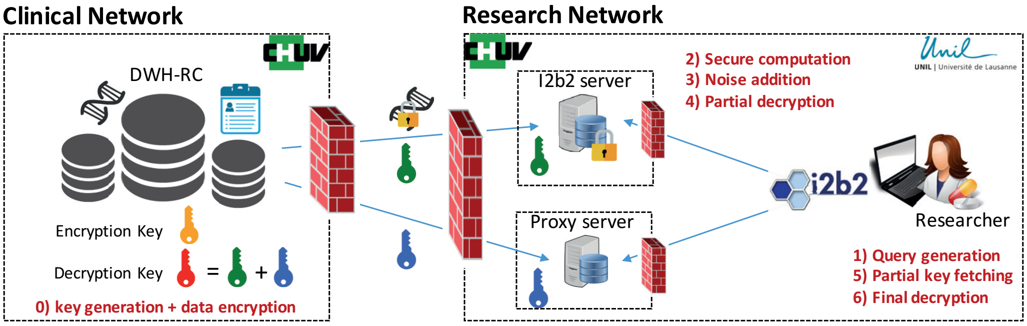


Figure 5.2: Architecture of the proposed solution.

5.4 Functional and Computational Requirements

The functional requirements of our privacy-preserving cohort explorer should be based on other well-known tools for exploration of genetic cohorts such as the Beacon system of Global Alliance for Genomics & Health (GA4GH) [87] and the ExAC browser of the Broad Institute [2]. For example, through the Beacon system researchers can query a database of genomes for the presence of a specific mutation, whereas researchers using the ExAC browser can also have information about the alternate allele count and frequency of the queried mutation.

As such, a user of our system should be able to obtain for all genetic variants in a selected chromosomal range:

- Reference/alternate allele frequencies
- Number of mutated genotypes (i.e., genotypes composed by at least one alternate allele)
- Number/frequency of genotypes that are homozygous with respect to the reference/alternate allele
- Number/frequency of genotypes that are heterozygous (with or without phase information)

Moreover, the storage and computational overhead introduced by the proposed solution should be kept at the lowest possible level in order to provide the same smooth user-experience as in the unprotected i2b2.

5.5 Proposed Solution

The architecture of the proposed solution is depicted in Figure 5.2. In order to protect the confidentiality of the data stored at the i2b2 server from a honest-but-curious adversary we introduce a new component to the i2b2 service, namely a *proxy* server, whose role is to support the decryption phase and provide the storage of partial decryption keys. Although part of the same service, the i2b2 server and the proxy server are physically separated from each other, protected by different firewalls and equipped with an intrusion detection system. Both servers cannot communicate with each other and we assume that they do not collude or, in other words, that they are not simultaneously compromised.

This assumption is reasonable in practice as in the CHUV research network the two servers are located in two distinct physical locations and are administered by different administrators. Ideally, to further minimize the risk of collusion, the proxy server should be hosted by another independent institution.

Based on this architecture, the privacy-preserving exploration of genomic cohorts consists of three main phases: (i) a *system initialization* phase where cryptographic keys are generated and the genetic variants in the VCF file are encoded, encrypted and pushed to the i2b2 server along with de-identified clinical data for secure storage, (ii) a *user assignment* phase, where access rights and cryptographic keys are assigned to each new user in the system and (iii) a *query execution* phase where the user builds a new query that is then sent to the i2b2 server and processed in a privacy-preserving way, i.e., without ever decrypting the original data. The query result is then decrypted by the user via the i2b2 Web-client.

5.5.1 System Initialization Phase

The system initialization phase takes place at the clinical research data-warehouse (DWH-RC) where clinical and genomic data are stored with patients' pseudonyms and can be accessed only by a group of a few trusted and authorized individuals. The first step consists in setting the parameters of the FV cryptosystem (l, t and q) used to encrypt the genomic data according to the desired security level (e.g., 80 bits security) and the maximum number of additions and multiplications to be supported by the system. In our case, the maximum number of additions should be at least twice the number of individuals in the database because it corresponds to the maximum number of alleles that could be involved in a counting query. Multiplication is not needed in this specific use case but could be used for other more complex applications. For the optimal selection of the FV parameters, we refer the reader to the original work by Fan and Vercauteren [82]. Then, a public key p and a secret key s are generated as described in Section 5.2.1.

After key generation, the VCF file is parsed and each genotype is encoded following the scheme described either in Table 5.2 (for phased genotypes) or in Table 5.3 (for unphased genotypes). For simplicity, in the rest of the paper we describe only the encoding for unphased genotypes described in Table 5.3 (but the encoding in Table 5.2 is equally supported by the proposed solution). As the number of possible genotype values is 6 (we are also including *no-calls* as they need to be used when allele or genotype frequencies are computed), we use three values for genotype encoding: the first two values indicate the presence of zero, one or two alternate alleles, whereas the third value reports the number of no-calls in the genotype.

For each individual in the VCF file, consecutive genotypes are packed into 3 sets of polynomials by using the packing technique described in Section 5.2.1. Let $(\text{gt1}_i, \text{gt2}_i, \text{no-call}_i)$ be the encoded genotype for variant i. Then, we can pack at most ℓ

Genotype value	**Genotype encoding**			
	gt1	gt2	gt3	no-call
.|.	0	0	0	2
.|0	0	0	0	1
.|1	1	0	0	1
0|.	0	0	1	1
1|.	0	1	0	1
0|0	0	0	0	0
0|1	1	0	0	0
1|0	0	1	0	0
1|1	0	0	1	0

Table 5.2: Encoding for phased genotypes.

Genotype value	**Genotype encoding**		
	gt1	gt2	no-call
./.	0	0	2
./0	0	1	1
./1	1	0	1
0/0	0	0	0
0/1	1	0	0
1/1	0	1	0

Table 5.3: Encoding for unphased genotypes

genotypes in three polynomials

$$g1_j = \sum_{k=0}^{\ell-1} \mathrm{gt1}_{j\ell+k} x^k, \tag{5.6}$$

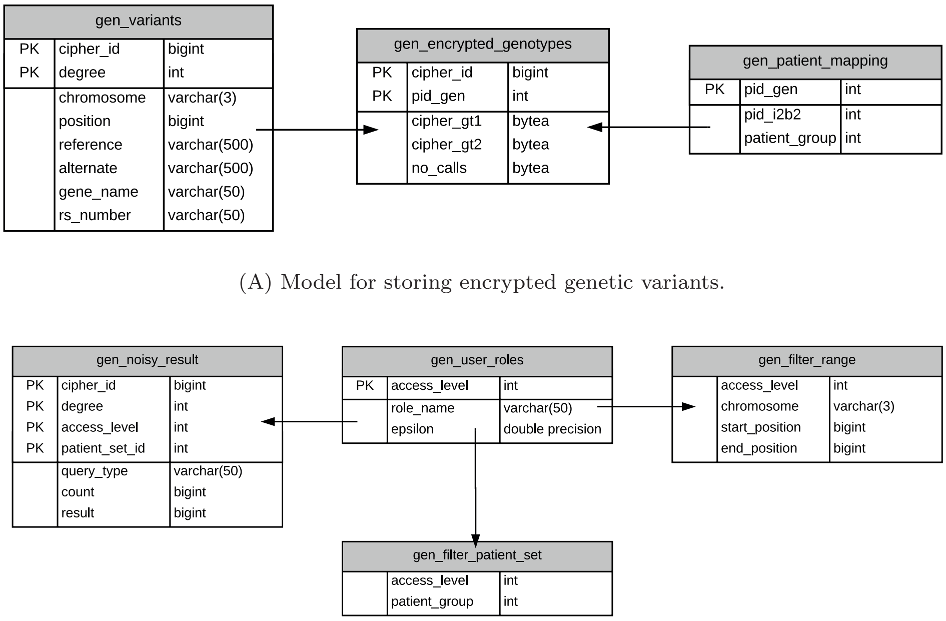
$$g2_j = \sum_{k=0}^{\ell-1} \mathrm{gt2}_{j\ell+k} x^k, \tag{5.7}$$

$$nc_j = \sum_{k=0}^{\ell-1} \text{no-call}_{j\ell+k} x^k, \tag{5.8}$$

where ℓ is the polynomial degree and j the ciphertext index. Polynomials $g1_j$ and $g2_j$ are finally encrypted with the public key p at the DWH-RC to obtain $\mathbf{c1_j} = Enc(g1_j, p)$ and $\mathbf{c2_j} = Enc(g2_j, p)$ which are pushed along with nc_j, patients' pseudonyms, de-identified clinical data, and public key p to the i2b2 server for storage according to the data model shown in Figure 5.3A. Note that nc_j does not need to be encrypted as it does not leak any information on the value of the genotype.

5.5.2 User Assignment Phase

Assignment of a new user also takes place at the DWH-RC. During this phase, for each new user, the secret key s is randomly divided into two shares s_1 and s_2 such that $s = s_1 + s_2$. The two partial secret keys s_1 and s_2 are then sent, for storage, to the i2b2 server and proxy server, respectively. This procedure avoids a single point of failure in



gen_variants		
PK	cipher_id	bigint
PK	degree	int
	chromosome	varchar(3)
	position	bigint
	reference	varchar(500)
	alternate	varchar(500)
	gene_name	varchar(50)
	rs_number	varchar(50)

gen_encrypted_genotypes		
PK	cipher_id	bigint
PK	pid_gen	int
	cipher_gt1	bytea
	cipher_gt2	bytea
	no_calls	bytea

gen_patient_mapping		
PK	pid_gen	int
	pid_i2b2	int
	patient_group	int

(A) Model for storing encrypted genetic variants.

gen_noisy_result		
PK	cipher_id	bigint
PK	degree	int
PK	access_level	int
PK	patient_set_id	int
	query_type	varchar(50)
	count	bigint
	result	bigint

gen_user_roles		
PK	access_level	int
	role_name	varchar(50)
	epsilon	double precision

gen_filter_range		
	access_level	int
	chromosome	varchar(3)
	start_position	bigint
	end_position	bigint

gen_filter_patient_set		
	access_level	int
	patient_group	int

(B) Model for storing users' access rights.

Figure 5.3: Data model used in the proposed solution.

the system: if one of the two servers is compromised, data are still protected because only with the full secret key s the attacker can successfully decrypt. Note that ideally, to ensure better security, the i2b2 server and the proxy server should be part of two completely different organizations. Yet, because of the CHUV's infrastructure these two servers are within the same network. Nevertheless, they are physically separated, managed by different administrators and cannot communicate with each other.

Access rights for each new user are also assigned during this phase and stored on the i2b2 server, as shown in Figure5.3B. Different users might have different rights to access patients' private information. Our system provides full customization on three different levels of access: (i) type of query result, (ii) set of accessible patients, and (iii) set of accessible genetic variants. For example, a junior researcher might have access only to perturbed query results, where some noise has been added on the true result to satisfy the notion of differential privacy, whereas a senior researcher can obtain accurate information. Also, depending on his specialization or on the IRB-approved study protocol, a researcher might have access only to a given subset of patients or a given subset of genetic variants. For example, an oncologist might have access only to data of patients with cancer and might not be allowed to query genetic variants related to other diseases such as diabetes or coronary artery disease.

We acknowledge that if the i2b2 server is compromised by a honest-but-curious adversary, information on users' access rights could leak because stored in cleartext. Yet, protecting users' access rights is not the focus of this project that addresses the protection of patients' data. Solutions based on *attribute-based somewhat homomorphic encryption (ABSHE)*[147] can be use on top of our system in order to thwart this attack.

5.5.3 Query Execution Phase

The query execution consists of five phases: (i) the query generation at the user-side and (ii) the query processing at the server-side, (iii) result perturbation at the server-side, (iv) result partial decryption at the server-side and (v) result decryption at the client-side.

(i) *Query generation.* In the query generation, after password-based authentication, the user of our privacy preserving cohort explorer (i.e., CHUV researcher) builds his query in two steps by using the i2b2 Web-client. In the first step, he selects a set of inclusion and exclusion clinical criteria in order to identify the set of patients for whom he wants to explore the genetic data. For example, a researcher might be interested in aggregate genetic information on some specific variants for patients with cancer who have undergone a specific treatment and who have had a positive outcome. In the second step, once the desired patient set has been identified, the user selects the set of variants of interest and the type of summary statistics he wants to obtain and finally submits the query.

(ii) *Query processing.* During the query processing, the i2b2 server verifies the legitimacy of the query and the access rights of the querier. If the verification is successful, the server retrieves from the database the list of ciphertexts containing the encrypted genotypes of the variants specified in the query. The ciphertexts are then homomorphically combined in order to compute the encrypted query result. We designed different secure algorithms for computing the different summary statistics described in Section 5.4. The details of these algorithms are explained in Section 5.5.5.

(iii) *Result perturbation.* Depending on the role and access level of the user, the i2b2 server can perturb the encrypted query result to satisfy the notion of differential privacy and prevent re-identification attacks. In particular, we assume users of the system hold a single account and do not collude. Moreover, users are assigned a total privacy budget ϵ_{tot} whose value is decided by i2b2 administrators in the user assignment phase and may depend on the user's role and level of trustworthiness. For each new query i from the same user, the i2b2 server draws independent noise from the Laplace distribution with mean 0 and scale $\frac{\Delta f}{\epsilon_i}$, where f is the requested aggregation function, encrypts it and homomorphically adds it to the encrypted query result in order to satisfy ϵ_{tot}-differential privacy. The user's privacy budget ϵ_{tot} is then reduced by ϵ_i and keeps decreasing every time a new query is answered; the i2b2 server will keep on providing query answers to the user until his budget runs out. The value of ϵ_i can be fixed, if set by the database administrator, or adaptive, if set by the user who may use a small value of ϵ_i and incur more noise for preliminary queries whose expected result is large and save the budget for more specific queries. What is the right value of ϵ_i is out of the scope of this chapter. We note that Δf is equal to 1 for count queries and to $\frac{1}{n}$ for predicate queries (i.e., queries asking the fraction of elements in a database that satisfy a specific predicate), where n is the number of patients in the database. For consecutive queries, Δf grows linearly.

(iv) *Result partial decryption.* After computation of perturbed/unperturbed encrypted query result, the i2b2 server partially decrypts it with the first part of the users' secret key s_1 (which is stored in the i2b2 database). In particular, from a ciphertext $\mathbf{c} = (c_0, c_1)$ we obtain, after partial decryption, a new ciphertext $\mathbf{c}' = (c_0', c_1')$, as described in Section 5.5.4. Finally, the i2b2 server sends back to the user the encrypted polynomial c_1' and the coefficients of the encrypted polynomial c_0' matching the variants specified by the query and user's access level. Note that, for the sake of information minimization, only the specified coefficients of the encrypted polynomial c_0' are sent by the server to the user. In other words, we do not want the user to obtain the summary statistics of all the variants packed in the same ciphertext, but only the ones he has requested access to.

(v) *Result decryption.* At the user-side, the Web-client fetches the second part of the user's secret key s_2 from the proxy server and performs the full decryption in order to obtain the plaintext final results of the query, as described in Section 5.5.4. For performance reasons, part of the full decryption (i.e., the polynomial multiplication $c_1' \cdot s_2$) could be run at the proxy server. Note that, by observing only s_2 and c_1', the proxy server cannot infer anything about the plaintext.

5.5.4 Partial Decryption With FV Scheme

As the Fan and Vercauteren (FV) secret key s has been split into two shares stored at the i2b2 server and at the proxy server respectively, decryption has to be done in two steps. The original FV cryptosystem does not have a partial decryption algorithm. Here we describe how this additional feature can be easily achieved.

Definition 5.1 (Partial Key Generation). *Let q be the ciphertext modulus, ℓ be the polynomial degree and $s \in \mathbb{Z}_q[x]/(x^\ell + 1)$ be the secret key [82]. Then, the partial key set (s_0, s_1) can be generated as $s_0 \overset{U}{\leftarrow} \mathbb{Z}_q[x]/(x^\ell + 1)$ and $s_1 = s - s_0$.*

Definition 5.2 (Partial Decryption). *Let q be the modulus, ℓ be the polynomial degree, t be the plaintext modulus, (s_0, s_1) be the partial key set from Definition 5.1 and $\mathbf{c} = (c_0, c_1)$ be the ciphertext where $c_0, c_1 \in \mathbb{Z}_q[x]/(x^\ell + 1)$. Then, we can define the partial decryption and the full decryption as follows,*

$$\begin{aligned} Partial\_Dec(\boldsymbol{c}, s_0) &= \boldsymbol{c'} = (c_0', c_1') = (c_0 + c_1 \cdot s_0,\ c_1) \\ Full\_Dec(\boldsymbol{c'}, s_1) &= Dec(\boldsymbol{c'}, s_1), \end{aligned} \tag{5.9}$$

where $Full\_Dec(Partial\_Dec(\boldsymbol{c}, s_0), s_1) = Dec(\boldsymbol{c}, s)$.

Proof. We have

$$\begin{aligned} &Full\_Dec(Partial\_Dec(\boldsymbol{c}, s_0), s_1) \\ &\qquad = \left\lfloor \frac{t}{q}((c_0 + c_1 \cdot s_0 + c_1 \cdot s_1) \bmod q) \right\rceil \bmod t \\ &\qquad = \left\lfloor \frac{t}{q}((c_0 + c_1 \cdot s) \bmod q) \right\rceil \bmod t \\ &\qquad = Dec(\boldsymbol{c}, s) \end{aligned} \tag{5.10}$$

By Definition 5.1, $s_0 + s_1 = s$, so (5.10) holds. Thus, $Full\_Dec(Partial\_Dec(\boldsymbol{c}, s_0), s_1)$ is equivalent to $Dec(\boldsymbol{c}, s)$.

■

5.5.5 Secure Algorithms

In this Section, we describe the algorithms developed to securely compute the summary statistics outlined in Section 5.4. We use $\oplus$ to denote homomorphic addition. All secure algorithms take as input the set of patients P satisfying the clinical predicate specified in the query (e.g., patients with HIV), the set of variants V in the chromosomal range specified in the query and packed into the same ciphertext, the set of no calls N_P for all patients in P, and the set of encrypted polynomials G_P containing the genotypes of all patients in P. Note that G_P consists of two ciphertexts $G_P.\mathbf{c1}$ and $G_P.\mathbf{c2}$ representing the encryption of the first and second values of the unphased genotype encoding, respectively. The output of each algorithm is a map M containing the encrypted results per set of variants that have been computed together. Algorithms 6-10 enable, respectively, the secure computation of:

1. The reference/alternate allele frequency for variants in a specific range;
2. The number of mutated variants (i.e., with at least one alternate allele) in a specific range;
3. The number or frequency of homozygous-alternate/heterozygous genotypes for variants in a specific range;
4. The number/frequency of homozygous-reference genotypes for variants in a specific range.

Algorithm 6: Secure allele frequency

```
Input  : $P$, $V$, $N_P$ and $G_P$ for each patient in $P$.
Output: $M$
$M \leftarrow$ an empty set of key-value pairs $M.addEntry(V, (\mathsf{Enc}(0), 0))$
for $p \in P$ do
    ones $\leftarrow$ Set of variants in $V$ whose $N_p = 1$
    twos $\leftarrow$ Set of variants in $V$ whose $N_p = 2$
    $M' \leftarrow$ an empty map
    for $(key, value) \in M$ do
        if $key \cap twos \neq \varnothing$ then
            $M'.addEntry(key \cap \text{twos}, value)$          // Computation for genotype ./.
        end
        if $key \cap ones \neq \varnothing$ then
            $M'.addEntry(key \cap \text{ones}, (value[0] \oplus G_p.c1, value[1] + 1))$ // Computation for
                genotypes ./0 and ./1
        end
        if $key \setminus (ones \cup twos) \neq \varnothing$ then
            $M'.addEntry(key \setminus (\text{ones} \cup \text{twos}), (value[0] \oplus G_p.c1 \oplus 2 \cdot G_p.c2, value[1] + 2))$
                // Computation for genotypes 0/0, 0/1 and 1/1
        end
    end
    $M \leftarrow M'$
end
return $M$ // $value[0]/value[1]$ will be computed at the client side
```

Algorithm 7: Secure mutation count

```
Input  : $P$, $V$, $N_P$ and $G_P$ for each patient in $P$.
Output: $M$
$M \leftarrow$ an empty set of key-value pairs $M.addEntry(V, \mathsf{Enc}(0))$
for $p \in P$ do
    ones $\leftarrow$ Set of variants in $V$ whose $N_p = 1$
    $M' \leftarrow$ an empty map
    for $(key, value) \in M$ do
        if $key \cap ones \neq \varnothing$ then
            $M'.addEntry(key \cap \text{ones}, value \oplus G_p.cipher\_gt1)$ // Computation for genotype
              ./0 and ./1
        end
        if $key \setminus ones \neq \varnothing$ then
            $M'.addEntry(key \setminus \text{ones}, value \oplus G_p.cipher\_gt1 \oplus G_p.cipher\_gt2)$ // Computation
              for genotype ./., 0/0, 0/1 and 1/1
        end
    end
    $M \leftarrow M'$
end
return $M$
```

Algorithm 8: Secure homozygous-alternate/heterozygous frequency

```
Input  : $P$, $V$, $N_P$ and $G_P$ for each patient in $P$.
Output: $M$
$M \leftarrow$ an empty set of key-value pairs
$M.addEntry(V, (\mathsf{Enc}(0), 0))$
for $p \in P$ do
    zeros $\leftarrow$ Set of variants in $V$ whose $N_p = 0$
    $M' \leftarrow$ an empty map
    for $(key, value) \in M$ do
        if $key \setminus zeros \neq \varnothing$ then
            $M'.addEntry(key \setminus \text{zeros}, value)$ // Computation for genotype ./., ./0 and ./1
        end
        if $key \cap zeros \neq \varnothing$ then
            $M'.addEntry(key \cap \text{zeros}, (value[0] \oplus G_p.cipher\_gt2, value[1] + 1)$
                // Computation for genotype 0/0, 0/1 and 1/1
            // $cipher\_gt1$ instead of $cipher\_gt2$ for heterozygous
        end
    end
    $M \leftarrow M'$
end
return $M$ // $value[0]/value[1]$ will be computed at the client side
```

5.6 Security Analysis

In this section, we discuss about the security of our system with respect to the protection of genomic data. The protection of clinical data is not the focus of this paper. However, differently from genomic data, various anonymization techniques can be applied to protect clinical data and satisfy formal notions of privacy such as *k-anonymity* [189], *l-diversity* [134] or *t-closeness* [129]. Because using anonymization techniques could modify the original clinical data and reduce the overall utility of the system, our system can also be adapted to clinical data in case full accuracy is required.

Our system consists of four different parties: the data-warehouse (DWH), the i2b2 server (IS), the proxy server (PS) and the i2b2 user (U). As DWH is trusted, we only focus on IS, PS and U. As discussed in Section 5.3.2, we assume the honest-but-curious

Algorithm 9: Secure homozygous-alternate/heterozygous count

```
Input  : $P$, $V$, $N_P$ and $G_P$ for each patient in $P$.
Output: $M$
$M \leftarrow$ an empty set of key-value pairs
$M.addEntry(V, \mathsf{Enc}(0))$
for $p \in P$ do
    ones $\leftarrow$ Set of variants in $V$ whose $N_p = 1$
    $M' \leftarrow$ an empty map
    for $(key, value) \in M$ do
        if $key \cap ones \neq \varnothing$ then
            $M'.addEntry(key \cap \text{ones}, value)$    // Computation for genotype ./0 and ./1
        end
        if $key \setminus ones \neq \varnothing$ then
            $M'.addEntry(key \setminus \text{ones}, value \oplus G_p.cipher\_gt2)$
                // $cipher\_gt1$ instead of $cipher\_gt2$ for heterozygous
            // Computation for genotype ./., 0/0, 0/1 and 1/1
        end
    end
    $M \leftarrow M'$
end
return $M$
```

Algorithm 10: Secure homozygous-reference count or frequency

```
Input  : $P$, $V$, $N_P$ and $G_P$ for each patient in $P$.
Output: $M$
$M \leftarrow$ an empty set of key-value pairs
$M.addEntry(V, (\mathsf{Enc}(0), 0))$
for $p \in P$ do
    zeros $\leftarrow$ Set of variants in $V$ whose $N_p = 0$
    $M' \leftarrow$ an empty map
    for $(key, value) \in M$ do
        if $key \setminus zeros \neq \varnothing$ then
            $M'.addEntry(key \setminus \text{zeros}, value)$ // Computation for genotypes ./., ./0 and ./1
        end
        if $key \cap zeros \neq \varnothing$ then
            $M'.addEntry(key \cap \text{zeros}, (value[0] \oplus G_p.cipher\_gt1 \oplus G_p.cipher\_gt2, value[1] + 1))$
                // Computation for genotype 0/0, 0/1 and 1/1
        end
    end
    $M \leftarrow M'$
end
for $(key, value) \in M$ do
    $value[0] \leftarrow \mathsf{Enc}(value[1] + value[1]x + value[1]x^2 + \cdots + value[1]x^{\ell-1}) \oplus -value[0]$
end
return $M$ // $value[0]/value[1]$ will be computed at the client side for the homozygous
    reference frequency
```

adversarial model for both IS and PS and the malicious-but-covert model for U. Moreover, we recall that s_1 represents the partial secret key stored at IS and s_2 represents the partial secret key stored at PS for a specific user. Similarly, S_1 and S_2 represent the sets of partial keys for all users in the system at IS and PS, respectively.

- i2b2 Server: If the i2b2 server is compromised, the adversary can access the encrypted genomic data, the set of partial secret keys S_1, the role and access level of each user, the amount of noise used for perturbing query results, accessible chromosomal ranges, and the history of queries. Moreover, from the history of queries and the accessible chromosomal ranges, the attacker can infer users' access patterns, their potential interests and medical specialties. Yet, we note that our goal is to preserve the privacy of patients, not of i2b2 users. Techniques such as private information retrieval (PIR) can be used on top of ours for this purpose.

As such, although the adversary has encrypted genomic data and the set of partial secret keys S_1, he cannot obtain any sensitive genomic information about the patients as he still needs at least one key from the other set of partial keys S_2 in order to decrypt. However, because IS and PS cannot be simultaneously compromised by assumption, the attacker cannot obtain any partial key in S_2 from PS. In addition, the recovery of a partial secret key $s_2 \in S_2$ at PS is still hard even if numerous partial keys $s_1 \in S_1$ are known. The adversary has to perform approximately $O(2^l)$ operations where l is the polynomial degree. Hence, the sensitive genomic data remain secure if PS discards its set of partial keys S_2 as soon as it detects that IS is compromised. In this case, there is no need to re-encrypt genomic data with a new secret key as the full secret key is never revealed. Only new partial keys need to be regenerated for all users in order to avoid the decryption of leaked data with a later attack on PS.

- Proxy Server: If the proxy server is compromised, only the set of partial secret keys S_2 is leaked. As before, because PS and IS cannot collude, the attacker cannot obtain any sensitive genomic information. Also in this case, new partial keys need to be generated for all users in order to avoid the decryption of leaked data with a later attack on IS.

- User: If a user is compromised, his credentials can be stolen and used by a malicious-but-covert adversary. Then, the adversary can get any aggregated query result that is originally accessible by the user. Since the adversary can also deduce the identifier of the user's partial keys s_1 and s_2, he can get s_2 from PS by sending a polynomial 1 along with the partial key identifier. This problem could be easily addressed by adding some noise after the multiplication at PS. Yet, this additional protection mechanism is not necessary as PS simply stores s_2 of a user on his behalf and the adversary has no mean to reconstruct the full secret key. Hence, there is no leakage of sensitive genomic information even if a user can obtain s_2.

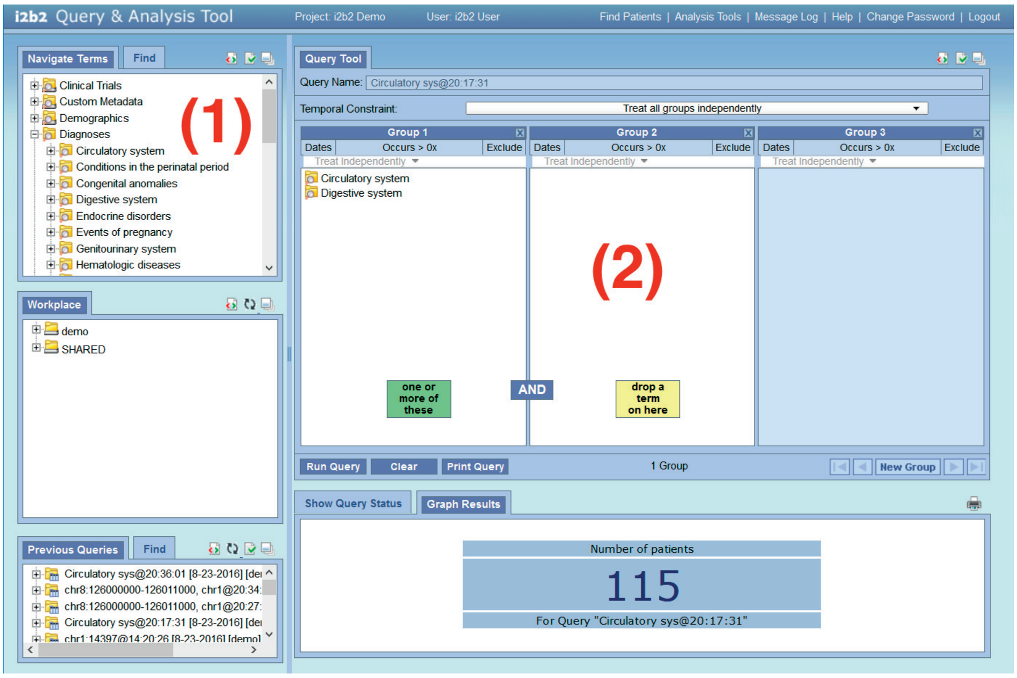
Yet, if the compromised user's role allows the adversary to obtain unperturbed query results, patients re-identification is still possible. An additional system independent from the user's privacy budget ϵ_{tot} should be put in place at IS to detect suspicious requests. We leave this investigation for future work.

5.7 Implementation and Performance Evaluation

In this section, we describe how we implemented and deployed our solution in the real operational setting of the Lausanne University Hospital (CHUV). We evaluate its performance on real genomic and clinical data for different database sizes and types of query.

5.7.1 Plugin Implementation

We implemented our privacy-preserving solution as a plugin of the i2b2 framework. The i2b2 architecture consists of two major components: The first is the back-end infras-



Query Name: Circulatory sys@20:17:31

Temporal Constraint: Treat all groups independently

Number of patients

115

For Query "Circulatory sys@20:17:31"

Figure 5.4: i2b2 native Web-client.

tructure (the “HIVe”) that is responsible for the security aspects, the access rights and for managing the underlying data repository. The second component is the user Web-client: a front-end application suite of query and mining tools that enables users to ask questions about patients’ data on the i2b2 server. The native version of i2b2 does not include any support neither for privacy-preserving data processing nor for managing genomic data but it is designed to enable cohort identification based mostly on clinical and demographic data. As shown in Figure 5.4, after logging in, a user can drag-and-drop search terms from the clinical ontology (1) into the Venn diagram-like interface (2) to construct his cohort of patients. The main advantage of i2b2 with respect to other existing platforms for clinical research resides in its modular design that enables adopters to easily extend the core architecture with independent plugins. As a consequence, we implemented our privacy-preserving plugin as a totally independent module that can be easily loaded during the setup of the standard i2b2. Our plugin is composed of four main parts: (i) a data importation tool, (ii) a back-end module for the i2b2 server, (iii) a back-end module for the proxy server, and (iv) a front-end module for the i2b2 native Web-client.

In the followings, we briefly describe each of these components.

- Data Importation Tool: The importation tool is written in C++ and is responsible for the system *initialization* and the new *user assignment*. For the system initialization, it takes as input the VCF file, the access rights policies and the FV public and secret keys. The tool parses the VCF file, encodes and packs the genotypes into polynomials and encrypts them by using the FV public key. The final output is a SQL script that can be used to import data in the i2b2 SQL database. For the new user assignment, the importation tool takes as input the new user’s identifier, his access level and the FV secret key. As output, it generates a SQL script to import into the i2b2 server the new user's access level along with the first part of the secret key and into the proxy server the second part of the secret key.

- i2b2 Server Module: The i2b2 server back-end module consists of two parts: the "main cell" and the "crypto engine". The main cell is written in Java and is part of the i2b2 server application. It is responsible for managing the data repository, handling the queries, computing homomorphic addition of ciphertexts, and, according to the user's access rights, adding noise on the computation result to satisfy the notion of differential privacy. The crypto engine is written in C++ and it is used for the partial decryption at server side. The Java Native Interface (JNI) is used to call function in the C++ crypto engine from the main cell.

- Proxy Server Module: The proxy server back-end module has a similar structure to the i2b2 server module. The main cell, written in Java, is responsible for managing users' partial keys stored in the data repository, whereas the crypto engine is responsible for helping the user with the full decryption by performing polynomial multiplication.

- Web-client Module: The Web-client front-end module is written in JavaScript and can be loaded from the native i2b2 Web-client. It consists of a *query builder* (Figure 5.5A), where the user can drag-and-drop a patient set (previously constructed through the native interface) and type into a search bar a set of genetic variants of interest, and a *result visualizer* (Figure 5.5B), where the user can visualize the results of his current and previous queries. The user is allowed to enter genetic variants by gene name, dbSNP identifier or chromosomal position and to select the summary statistic he is interested in.

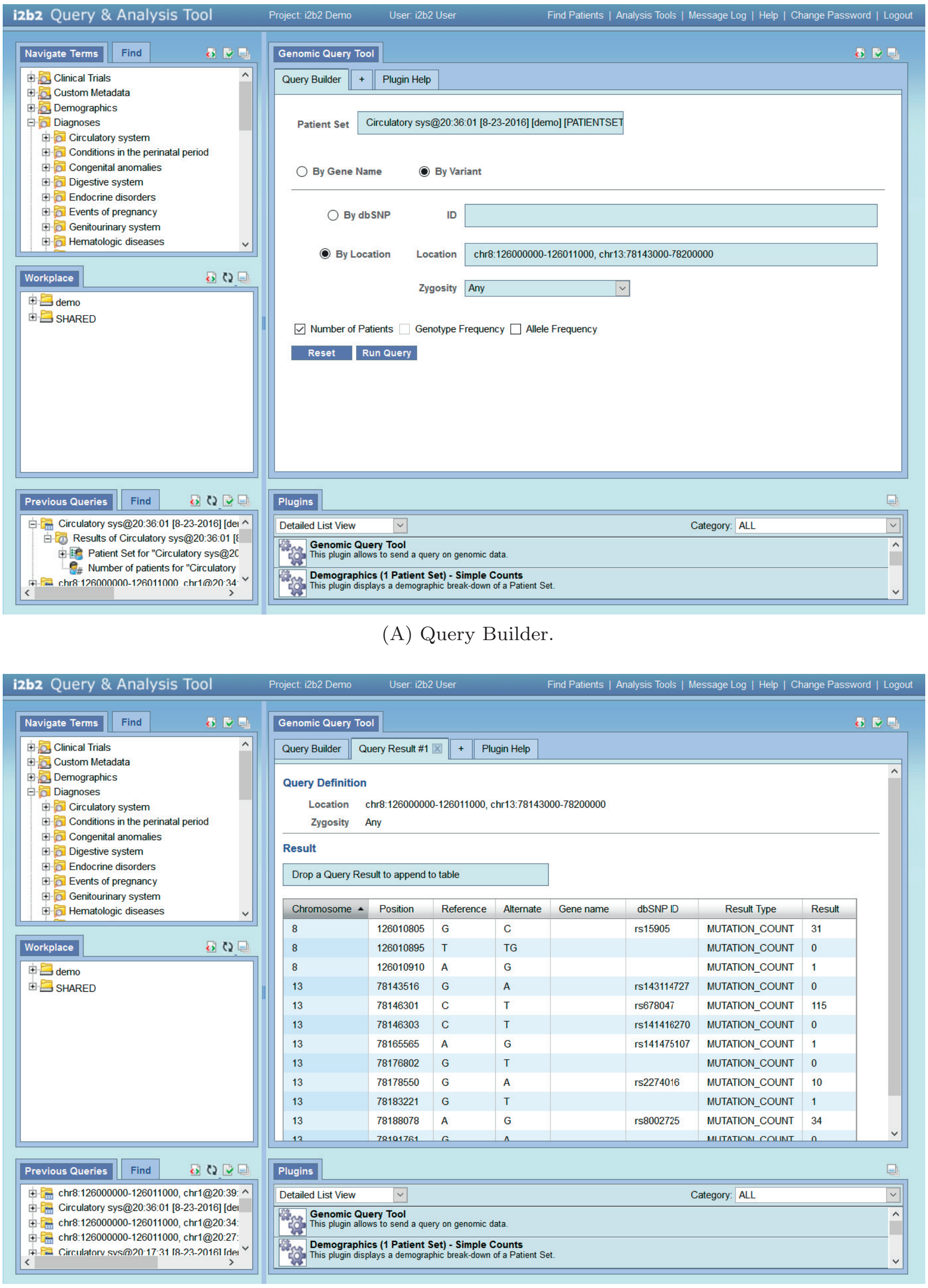
5.7.2 Performance Evaluation

To evaluate the performance of our proposed solution in the real operational setting of the Lausanne University Hospital, we have performed several tests on real cohorts of patients with clinical and genomic data.

- Experimental Setup: To show the practicality of our solution, we used only commodity hardware for our experiments. The i2b2 server module and the proxy server module run on two servers in the CHUV's research network. Their configurations are described in Table 5.4. For both servers, we limited the number of threads to 8. At the user side, we used an off-the-shelf laptop equipped with Windows 10, intel i7-3517U processor and 10 GB of memory. We ran the i2b2 Web-client on Firefox 47.0.1.

	Data warehouse i2b2 Server	Proxy server
Operating System	Ubuntu 14.04	Ubuntu 14.04
Processor	Intel Xeon E3-1270	Intel Core i7-620M
Memory	16 GB	4 GB
Max Memory for JVM	8 GB	512 MB
Database	PostgreSQL 9.4	MySQL 5.6

Table 5.4: Server Setting



Patient Set: Circulatory sys@20:36:01 [8-23-2016] [demo] [PATIENTSET

○ By Gene Name ● By Variant

○ By dbSNP ID

● By Location Location: chr8:126000000-126011000, chr13:78143000-78200000

Zygosity: Any

☑ Number of Patients □ Genotype Frequency □ Allele Frequency

Circulatory sys@20:36:01 [8-23-2016] [de
Results of Circulatory sys@20:36:01 [8
Patient Set for "Circulatory sys@20
Number of patients for "Circulatory
chr8:126000000-126011000, chr1@20:34

Genomic Query Tool
This plugin allows to send a query on genomic data.

Demographics (1 Patient Set) - Simple Counts
This plugin displays a demographic break-down of a Patient Set.

(A) Query Builder.

Query Result #1

Query Definition

Location: chr8:126000000-126011000, chr13:78143000-78200000

Zygosity: Any

Result

Drop a Query Result to append to table

Chromosome ▲	Position	Reference	Alternate	Gene name	dbSNP ID	Result Type	Result
8	126010805	G	C		rs15905	MUTATION_COUNT	31
8	126010895	T	TG			MUTATION_COUNT	0
8	126010910	A	G			MUTATION_COUNT	1
13	78143516	G	A		rs143114727	MUTATION_COUNT	0
13	78146301	C	T		rs678047	MUTATION_COUNT	115
13	78146303	C	T		rs141416270	MUTATION_COUNT	0
13	78165565	A	G		rs141475107	MUTATION_COUNT	1
13	78176802	G	T			MUTATION_COUNT	0
13	78178550	G	A		rs2274016	MUTATION_COUNT	10
13	78183221	G	T			MUTATION_COUNT	1
13	78188078	A	G		rs8002725	MUTATION_COUNT	34
13	78191761	G	A			MUTATION_COUNT	0

chr8:126000000-126011000, chr1@20:39:
Circulatory sys@20:36:01 [8-23-2016] [de
chr8:126000000-126011000, chr1@20:34:
chr8:126000000-126011000, chr1@20:27:
Circulatory sys@20:17:31 [8-23-2016] [de

Genomic Query Tool
This plugin allows to send a query on genomic data.

Demographics (1 Patient Set) - Simple Counts
This plugin displays a demographic break-down of a Patient Set.

(B) Result Visualizer.

Figure 5.5: i2b2 front-end plugin.

We used the implementation of the FV cryptosystem provided within the NFLlib library [22]. The NFLlib is an optimized open-source C++ library dedicated to ideal lattice cryptography in the polynomial ring $\mathbb{Z}_q[x]/(x^l + 1)$ for l a power of 2. We chose this library because, to the best of our knowledge, it is the most efficient one for computations over polynomials. Indeed, NFLlib uses a mixed NTT-CRT representation to reduce computational costs: Number-Theoretic Transform (NTT) for polynomials [95] and Chinese Remainder Theorem (CRT) for their coefficients.

We used the FV encryption parameters, as reported in Table 5.5, in order to have 128 bits of security level.

Parameter	Value
Polynomial Degree	2048
Ciphertext Modulus	4,611,686,018,326,724,609 (62 bits)
Plaintext Modulus	1,000,000 (20 bits)

Table 5.5: Encryption Parameters

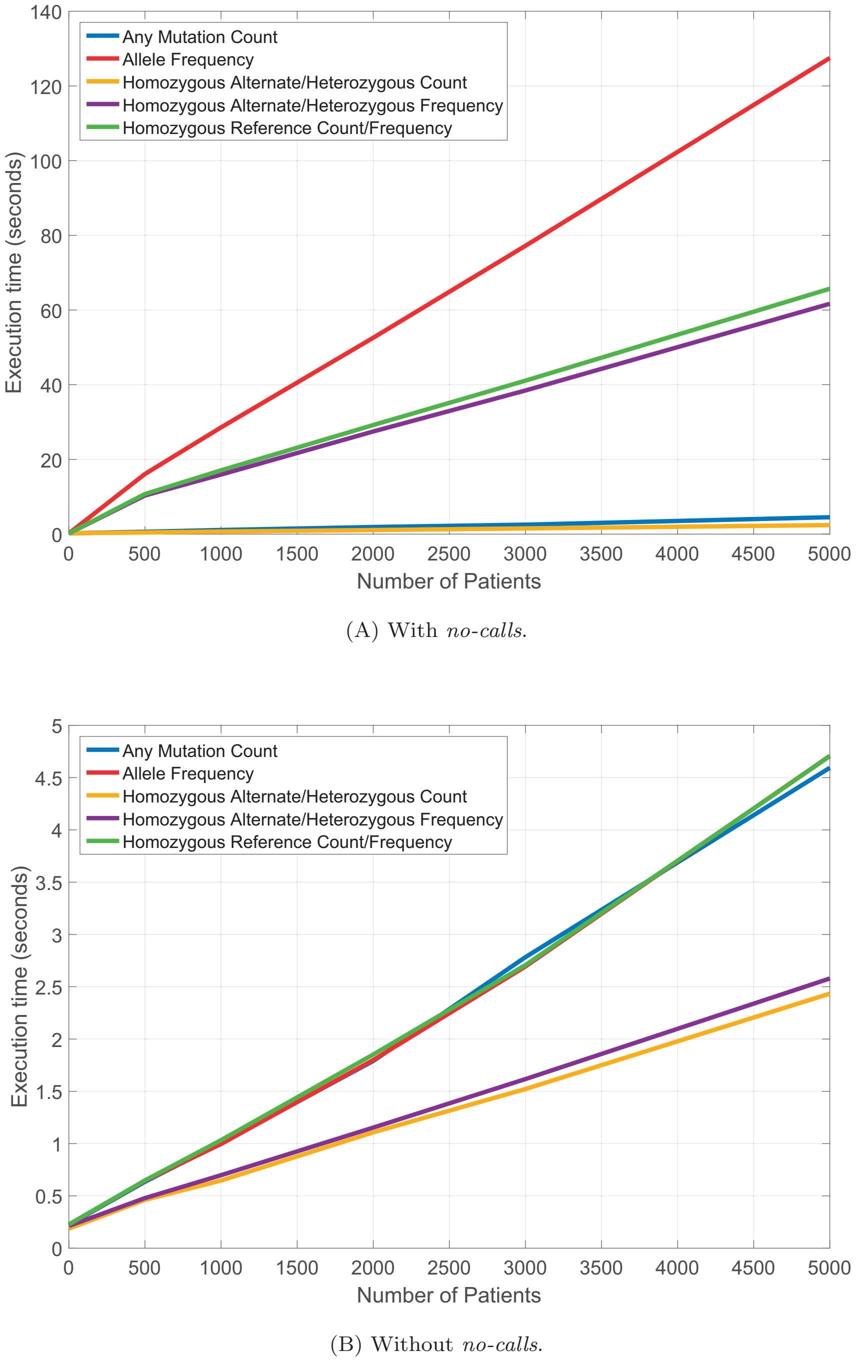
We used real genomic data coming from the exome sequencing of 392 samples giving a genotyping for 472,845 variants each. The resulting VCF file contained a total of 185,355,240 unphased genotypes. For clinical data, we used 90,454 clinical records from 134 patients available from the i2b2 demo version [5]. Patients from the i2b2 demo were duplicated in order to match the number of patients with genomic records. As a result, we had an initial cohort of 392 individuals with both clinical and genomic data. To test the scalability of our solution, this initial cohort was further extended by replicating individuals in order to obtain a cohort of 5,000 patients.

- Performance Analysis: We assessed the performance of our proposed solution in terms of storage and computational overheads.

To measure the storage overhead, we compared the initial size of genotypes within the original VCF file with the size of the encrypted genotypes stored on the i2b2 server. We did not consider the meta-data in the VCF file as this information is never modified before being stored on the i2b2 server. In general, in a VCF file, a genotype is represented by 4 bytes: 2 bytes for 2 alleles, 1 byte for '/' or '|', and 1 byte for a delimiter. As there were a total of 185,355,240 genotypes in our VCF file, their corresponding size was 707.07 MB. After encoding, packing and encrypting all genotypes in the VCF file, we obtained a set of 181,104 ciphertexts whose size is 5.82 GB. This corresponds to a storage overhead of 8x compared to the unencrypted VCF.

To measure the computational overhead, we ran experiments on all the privacy-preserving algorithms for summary statistics for different sets of variants and different cohort sizes. Experiments were run 100 times for each scenario and we report the average execution time in the following. We evaluated the different steps of the query execution phase, described in detail in Section 5.5.

Figure 5.6A and Figure 5.6B show the total execution time at the i2b2 server needed to compute the different summary statistics for increasing cohort sizes and a fixed query including 3,000 genetic variants with and without *no-calls*. It is easy to observe how the execution time increases linearly with the number of patients in the cohort and how the presence of *no-calls* in the VCF file has a significant impact on performance. Differences between the execution time for computing homozygous alternate/heterozygous counts/frequencies (yellow and purple curves) and the other summary statistics are mainly due to the different number of ciphertexts involved in the computation.



(A) With *no-calls*.

(B) Without *no-calls*.

Figure 5.6: Total query execution time at i2b2 server per number of patients in the cohort for a query involving 3000 genetic variants.

The execution time at the i2b2 does not depend only on the number of patients in the cohort but also on the number of consecutive genetic variants specified in the query, as shown in Figure 5.7. The differences in execution time between the different summary

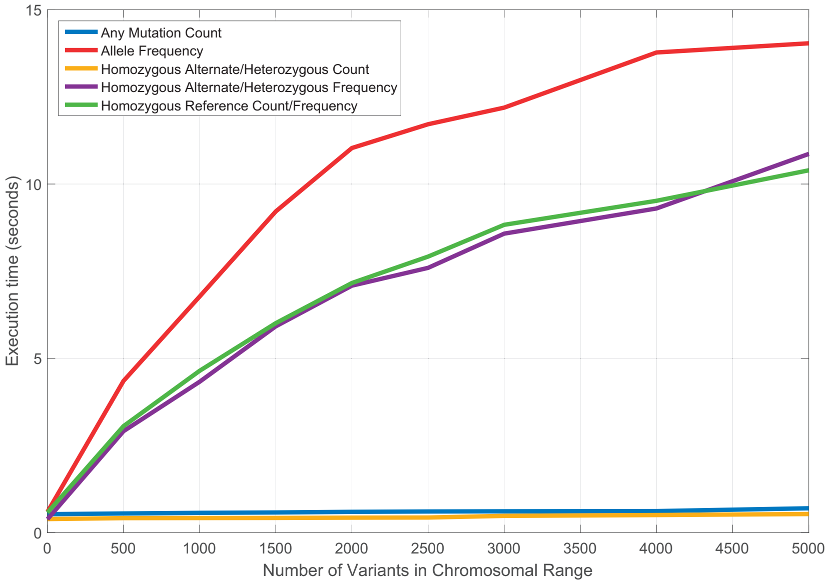


Figure 5.7: Total execution time at i2b2 server per number of consecutive genetic variants.

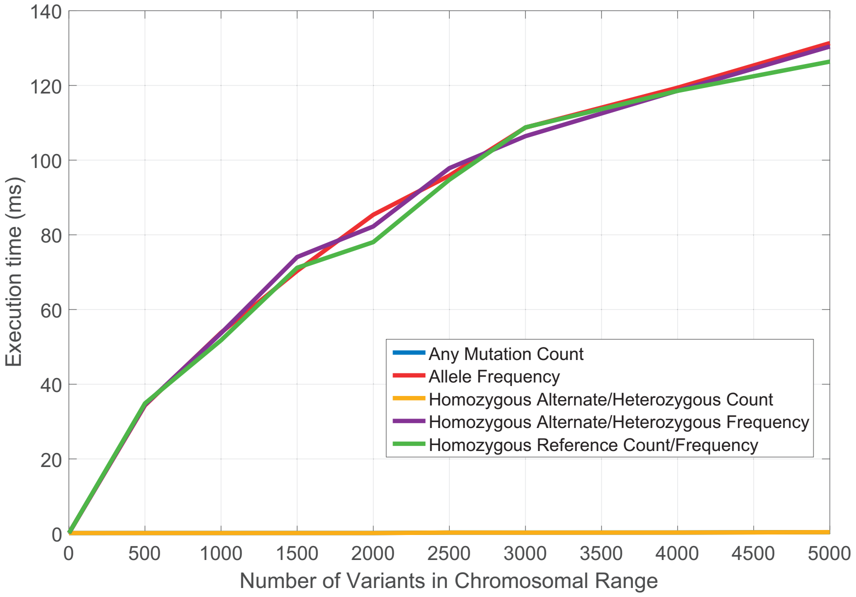
statistics depend on genotypes with no-call values. In particular, the computations of the number of mutations (blue curve) and the number of homozygous alternate/heterozygous (yellow curve) are significantly influenced only by the existence of genotypes with 1 no-call, whereas the other computations are also influenced by genotypes with 2 no-calls. Note that in our data we do not have genotypes with 1 no-call. From Fig.5.8A and Fig.5.8B we can observe that most of the computational time at the i2b2 server is due to data retrieval from the database and homomorphic computations.

Finally, Figure 5.9A and Figure 5.9B show the execution time for a full decryption of the query results for an increasing number of consecutive genetic variants for the contribution of the proxy server and the client, respectively. It is easy to observe that the execution time at the proxy server is similar to the execution time at the i2b2 server for partial decryption as the number of processed ciphertexts is the same. At the client side, we can observe a different behavior for the execution time as decryption is done variant-by-variant instead of ciphertext-by-ciphertext. Note that, because of genotypes with no-call values, the number of ciphertexts including the query results can be different from the number of ciphertexts including genetic variants.

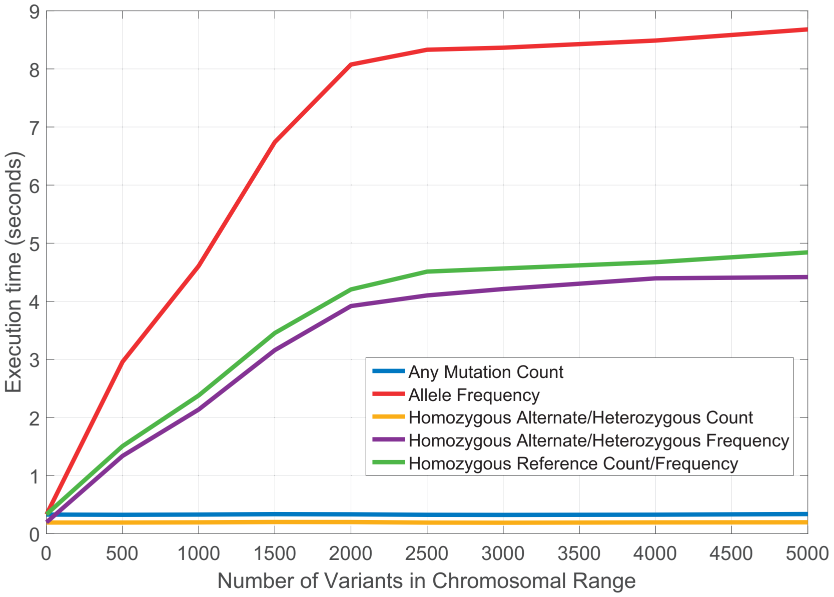
5.8 Discussion

We have thoroughly evaluated the performance of our solution on real data. Results show that generally privacy-preserving solutions, such as the one proposed in this work, can already be used in medical settings as new efficient enablers. Yet, some important points need to be further discussed.

- **Performance:** As it can be observed from the results of Figure 5.6A and Fig-



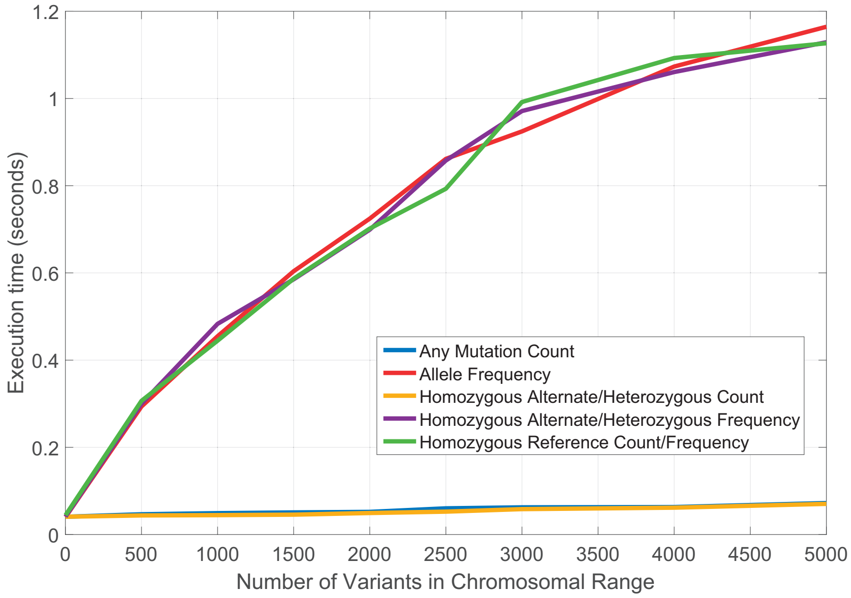
(A) Partial decryption.



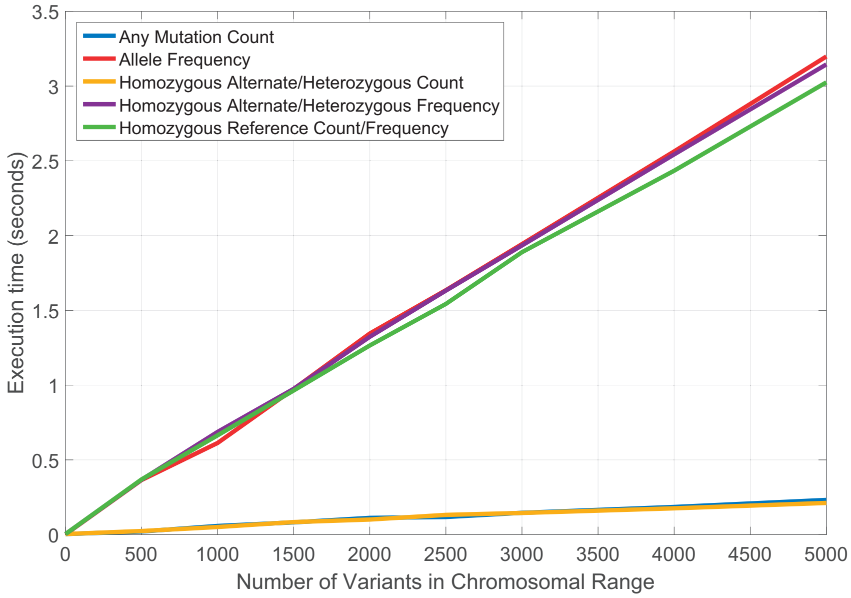
(B) Data retrieval and homomorphic computations.

Figure 5.8: Breakdown of total execution time at i2b2 server per number of consecutive genetic variants.

ure 5.6B, the main bottleneck for the execution time of queries involving specific types of summary statistics (e.g., allele or genotype frequencies) is due to the presence of genotypes with *no-call* values. Indeed, the number of key-value pairs (i.e., set of variants that



(A) At proxy server.



(B) At client.

Figure 5.9: Execution time for full decryption for queries with increasing number of consecutive genetic variants.

can be processed in parallel) generated by Algorithms 6, 8 and 10 at the i2b2 server can significantly grow if the distribution of *no-calls* is very different among patients. Yet, some quick alternative approaches can be used to easily address this issue. A first potential approach consists in using a different encoding for genotype values, as the one shown in Table 5.6, which maximizes performance at the expense of increasing the storage overhead from a factor of 8 to a factor of 20. Note that a storage overhead of 20x can be prohibitive for most institutions in case of large studies such as whole genome sequencing. It is definitely acceptable for studies on the exome. Another potential approach would be to perform genotype imputation before the genotype encoding in order to replace *no-calls* with imputed values at the expense of a slight decrease in accuracy. This said, it is easy to observe how the first alternative approach prioritizes high performance and high accuracy instead of low storage overhead, whereas the second approach ensures high performance and low storage overhead rather than full accuracy. Note that, by slightly sacrificing performance, our current solution assigns the highest priority to low storage cost and high accuracy as specified by CHUV's requirements (see Section 5.4). We leave for future work further investigations on how to improve our secure algorithms in Section 5.5.5 in order to optimize both performance and storage without sacrificing accuracy.

Genotype value	Genotype encoding				
	gt1	*gt2*	*gt3*	*gt4*	*gt5*
./.	0	0	0	0	0
./0	0	0	0	1	0
./1	0	0	0	1	1
0/0	1	0	0	2	0
0/1	0	1	0	2	1
1/1	0	0	1	2	2

Table 5.6: Genotype encoding optimizing performance.

As it is well-known in the security field, the perfect solution does not exist. It is always a matter of finding the best trade-offs between protection overhead, efficiency, and accuracy of the result. Our proposed solution is general enough to be fine-tuned according to the requirements.

- Query result perturbation: As explained in Section 5.5, our solution applies the standard and well-established Laplace mechanism [75] to independently perturb query results in order to satisfy the notion of differential privacy and prevent patient re-identification. Yet, the amount of noise that the i2b2 server needs to add to new queries of a given user grows linearly with the number of queries already answered for that user. This can substantially degrade the utility of the system as the results of later queries would be useless or, in other words, the number of useful queries would be limited. For this reason more sophisticated mechanisms could be used to obtain sublinear noise. For example, the *median* mechanism [174], the *exponential* mechanism [142] or the *multiplicative weights* mechanism [101] can answer exponentially more predicate queries than the Laplace mechanism. Indeed, the algorithm proposed by Vinterbo *et al.* [206] provides the option to incorporate user preferences with regard to individual query responses, thereby increasing utility to users without compromising privacy. The authors propose as well an evaluation of the privacy/utility trade-off with i2b2 which shows the efficacy of their method. This can be easily implemented in our system.

5.9 Summary

In this chapter, we have described how we designed, implemented and deployed, for the first time, a secure and efficient privacy-preserving solution for exploring genomic cohorts in a real operational scenario at the Lausanne University Hospital. So far, without proper security and privacy guarantees, the exploration of genetic cohorts has been extremely difficult and time-consuming. Thanks to its efficiency and strong security, we believe that our solution represents a powerful enabler in this context, especially when there is a need for sharing sensitive information in less protected environments. The adoption of privacy-preserving systems such as ours will undoubtedly foster data sharing and translational research on a larger scale.

To conclude, we acknowledge that the proposed solution addresses a simple use case by providing genomic summary statistics that can be securely computed through homomorphic additions. Yet, thanks to the flexibility of the FV scheme, more complex use cases such as privacy-preserving phenome-wide association studies (PheWAS) or GWAS can be envisioned and developed on top of our solution. We plan to extend the functional capabilities of the current system by addressing more complex use cases in future work. Finally, we want to emphasize that the goal of this chapter was not to discuss the parameters' values that determine the best privacy and accuracy trade-off when using differential privacy, but to provide the tools that enable privacy-preserving exploration of genetic cohorts. We believe that such a discussion should be a prerogative of database administrators, end-users and hospitals' institutional review boards.

Chapter 6

Addressing Beacon Re-Identification Attacks: Quantification and Mitigation of Privacy Risks

In the previous chapter, we have addressed the problem of re-using, in a privacy-preserving way, clinical and genomic data for medical research. We have focused on the particular use case of a medical institution willing to outsource the storage of its data to an untrusted centralized repository so that multiple researchers can access it. Yet, the amount of data required for significantly advancing medical research usually goes beyond the capability of a single institution. The need to share clinical and genomic data among multiple and mutually-distrustful stakeholders is constantly increasing but privacy and security concerns often represent roadblocks difficult to overcome. In this chapter, we consider the first existing federated system for genomic variant discovery, the Beacon Project of the Global Alliance for Genomics and Health. We analyze one of the most daunting privacy risks affecting this kind of systems, i.e., the re-identification risk, and propose a set of practical mitigation strategies.

6.1 Introduction

The Global Alliance for Genomics and Health (GA4GH) [92] conceived the Beacon Project as a means of testing the willingness of international sites to share genomic data in the simplest of all technical contexts: a public web service that any data holder could implement to enable users to submit queries of the form "Do you have any genomes with nucleotide A at position 100,735 on chromosome 3?," to which the service would respond with "Yes" or "No". A site offering this service is called a *beacon* and is responsible for assuring that genomic data are exposed through the Beacon service only with the permission of the individual to whom the data pertain, and in accordance with the GA4GH ethical framework [125] and privacy and security policy [93]. Thus the Beacon service is designed to be technically simple, easy to implement, and privacy protective.

The availability of vast quantities of high-quality genomic and health data is essential to the advancement of biomedical knowledge. Yet privacy concerns often limit researchers' ability to access potentially identifiable health data. Indeed, in some cases, privacy laws and regulations may actually impede individuals' ability to make their own data available to researchers [193]. This problem is particularly acute in the field of genomics, where the vast majority of variants predicted to be functionally important are extremely rare, occurring in <0.5% of the population [192]. As a result, it is unlikely that any single institution will hold enough data to achieve sufficient statistical power in studying any particular condition. Recognizing the urgent need for federation across organizations, the GA4GH was formed in 2013 to enable responsible sharing of genomic and health-related data by establishing consistent policy, and interoperable standards and protocols.

From its inception, the GA4GH has been committed to achieving a responsible and effective balance between data sharing and individual privacy, a challenge that has been extensively explored in the literature [107, 176, 77, 99]. In 2008, Homer et al. [107] showed that statistical techniques can reveal the presence or absence of an individual in a genomic data set, even when the targeted individual's genome accounts for <0.1% of the total data. The publication of this paper had a significant impact, prompting several major institutions, including the Wellcome Trust and the US National Institutes of Health, to limit public access to data formerly adjudged to be safely anonymous [96]. As this scenario demonstrates, privacy concerns can undermine the ability of researchers to publish and access genomic data.

Initially, the GA4GH recognized that the Beacon approach could reveal information about individuals in a data set. However, in performing the risk assessment, the GA4GH recognized several aspects that served to mitigate the risk that any individual would be identified based on Beacon search. First, the Beacon user interface is extremely restrictive, enabling query only for the presence or absence of the four nucleotides (A, C, T, G) that comprise every individual's genome. Second, the number of individual genomes aggregated in each beacon is very large. Third, for a data seeker to be able to identify an individual through Beacon queries would require as a pre-condition that the data seeker possess a significant amount of genomic data associated with the targeted individual, such as a variant call format (VCF) file of the individual's whole genome sequence. In such case, a potential adversary would know all variants in the individual's genome, and would have much more efficient means of discovering a disease association than persistent beacon queries. Thus GA4GH concluded that the risk of a data seeker identifying an individual through Beacon queries was acceptably low, even for the case of a data seeker willing to violate GA4GH's ethical standards.

However, Shringarpure and Bustamante [181] describe an attack in which an anonymous adversary, even with knowledge of only a small portion of a target's genome can successfully launch a re-identification attack: In a beacon comprising 1,000 individuals for instance, 5,000 queries suffice. Such an attack relies on a likelihood ratio test whose power is a function of the responses returned by the beacon, the size of the data set, the allele-frequency spectrum, and the sequencing error rate. Their paper demonstrates that under certain conditions, the anonymous-access model implemented by the Beacon Project does not prevent identification of individuals whose genomes could be exposed through a Beacon interface.

The goal of this chapter is to examine potential vulnerabilities and risks associated with the Beacon model, and to explore ways of mitigating re-identification risks – thus enhancing Beacon privacy protections. Re-identification is the process by which

Notation	Description
N	Total number of genomes in the beacon.
$Q = \{q_1, ..., q_n\}$	Set of n queries.
$R = \{x_1, ..., x_n\}$	Set of n responses returned by the beacon.
H_0	Null hypothesis: query genome is not in beacon.
H_1	Alternative hypothesis: query genome is in beacon.
f_i	Alternate allele frequency at the SNP corresponding to query q_i.
p_i	Reference allele frequency at the SNP corresponding to query q_i, ($p_i = 1 - f_i$).
$L(R)$	Log-likelihood of a response set $R = \{x_1, ..., x_n\}$.
$L_{H_0}(R)$, $L_{H_1}(R)$	Log-likelihood under the null/alternative hypothesis.
$beta(a, b)$	Alternate allele frequency distribution assumed in the original by Shringarpure and Bustamante.
D^i_{N-1}	Probability that none of the $N - 1$ genomes in the beacon has an alternate allele for query q_i.
D^i_N	Probability that none of the N genomes in the beacon has an alternate allele for query q_i.
δ	Probability of mismatch between the query genome and its copy in the beacon due to sequencing errors.
$j \in 1, ..., N$	Index of individuals in the beacon.
$i \in 1, ..., n$	Index of queries.
α	Type I error: $P(\text{reject } H_0 \mid H_0 \text{ is true})$.
β	Type II error: $P(\text{accept } H_0 \mid H_1 \text{ is true})$.
$power$	$P(\text{reject } H_0 \mid H_1 \text{ is true}) = 1 - \beta$.
r_i	Risk of query i.
b_j	Budget of patient j.
LRD_{H_1}, LRD_{H_0}	Likelihood ratio distribution under the alternative/null hypothesis.
Λ	LRT statistic.
t	Cut-off for the LRT statistic Λ (the null hypothesis is rejected if $\Lambda < t$).
Q^j	Set of queries answered by individual j.
k	Threshold on the number of individuals carrying an alternate allele at the queried SNP (used in defense $S1$).
ϵ	Probability of adding noise on unique alleles (used in defense $S2$).
B_j	Budget for individual j. Initially $B_j = -log(p)$ for every j (used in defense $S3$).
r_i	Risk for query i (i.e., how much budget for every individual j is deducted from B_j if the beacon answers query i).
LRT	Likelihood Ratio Test.
SNP	Single Nucleotide Polymorphism.
VCF	Variant Call Format.

Table 6.1: Notation used throughout the chapter.

anonymized personal data is matched with its true owner [14]. We first analyze the re-identification threat described by Shringarpure and Bustamante and the vulnerability the attack exploited. We then propose an optimized version of the attack that considers an adversary with some background knowledge about the allele frequencies (AFs) in the targeted beacon. We describe three potential strategies for mitigating the risk of re-identification, and assess their effectiveness through several experiments with data obtained from the 1000 Genomes Project [194]. We conclude the chapter by discussing strengths and weaknesses of the proposed strategies and by providing some recommendation for strengthening Beacon privacy protection. We summarize the notation used in this Chapter in Table 6.1.

6.2 Original Re-Identification Attack

We begin by describing the re-identification attack proposed by Shringarpure and Bustamante [181]. In the following we refer to it as the "SB attack".

We assume a beacon of N unrelated individuals. Queries to the beacon are of the form $q = \{C, P, A\}$. The beacon responds "*Yes*" (encoded as 1) if A is an alternate allele at position P on chromosome C, and if it appears at least once in the beacon population. Otherwise it responds "*No*" (encoded as 0). For a set of n queries $Q = \{q_1, \dots, q_n\}$, the beacon returns responses $R = \{x_1, \dots, x_n\}$.

As noted earlier, the setting of the SB attack is similar to that of previous works such as that of Homer et al. [107]. The attacker is assumed to have access to the VCF file of a target victim's genome and queries the beacon at heterozygous positions to determine whether the victim is in the beacon or not. The SB attack relies on a likelihood-ratio test (LRT) that evaluates the likelihood of the beacon's responses under two possible hypotheses:

- The null hypothesis H_0: The queried victim's genome is not in the beacon.
- The alternative hypothesis H_1: The queried victim's genome is in the beacon.

The re-identification risk is measured by the power of such a test, i.e., $\Pr(\text{reject}\, H_0 | H_1\, \text{true})$. To make their test as general as possible, Shringarpure and Bustamante assume only that the attacker knows the beacon size N, as well as the site frequency spectrum of the beacon population. Formally, the alternate allele frequency f_i of a heterozygous SNP observed in the population is assumed to be distributed as $f_i \sim \text{beta}(a, b)$ for population parameters a, b. Their LRT test further allows for a probability δ of *sequencing errors*, resulting in a mismatch between the attacker's copy of a genome and the copy in the beacon. Given a set of beacon responses $R = \{x_1, \dots, x_n\}$, the log-likelihood of the sequence is

$$L(R) = \sum_{i=1}^{n} x_i \log \Pr(x_i = 1) + (1 - x_i) \log \Pr(x_i = 0) \; . \tag{6.1}$$

Under H_1, let D^i_{N-1} denote the probability that none of the $N - 1$ other genomes in the beacon have an alternate allele at position i. Similarly, under H_0 we denote by D^i_N the probability that none of the N genomes in the beacon have an alternate allele at i. Then, under the two hypotheses, we have

$$L_{H_1}(R) = \sum_{i=1}^{n} x_i \log(1 - \delta D^i_{N-1}) + (1 - x_i) \log(\delta D^i_{N-1}) \; , \tag{6.2}$$

$$L_{H_0}(R) = \sum_{i=1}^{n} x_i \log(1 - D^i_N) + (1 - x_i) \log(D^i_N) \; . \tag{6.3}$$

Shringarpure and Bustamante show that under their assumptions, for any position i we have $D^i_{N-1} = \mathbb{E}[p_i^{2N-2}]$ and $D^i_N = \mathbb{E}[p_i^{2N}]$, where $p_i \sim \text{beta}(b, a)$. The log of the likelihood-ratio test is given by

$$\Lambda = L_{H_0}(R) - L_{H_1}(R) = nB + C \sum_{i=1}^{n} x_i \; , \tag{6.4}$$

where B and C are constant for N, δ, a, b fixed. Thus, $\sum_{i=1}^{n} x_i$ (the number of "yes" responses from the beacon) is a sufficient statistic for the LRT.

6.3 "Optimal" Attack With Real Allele Frequencies

The SB attack removes direct dependency on allele frequencies and sets conservative bounds for the number of queries required for successful re-identification. We consider here a more capable and determined attacker who has access to some background knowledge on allele frequencies and optimizes his attack by querying the rarest alleles in the victim's genome first. In other words, similarly to best practices in forensics, the attacker makes use of alleles with maximum re-identification power instead of performing random requests. This assumption appears reasonable in practice, as allele frequency information for different ancestry groups is already publicly available on the Web (e.g., 1000 Genomes Project [64], HapMap project [65], etc.) and easily accessible even by inexpert attackers. We show through several experiments (see Section 6.5) that this new attack is significantly more powerful than the original SB attack, even when the attacker has incomplete knowledge of AFs in the beacon.

Formally, the attacker assumes allele frequencies $f_1, f_2, \ldots, f_M$ for the M SNPs in the victim's genome. Without loss of generality, we assume the frequencies are already ordered (i.e, $f_1 \leq f_2 \leq \cdots \leq f_M$). Then, the attacker will maximize its re-identification power by first querying those SNPs which are least likely to appear in the beacon under H_0, namely those with lowest frequency. In this setting, Equations (6.2) and (6.3) still hold, but the computation of D^i_{N-1} and D^i_N is different. Under the alternative hypothesis, we have

$$\begin{aligned} D^i_{N-1} &= \Pr(\text{none of the other } N-1 \text{ genomes have an alternate allele at position } i) \\ &= \left((1-f_i)^2\right)^{N-1} \\ &= (1-f_i)^{2N-2} . \end{aligned}$$

Similarly, under H_0 we have $D^i_N = (1-f_i)^{2N}$.

As the probabilities D^i_{N-1} and D^i_N now directly depend on the position i, we have the following LRT

$$\begin{aligned} \Lambda &= L_{H_0}(R) - L_{H_1}(R) \\ &= \sum_{i=1}^{n} \log\left(\frac{D^i_N}{\delta D^i_{N-1}}\right) + \log\left(\frac{\delta D^i_{N-1}(1-D^i_N)}{D^i_N(1-\delta D^i_{N-1})}\right) x_i \\ &= \sum_{i=1}^{n} \log\left(\delta^{-1}(1-f_i)^2\right) + \log\left(\frac{\delta}{(1-f_i)^2} \cdot \frac{1-(1-f_i)^{2N}}{1-\delta(1-f_i)^{2N-2}}\right) x_i . \end{aligned} \tag{6.5}$$

We will evaluate the power of this test empirically, through experiments in a variety of settings with real data and different levels of adversarial background knowledge. We will estimate the null distribution of the LRT by computing Equation (6.5) for a number of control individuals known not to be in the beacon. The null hypothesis is rejected if $\Lambda < t$ for some threshold t. We then let t_α be such that $\Pr[\Lambda < t_\alpha \mid H_0] = \alpha$. The power of the test is computed as $1 - \beta = \Pr[\Lambda < t_\alpha \mid H_1]$, where the distribution of Λ given H_1 is estimated by querying individuals in the experimental beacon.

6.4 Risk Mitigation Strategies

Based on the "optimal" version of the re-identification attack, we propose three different practical strategies to mitigate the risk. Without loss of generality, we can assume that any defense mechanism that effectively mitigates the "optimal" re-identification attack also effectively mitigates the original SB attack. Our experimental results in Section 6.5 show the validity of this assumption.

6.4.1 Beacon Alteration Strategy

The first strategy ($S1$) relies on the observation that most of the statistical power in the re-identification attack comes from queries targeting unique alleles in the beacon database. In particular, $S1$ alters the beacon by answering a query with *"Yes"* only if there are at least $k > 1$ individuals sharing the queried allele. In other words, k is the minimum number of queried alleles present in beacon when returning *"Yes"*. Note that for the case where $k = 2$, this is different from responding *"Yes"* when at least 2 out of the $2N$ alleles in the beacon are alternate (N represents the number of individuals in the beacon), as a single individual may have 2 copies of the alternate allele. Current beacons set $k = 1$; i.e., when there are one or more individuals in the population with the queried allele, the answer will be *"Yes"*. We assume the value of k is made public, hence the attacker will modify the attack to accommodate this change. Yet, already for $k = 2$ we found that in practice what the attacker can infer is limited (see Section 6.5.3 for details).

Formally, the attacker knows the allele frequencies for the SNPs in the victim's genome, and these frequencies can be ordered randomly or sequentially. In this setting, Equation (6.1) still holds, but Equations (6.2) and (6.3) needs to be modified as they now depend on k.

Under the alternative hypothesis, the beacon responds "*No*" if either of the following two conditions is met.

- A sequencing error δ occurred and less than k other individuals have a copy of the allele
- No sequencing error occurred but less than $k - 1$ other individuals have a copy of the allele

Hence, we have

$$\begin{aligned} L_{H_1}(R,k) &= \sum_{i=1}^{n} x_i \log(\Pr(x_i = 1|H_1,k)) + (1-x_i)\log(\Pr(x_i = 0|H_1,k)) \\ &= \sum_{i=1}^{n} x_i \log(\delta(1 - D^i_{N-1}(k)) + (1-\delta)(1 - D^i_{N-1}(k-1))) \\ &\quad + (1-x_i)\log(\delta D^i_{N-1}(k) + (1-\delta) D^i_{N-1}(k-1)) \end{aligned} \tag{6.6}$$

where $D^i_{N-1}(k)$ denotes the probability that fewer than k out of $N-1$ individuals have an alternate allele (for query q_i). Let X_{N-1,s_i} be a random variable following a binomial distribution with $N-1$ trials and success probability $s_i = 1 - (1 - f_i)^2$, where s_i represents the probability that a given individual (other than the victim) has at least one copy of an alternate allele (for query q_i) with frequency f_i. Then,

$$
\begin{aligned}
D^i_{N-1}(k) &= \Pr(\text{less than } k \text{ out of } N-1 \text{ genomes have an alternate allele at position } i)\\
&= \Pr(X_{N-1,s_i} < k) = \sum_{j=0}^{k-1} \binom{N-1}{j} (1-(1-f_i)^2)^j ((1-f_i)^2)^{N-1-j}. \qquad (6.7)
\end{aligned}
$$

Similarly, under the null hypothesis, the probability that the beacon responds "*No*" to a query q_i for an allele with frequency f_i is the probability that at most $k-1$ individuals have a copy of the query allele. Hence, we have

$$
\begin{aligned}
L_{H_0}(R,k) &= \sum_{i=1}^{n} x_i \log(\Pr(x_i = 1|H_0,k)) + (1-x_i)\log(\Pr(x_i = 0|H_0,k))\\
&= \sum_{i=1}^{n} x_i \log(1 - D^i_N(k)) + (1-x_i)\log(D^i_N(k)) \qquad (6.8)
\end{aligned}
$$

Therefore, the likelihood ratio test statistic $\Lambda(k)$ when $k \geq 2$ can be computed by

$$
\begin{aligned}
\Lambda(k) &= L_{H_0}(R,k) - L_{H_1}(R,k)\\
&= \sum_{i=1}^{n} \log\left(\frac{D^i_N(k)}{\delta D^i_{N-1}(k) + (1-\delta)D^i_{N-1}(k-1)}\right)\\
&+ \log\left(\frac{(1-D^i_N(k))(\delta D^i_{N-1}(k) + (1-\delta)D^i_{N-1}(k-1))}{D^i_N(k)(\delta(1-D^i_{N-1}(k)) + (1-\delta)(1-D^i_{N-1}(k-1)))}\right) x_i \; . \qquad (6.9)
\end{aligned}
$$

Note that if $k = 1$, from Equations (6.6) and (6.8) we can obtain Equations (6.2) and (6.3), respectively.

An alternative approach is to hide the precise number of individuals within a beacon database and instead provide an approximate database size (e.g., the reported database size is 100 although the actual database size is 1000). In this case, let the approximate size of a beacon database that the attacker knows be N_a; thus, the LRT statistic Λ can be calculated according to Equation (6.5), where $N = N_a$.

$$
\Lambda = \sum_{i=1}^{n} \log\left(\delta^{-1}(1-f_i)^2\right) + \log\left(\frac{\delta}{(1-f_i)^2} \cdot \frac{1-(1-f_i)^{2N_a}}{1-\delta(1-f_i)^{2N_a-2}}\right) x_i \; . \qquad (6.10)
$$

6.4.2 Random Flipping Strategy

The second strategy ($S2$) relies on the same observation but instead of altering the beacon response, it introduces noise into the original data. The disadvantage of $S1$ is that only a subset of variations (e.g., the *non-unique* SNPs when $k = 2$) in the beacon population can be queried. In practice, *unique* alleles that are likely to be the most useful in human genetics research, are completely hidden. $S2$ improves the usability of the beacon over $S1$ as it hides only a portion ϵ of unique alleles, but not all. In other words, a beacon with $S2$ will add noise with probability ϵ only to unique alleles in the database and provide false answers (e.g., "*No*" instead of "*Yes*") to queries targeting these unique alleles. The main goal of $S2$ is to share as many unique alleles as possible while reducing the likelihood that the information released will be sufficient to re-identify an individual in the database. Without loss of generality, we assume the value of ϵ is public. As for $S1$, the attacker will adapt the LRT statistic to take it into account.

Formerly, and also in this case, the attacker knows the allele frequencies for the SNPs in the victim's genome and performs queries by following the *rare-allele-first* model. Similarly to $S1$, Equation (6.1) still holds, but Equations (6.2) and (6.3) needs to be modified again as they now depend on ϵ.

Under the alternative hypothesis, the beacon responds "*No*" if either of the following two conditions is met.

- A sequencing error δ occurred and none of the other $N-1$ participants has a copy of the allele.
- An artificial error ϵ occurred and the allele is unique. Note that an allele is unique if a sequencing error occurred and another participant has a copy of the allele or if no sequencing error occurred and none of the other $N-1$ participants has a copy of the allele.

Hence, we have

$$L_{H_1}(R,\epsilon) = \sum_{i=1}^{n} x_i \log(\Pr(x_i = 1|H_1,\epsilon)) + (1-x_i)\log(\Pr(x_i = 0|H_1,\epsilon)), \tag{6.11}$$

where the probability of a "*No*" answer is

$$\begin{aligned}\Pr(x_i = 0|H_1,\epsilon) &= \Pr(\text{none of } N-1 \text{ genomes have an alternate allele at position } i)\\ &\quad + \epsilon \Pr(\text{allele at position } i \text{ is unique})\\ &= \delta D^i_{N-1} + \epsilon(\delta \Pr(X_{N-1,s_i} = 1) + (1-\delta)D^i_{N-1})\\ &= \epsilon\delta \Pr(X_{N-1,s_i} = 1) + (\delta + \epsilon - \epsilon\delta)D^i_{N-1}. \end{aligned} \tag{6.12}$$

Note that $\Pr(X_{N-1,s_i} = 1)$ denotes the probability that another participant has a copy of the allele at position i. As in Sec. 6.4.1, we can derive such a probability as

$$\begin{aligned}\Pr(X_{N-1,s_i} = 1) &= \binom{N-1}{1}(1-(1-f_i)^2)((1-f_i)^2)^{N-1}\\ &= (N-1)(1-(1-f_i)^2)((1-f_i)^2)^{N-1}.\end{aligned} \tag{6.13}$$

Similarly, under the null hypothesis we have

$$L_{H_0}(R,\epsilon) = \sum_{i=1}^{n} x_i \log(\Pr(x_i = 1|H_0,\epsilon)) + (1-x_i)\log(\Pr(x_i = 0|H_0,\epsilon)), \tag{6.14}$$

where the probability of receiving a "*No*" answer from the beacon is

$$\begin{aligned}\Pr(x_i = 0|H_0,\epsilon) &= \Pr(\text{none of } N \text{ genomes have an alternate allele at position } i)\\ &\quad + \epsilon \Pr(\text{allele at position } i \text{ is unique})\\ &= D^i_N + \epsilon \Pr(X_{N,s_i} = 1)\end{aligned} \tag{6.15}$$

Finally, the likelihood ratio test statistic $\Lambda(\epsilon)$ can be easily derived from Equations (6.11) and (6.14) as in previous sections.

6.4.3 Query Budget Per Individual Strategy

The third strategy ($S3$) mitigates the re-identification risk by assigning a budget to every individual in the database; this budget is applied to each authenticated Beacon user. With respect to strategies $S1$ and $S2$, $S3$ leverages two additional assumptions:

- Each Beacon user has been identity proofed, holds a single account, is authenticated, and does not collude. If users are allowed to collude, then to be effective, $S3$ will have a dramatic impact on the utility of the system. This assumption appears reasonable in practice as, in order to collude, a user needs by definition to involve someone else. We assume that each user holds a single Beacon account to eliminate the possibility of a single user simulating multiple profiles in collusion, which carries higher risk than either collusion among multiple users or a re-identification attack that can be undertaken at an individual scale. This is because an attack involving multiple accounts, all working on behalf of a single attacker, does not require exchanging files with other users, and could be conducted more quickly than a single-threaded attack.
- The attacker has accurate genomic information, which means $\delta = 0$. This is a worst-case assumption because if we can prevent re-identification under this condition, we can prevent the proposed "optimal" attack, too. Note that in practice, because there are some sequencing errors (i.e., $\delta > 0$), the attacker will actually have less power. Hence, this approach is conservative from a re-identification point of view. Moreover, by assuming $\delta = 0$, we can significantly simplify the analytical treatment of the problem.

The basic idea is that each time an individual's genome contributes to a *"Yes"* answer for a given query (i.e., the individual has the allele specified by the query), her corresponding budget for that user is reduced by an amount that depends on the frequency of the queried allele. If her budget is less than this amount, her information will not be used to answer that query and the individual will be removed from the dataset, as shown in Algorithm 11. In this way, the privacy of the individual will be always preserved at a cost of a slight decrease of utility.

Let R be the set of responses of the beacon; the goal of $S3$ is to keep track of the power of the attack, which is based on the LRT $\Lambda = L_{H_0}(R) - L_{H_1}(R)$, to prevent any individual genome from contributing to a query response that can leak identity information with high confidence.

More formally, we define a cut-off threshold t_α on the value of Λ to determine which hypothesis to accept (i.e., the null hypothesis is rejected if $\Lambda < t_\alpha$). Then the false-positive rate is $\alpha = \Pr[\Lambda < t_\alpha \mid H_0]$ and the power of the test is $1 - \beta = \Pr[\Lambda < t_\alpha \mid H_1]$.

So to validate that the original attack is mitigated by $S3$, we first need to know the distribution of Λ under H_0 and H_1. In the analysis by Shringarpure and Bustamante, it is shown that Λ is asymptotically Gaussian under both hypotheses (with different parameters). In our case, this result does not hold because we set $\delta = 0$ and assume fixed allele frequencies f_i for each allele.

The crucial observation here is that since $\delta = 0$, if the queried individual is in the beacon it must be that the beacon responds "Yes" to all queries $q_i \in Q$ made by the adversary for a query individual. Let R_{yes} denote the sequence of all "Yes" responses. We consider two cases:

Algorithm 11: Algorithm describing mitigation strategy S3

```
Require: Upper bound on test errors, $p$.
1: Set all $b_j = -\log(p)$.
2: Receive $i$-th query and check whether it has been asked before. If yes, go to Step 3. If no,
   go to Step 4.
3: Return the previous answer, then go to Step 2.
4: Compute the risk $r_i = -\log(1 - D_i^N)$.
5: Check whether there are any records with the asked variant and $b_j > r_i$. If no, return no
   and go to step 2.
6: For all the individuals with such variant and $b_j > r_i$, reduce their budgets by $r_i$. Then
   return yes.
7: Go back to step 2 and wait for the next query.
```

- $R = R_{\text{yes}}$. One then easily obtains:

$$L_{H_1}(R) = 0, \quad L_{H_0}(R) = \sum_{i=1}^{n} \log(1 - D_N^i), \quad \Lambda = \sum_{i=1}^{n} \log(1 - D_N^i) \ . \tag{6.16}$$

- $R \neq R_{\text{yes}}$. Then, we have:

$$L_{H_1}(R) = -\infty, \quad L_{H_0}(R) \in \mathbb{R}, \quad \Lambda = \infty \ . \tag{6.17}$$

So we see that in any case, the random variable Λ can only take on two values, either $\sum_{i=1}^{n} \log(1 - D_N^i)$ or ∞.

Now, if H_1 is true, R must be R_{yes}. Thus, we have that the distribution of Λ under H_1 reduces to the constant $\sum_{i=1}^{n} \log(1 - D_N^i)$.

If H_0 is true, the beacon responds "Yes" to query q_i with probability $1 - D_N^i$. Thus, $\Pr[R = R_{\text{yes}} \mid H_0] = \prod_{i=1}^{n}(1 - D_N^i)$. Then, under H_0, Λ is a random variable that takes value $\sum_{i=1}^{n} \log(1 - D_N^i)$ with probability $\prod_{i=1}^{n}(1 - D_N^i)$, and value ∞ otherwise. In summary:

$$\Lambda \mid H_1 = \sum_{i=1}^{n} \log(1 - D_N^i) \quad \text{with probability } 1 \ ,$$

$$\Lambda \mid H_0 = \begin{cases} \sum_{i=1}^{n} \log(1 - D_N^i) & \text{with probability } \prod_{i=1}^{n}(1 - D_N^i) \ , \\ \infty & \text{otherwise.} \end{cases}$$

So the cut-off threshold t must be chosen somewhere in $\left]\sum_{i=1}^{n} \log(1 - D_N^i), +\infty\right[$. According to the above, the power of the adversary will always be 1 (the adversary will never conclude that the victim is not in the beacon when she actually is). So our only control is over the false-positive rate $\alpha = \prod_{i=1}^{n}(1 - D_N^i)$. The goal of our strategy here is to dismiss an individual from consideration for any further query responses as soon as including her data would enable the adversary to construct a powerful re-identification test for that individual. By this, we mean a test with power 1 and false positive rate $\alpha \leq p$, for some chosen p. Our budget method sets $b_j = -\log(p)$ at first and then each time a query is made for an allele that a individual possesses, we first check whether the budget of the individual is larger than $-\log(1 - D_N^i)$, then reduce his budget by $-\log(1 - D_N^i)$. In this way we ensure that for each individual j, $\prod_{i \in Q^j}(1 - D_N^i) > p$, where Q^j represents the subset of queries made for alleles that individual j possesses, and for which individual j was considered when constructing the response.

For simplicity, we consider here that an adversary that wishes to re-identify individual j will only query SNPs for which j possesses the alternate allele (assuming $\delta = 0$). Indeed, for a query for a variant that j does not possess, we have $\Pr[x_i = 1 \mid H_1] = 1 - D_{N-1}^i$ and $\Pr[x_i = 1 \mid H_0] = 1 - D_N^i$, which are negligibly close for large N. Thus, such queries can simply be considered as useless for distinguishing H_0 from H_1.

6.5 Experiments With Real Data

To evaluate the effectiveness of the proposed strategies in reducing risk under the "optimal" attack with real AFs, we designed and ran several experiments on real data with the following setup. We created a beacon composed of $1,235$ samples of chromosome 10 randomly chosen from the $2,504$ individuals in phase 3 of the 1000 Genomes Project [64]. A total of 31 relatives were removed. The resulting data set consists of individuals with either European, African, admixed American, East Asian or South Asian ancestries. Among these samples, 100 were selected as the control set. Similarly, from the remaining individuals not in the beacon, 100 were selected as the test set.

The null distribution of the LRT statistic was obtained through the exact-test computation on the 100 individuals in the test set (i.e., not in the beacon). With a false positive rate of $\alpha = 5\%$ we computed the power $(1 - \beta)$ as the proportion of test rejected (i.e., when $\Lambda < t_\alpha$) for the control set (i.e., how many individuals in the control set, hence in the beacon, were successfully re-identified).

6.5.1 "Optimal" Re-Identification Attack in Single-Population Beacon

We evaluated the re-identification power of our attack on a beacon composed by individuals coming from the same ancestry group. From phase 3 of the 1000 Genomes Project, we selected 502 samples of European (EUR) ancestry and we randomly picked half of them to set up the beacon. The remaining half was used to compute the EUR population allele frequencies. We considered several scenarios where the attacker has different types of background information.

As expected, results in Figure 6.1 show that the worst-case scenario is represented by an attacker knowing the exact ancestry of the population in the beacon. With only three SNPs, beacon membership could be re-identified with 100% power and 5% false-positive rate. Yet, because the beacon ancestry information is not always public, a more realistic scenario is to consider an attacker that only knows the allele frequencies of a random population possibly from a different ancestry than the one of the beacon. Even with the least precise background information (in this case the AFs from East Asian (EAS) ancestry), 36 SNPs are sufficient to re-identify an individual. Figure 6.2 shows the Kendall rank correlation coefficient between the actual allele frequencies in the beacon and the allele frequencies from different ancestry groups. By combining the information in Figure 6.1 and 6.2, it is easy to observe that the higher the ordinal association between the beacon AFs and the AFs known by the attacker, the fewer queries are needed to re-identify with 100% power and 5% false-positive rate.

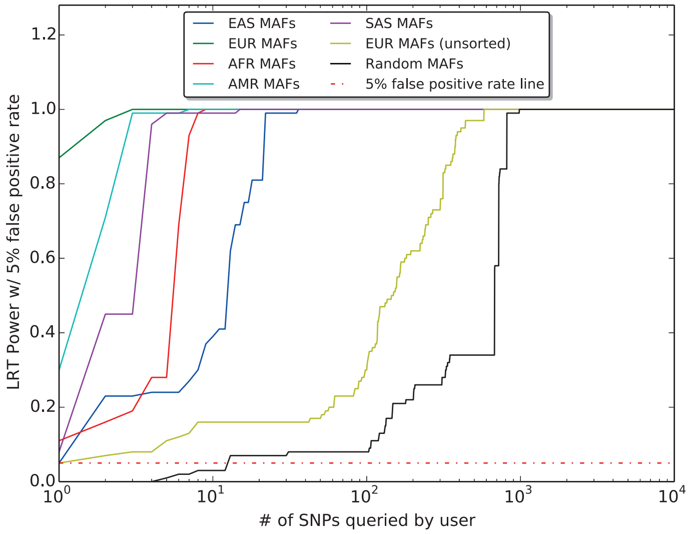


Figure 6.1: "Optimal" Re-Identification Attack in Single-Population Beacon. Different power rates per number of SNPs queried from an unprotected beacon with a single population (EUR) by an adversary with different types of background knowledge: (Green) The attacker knows the allele frequencies of a population from the same ancestry (EUR) as the one in the beacon and performs queries following the rare-allele-first logic; (Red, Cyan, Blue and Purple) The attacker knows the allele frequencies of a population from an ancestry different from the one in the beacon and performs queries following the rare-allele-first logic (African (AFR), admixed American (AMR), East Asian (EAS) or South Asian (SAS), respectively); (Yellow) The attacker knows the allele frequencies of a distinct population with the same ancestry (EUR) other than the one in the beacon but performs queries in random order; (Black) The attacker does not have any information on allele frequencies (i.e., the original attack by Shringarpure and Bustamante [181]).

6.5.2 "Optimal" Re-Identification Attack in Multi-Population Beacon

Beacons often contain individuals coming from different ancestry groups. As a consequence, we further evaluated the attack based on real allele frequencies on a multi-population beacon and considered the case where an attacker might have only partial information about the different ancestries in the beacon. We set up a different beacon by removing individuals with EUR ancestry from phase 3 data set of the 1000 Genomes Project, and by selecting 1,235 random individuals from the remain ones. The resulting population is composed by individuals with African (AFR), Ad Mixed American (AMR), East Asian (EAS) or South Asian (SAS) ancestries. We picked 100 random samples from the beacon and 100 random samples not in the beacon a not of EUR ancestry to compose the query set.

As expected, results in Figure 6.3 show that also in the multi-population beacon the new re-identification attack based on allele frequencies is more effective than the one of Shringarpure and Bustamante. Especially, when the attacker knows the allele frequencies for a population with the same mix of ancestries of the individuals in the beacon (blue curve), 5 queries on average[1] are enough to obtain 100% of statistical power with 5% false-positive rate. As expected, with the same background knowledge but by querying alleles in random order, the attacker needs 750 more queries (azure curve) to obtain the same statistical power. A more realistic scenario is represented by the attacker knowing partial

[1] The attack is repeated on 100 different individuals.

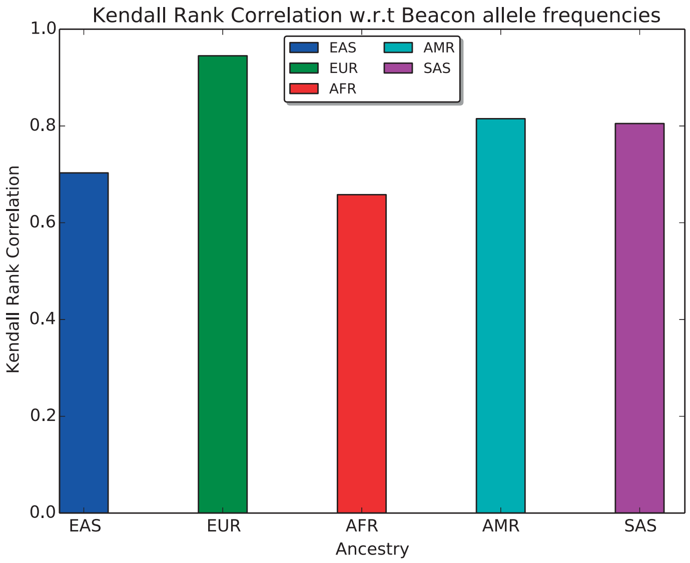


Figure 6.2: Kendall Rank Correlation Coefficient with respect to true beacon allele frequencies. Kendall rank correlation coefficient between the actual allele frequencies of the single-population beacon of Fig.1 and the allele frequencies of populations with different ancestries. Values closer to represent higher correlation. Colors mapping as in Figure 6.1.

(e.g., allele frequencies from a population with AFR ancestry) or unrelated information (e.g. allele frequencies from a population with EUR ancestry) about the ancestries in the beacon. In these cases, 100% of statistical power with 5% false-positive rate can be obtained with 20 (green curve) or 37 (red curve) queries, respectively.

6.5.3 "Optimal" Re-Identification Attack in Beacon with $S1$

We evaluated the proposed solution $S1$ by considering an attacker who knows the AFs of the 1000 Genomes Project and the value of threshold parameter k. As such, we set up a beacon as described at the beginning of Section 6.5 and computed the LRT statistic as in Equation (6.9). Figure 6.4 shows that, under such an attack, no individual in the beacon can be re-identified if a "*Yes*" answer is provided only when the queried allele appears at least $k = 2$ times in the database. Yet, the downside of this method is that only a fraction of the alleles that are in the beacon can be shared. For example, in the experimental beacon, only 60.30% of the alleles are shared by two or more individuals and thus can be shared; the queries to the remaining rare alleles ($\approx$ 40%) will receive a "*No*" answer even though they are actually present in the Beacon database.

6.5.4 "Optimal" Re-Identification Attack in Beacon with $S2$

To evaluate the effectiveness of $S2$ against an attack with background knowledge on AFs, we consider an attacker who knows the AFs of the 1000 Genomes Project and the value of the parameter ϵ. Figure 6.5 shows how the statistical power of the attacker decreases when different portions (ϵ) of unique alleles are hidden. When ϵ is set to be 0.001, the attacker has to query around 10^4 unique alleles to obtain a strong power of re-identification, compared to 200 queries for 100% re-identification when no random

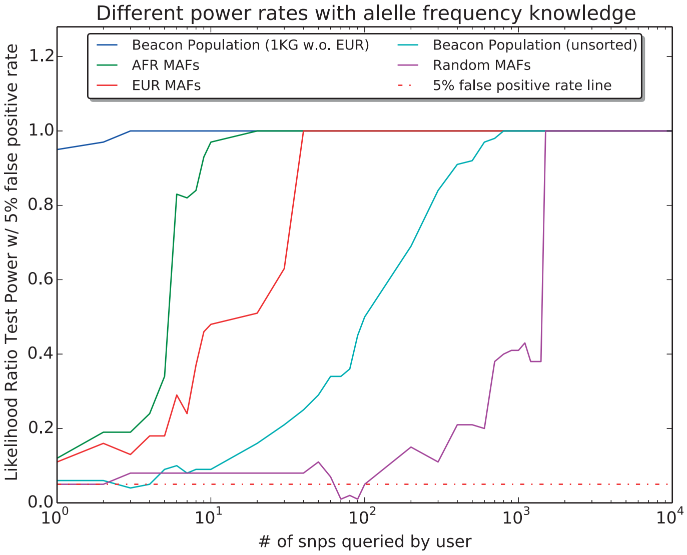


Figure 6.3: "Optimal" Re-Identification Attack in Multi-Population Beacon. Different power rates per number of SNPs queried from an unprotected multi-population beacon (the beacon contains individuals from all ancestry in the 1000 Genomes Project but the European ancestry) by an adversary with background knowledge on allele frequencies. Different colors represent different types of background knowledge.

masking on unique alleles (of $S2$ strategy) is applied. When $\epsilon \geq 0.15$, the re-identification power will not increase above $\approx 35\%$, which will keep the power at an acceptable risk level (i.e., relatively low confidence of re-identification).

6.5.5 Budget Evaluation in Beacon with $S3$

We evaluated strategy $S3$ with the same experimental setting as for $S1$ and $S2$. By default, we set $p = 0.05$, which means the statistical power of attack cannot exceed 0.95. Differently from experiments performed on solutions $S1$ and $S2$, which show an increase in re-identification risk given certain levels of utility of the beacon, we evaluate the efficacy of $S3$ by computing the decrease of utility across queries for a certain level of privacy loss.

To this purpose, we emulate the query behavior of a typical honest beacon user by generating queries based on the distribution of query frequency per allele frequency extracted from ExAC browser logs over a period of 12 weeks (data on beacon query frequencies were not available at the time of this work). During this time frame, a total of $1,345,291$ queries were asked on $934,680$ variants present in ExAC. Table 6.2 shows the proportion of queries and allele per range of allele frequencies (AF).

Fig.6.6 shows how the number of individuals with enough budget decreases with respect to the number of queries answered by the beacon. Note that the beacon's utility is completely preserved for the first $2,000$ queries.

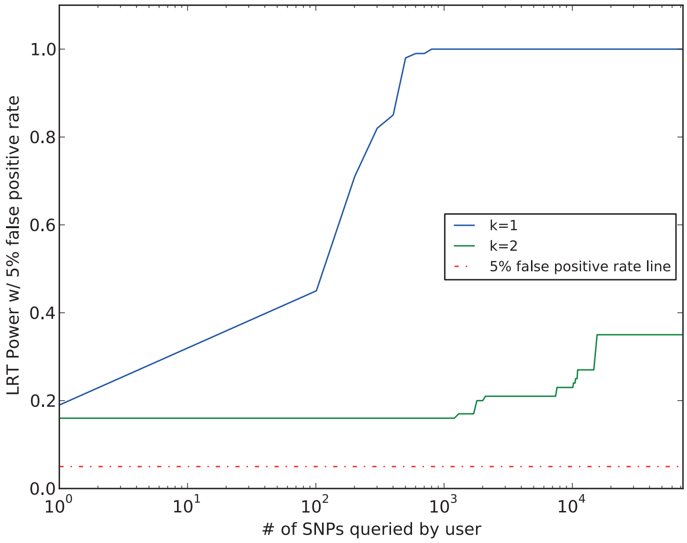


Figure 6.4: "Optimal" Re-Identification Attack in Beacon with S1. Different power rates per number of SNPs randomly queried from a beacon with mitigation S1 by an adversary with knowledge on and on allele frequencies from the 1000 genomes project: (Blue) $k = 1$; (Green) $k = 2$.

Allele Frequency	<0.001	0.001∼0.01	0.01∼0.05	0.05∼0.5	>0.5
Queries in ExAC	0.853	0.0.076	0.023	0.033	0.014

Table 6.2: Proportions of alleles and queries (over a period of 12 weeks) for each range of allele frequency.

6.6 Computational Complexity Evaluation

The first and second strategies induce very little overhead. The allele frequencies can be pre-calculated, which takes only linear time to the size of the database, and kept as a table in the database. Once k or ϵ is pre-determined, the beacon will just need to check if the query allele's frequency is smaller than k (for strategies $S1$ and $S2$) and to generate a random number (for $S2$) before composing a response of *"Yes"* or *"No"*. For mitigation strategy $S3$, we can easily compute the complexity of the proposed Algorithm 11. Suppose there are N individuals in the dataset, then for given a query, we need to:

1. Compute the risk of query, which can be done in constant time $O(1)$,
2. Check whether there are individuals that have the queried allele and a budget greater than the risk, $O(0)$,
3. If there is no such person, answer *"No"*, which can be done in constant time $O(1)$,
4. If there is at least one, answer *"Yes"*, then reduce those people's budget by the risk. This can be done in linear time $O(N)$.

So in total, the computational time required for each query on a beacon with mitigation strategy $S3$ is linear with respect to the number of individuals in the beacon. We

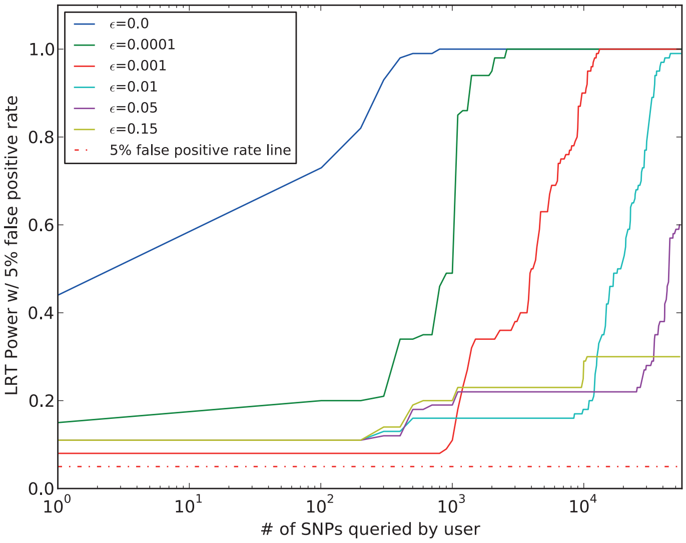


Figure 6.5: "Optimal" Re-Identification Attack in Beacon with S2. Different power rates per number of SNPs queried (with rare-first logic) from a beacon with mitigation S2 by an adversary with knowledge on ϵ and on allele frequencies from the 1000 genomes project. Different colors for different values of ϵ.

note that the required time for $S3$ is the same as if no privacy-preserving mechanisms were imposed.

6.7 Discussion

In this chapter, we have analyzed in detail the beacon re-identification attack originally proposed by Shringarpure and Bustamante and a new and "optimal" version of it by considering a smarter adversary who makes use of public information on AFs. We evaluated the power of our new attack through several experiments on real data by considering different conditions of adversarial background knowledge. Our results show that our attack always outperforms the original SB attack. As one might expect, we have observed that the power of an adversary's re-identification attack is directly related to the completeness and accuracy of the adversary's knowledge of the AF of the targeted Beacon. As already analyzed by Shringarpure and Bustamante, the underlying LRT test can be extremely harmful when a beacon is linked to sensitive phenotypes. Yet it is important to emphasize that, although our attack further reinforces SB's concern, the re-identification risk is relative to each beacon. These attacks fundamentally rely on the assumption that the attacker already has access to the genome of the victim.

Despite such a strong assumption, several research efforts in genomic privacy have studied the problem of re-identification of membership in genetic databases and have shown that it is extremely hard to prevent and sometimes even impossible [79].

Based on the "optimal" re-identification attack, we have proposed three different strategies aimed at effectively thwarting beacon membership re-identification. Because the accuracy of the beacon re-identification attack depends on the power and false positive rate of the LRT test, the probability that a test behaves correctly (rejecting the null hypothesis when it is false and failing to reject when it is true) is

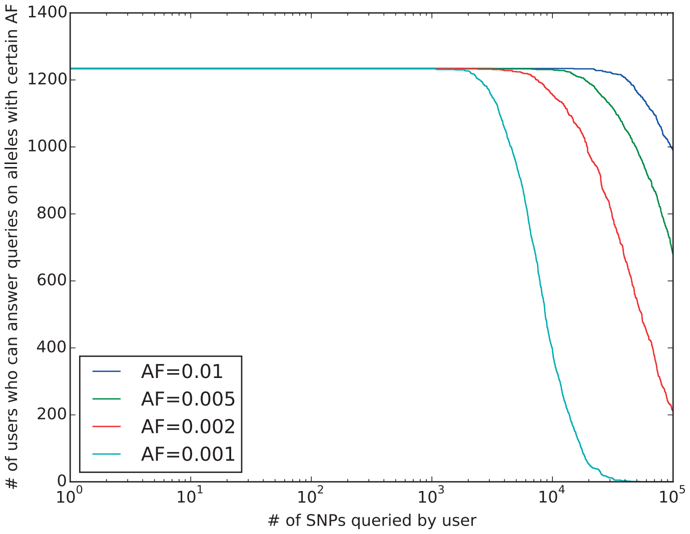


Figure 6.6: Budget Evaluation in Beacon with S3. Behaviors of individual budgets per number of SNPs queried according to the typical user's query profile obtained from ExAC log data. The cyan curve represents the number of individuals with enough budget to answer "Yes" to queries targeting alleles with AF=0.001. Red, Green and Blue curves correspond to 0.002, 0.005, 0.01, respectively.

Strategy	**Disadvantages**	**Advantages**
$S1$: Beacon Alteration	Eliminates possibility of querying for unique alleles highly likely to be most useful in genetic research	Protects privacy of individuals possessing variants most likely to be targeted by attackers
$S2$: Random Flipping	Decreases rate of true answers returned from querying unique alleles likely to be useful in genetic research	Permits some unique alleles to be discoverable and to fine-tune the privacy-utility trade-off
$S3$: Query Budget per Individual	Requires the assumption of Beacon user being non-anonymous and holding no more than one Beacon account; may require complicated accounting scheme	Enables all alleles to be discoverable until budget is exceeded

Table 6.3: Summary of advantages and disadvantages of the three proposed mitigation strategies.

given by: Power $*$ (Probability of alternative hypothesis) $+$ $(1 -$ False positive rate) $*$ (Probability of null hypothesis). From the perspective of a beacon administrator, the attacker's test should be incorrect most of the time; i.e., power should be low and/or the false positive rate should be high. The three proposed strategies all address the mitigation problem by controlling the power or the false positive rate. The first (S1) and second (S2) strategies reduce the power to nearly zero when the LRT must have a small false positive rate, whereas in the third solution (S3), the test always has 100% power but a high false positive rate. In particular, S1 and S2 directly alter the beacon to reduce the inference power of the attacker, whereas S3 introduces a new idea of personal budget that decreases when the genome of the individual is used to positively answer a query.

Results of our experiments have shown that all proposed mitigation strategies have advantages and disadvantages, as summarized in Table 6.3. $S1$ effectively mitigates the attack by keeping the power of the LRT to 0.2 if all unique alleles are flipped. Yet, it generates a significant loss in utility of the beacon because the majority of the queries of a typical user of beacon usually target rare alleles. We define the utility of a beacon as the proportion of true answers it can provide. $S2$ can be considered a more sophisticated version of $S1$ because it flips only a portion of unique alleles, affording a more fine-grained control over the utility vs. privacy trade-off. The attack inference power can be confined to a secure level by masking only 15% of unique alleles (which means a drop in utility of 6% against 40% of $S1$). Note that the utility of a beacon adopting $S1$ or $S2$ is fixed a priori and does not change along with the power of the attack.

Finally, results of experiments on $S3$ show that, given a certain assurance level ($p = 0.05$), the beacon utility is completely preserved for the first $2,000$ queries. Yet, $S3$ relies on the assumption that the beacon system is not anonymous and has a controlled level of access with user authentication and identity proofing. Based on data collected from the ExAC browser logs, a budget of $2,000$ query per beacon user seems a reasonable compromise between privacy and utility.

Preventing inference attacks on large databases is widely known to be one of the most daunting of database security challenges [20]. This fact has been a major consideration in the development of GA4GH's framework for responsible sharing of genomic and health-related data, privacy and security policy, and security infrastructure. Effective risk management must leverage policy, technology, and community governance to address re-identification risks. Effective risk management is fundamental to facilitating and promoting data sharing across the GA4GH global community. We emphasize that security and privacy are components of risk management. Technical risk-management strategies such as those proposed in this chapter are practical and can be adapted according to the context of each beacon. Therefore, they represent a valuable set of options for assessing and mitigating risk within the GA4GH community.

6.8 Summary

The risk of re-identification based on the binary yes/no allele-presence query responses was initially adjudged as acceptable in the GA4GH Beacon Project. However, recent work demonstrated that, given a beacon with specific characteristics (including relatively small sample size, and an adversary who possesses an individual's partial genome sequence), the individual's membership in a beacon can be inferred through repeated queries for variants present in the individual's genome. In this chapter, we have improved upon the initial attack by considering a smarter attacker exploiting background information on allele frequencies and we proposed three practical strategies for reducing re-identification risks in beacons. The first two strategies manipulate the beacon such that the presence of rare alleles is obscured; the third strategy budgets the number of accesses per user for each individual genome. Using a beacon containing data from the 1000 Genomes Project, we demonstrated that the proposed strategies can effectively reduce re-identification risk in beacon-like datasets.

Chapter 7

MedCo: Enabling Privacy-Conscious Exploration of Distributed Clinical and Genomic Data

In the previous chapter, we have shown how to thwart the re-identification risk in beacons-like systems for genomic variant discovery. Yet, variants discovery represents only the first step in the process of identifying data sets of interest for particular research studies. In this chapter, we move one step forward by proposing a new solution enabling the privacy-conscious exploration of distributed clinical and genomic data necessary to identify cohorts of well-characterized individuals to be included in clinical or population health studies.

7.1 Introduction

With the increasing digitalization of clinical and genomic information, data sharing is becoming the keystone for realizing the promise of personalized medicine to its full potential. Several initiatives, such as the Patient-Centered Clinical Research Network (PCORNet) [180] in USA, eTRIKS/TranSMART [27] in EU, the Swiss Personalized Health Network (SPHN) [190] in Switzerland, and the Global Alliance for Genomics and Health (GA4GH) [195], are laying down the foundations for new biomedical research infrastructures aimed at interconnecting (so far) siloed repositories of clinical and genomic data.

In this global ecosystem, the ability to provide strong privacy and security guarantees in order to comply with strict regulations (e.g., HIPAA [**?**] in USA or the new GDPR [80] in EU) is crucial, yet extremely challenging to achieve, for biomedical research to be able to scale up. The number of health-data breaches constantly increases [203] and there is significant public pressure to ensure that the privacy and security of the data can be properly protected. Yet, because of the current cultural gap between the medical and the privacy/security communities, currently deployed technical solutions enabling the sharing of medical and genomic data still provide very limited guarantees in this sense. This constrains researchers to access very limited medical information, often of relatively low interest for research. For example, the Beacon Network of the GA4GH [67] can provide only presence/absence information of a given variant in a distributed database, and the SHRINE system [213] enables a researcher to access only aggregate information (e.g., the number of patients satisfying specific research criteria) from HIPAA-compliant “limited data sets” that exclude any sort of genetic or identifying clinical information. As a result, the development of new technologies that (i) are compliant with regulations, (ii) allow sharing data also beyond the “limited data set”, and (iii) can be easily integrated on top of existing systems, is now more urgent than ever for medical research.

We address this challenge by introducing MedCo, the *first operational system* that enables the privacy-preserving exploration of distributed sensitive (and identifying) medical data by using strong collective encryption. Its purpose is to foster data sharing by distributing trust among different medical institutions that want to expose their data to external queries, in a way that is compliant with regulations. To achieve this, MedCo takes the best of both worlds (medical informatics and IT privacy/security) by building on top of existing and well-established open-source technologies (i) for clinical data exploration, i2b2 [146] and SHRINE [213], and (ii) for distributed and secure data processing, UnLynx [89]. In particular, MedCo enables medical institutions to federate and collectively encrypt their sensitive clinical and genetic data with homomorphic encryption in order to protect them against undesired and illegitimate access (e.g., hackers or insiders), and to enable their exploration through a set of secure distributed protocols.

In light of its low overhead, MedCo can dramatically accelerate and partially replace IRB review processes for sharing sensitive (and identifying) medical data with external researchers. These review processes can take several weeks, if not months, to permit researchers to access the data, and they are often denied because the necessary privacy and security guarantees cannot be provided. As such, MedCo paves the way to new and unexplored use-cases where, for example, (i) researchers will be able to securely query massive amounts of distributed clinical and genetic data that go beyond the “limited data set” category and to obtain descriptive statistics indispensable for generating new hypotheses in clinical research studies, or (ii) clinicians will be able to find patients with similar (possibly identifying) characteristics to those of the patient under examination in order to take more informed decisions in terms of diagnosis and treatment.

In summary, in this chapter, we make the following contributions:

- We introduce MedCo, the first operational system enabling the sharing of sensitive clinical and genomic information based on state-of-the-art open-source technologies.

- We extensively tested MedCo in a simulated federation of three sites, focusing on a clinical-oncology case with public somatic DNA and lung cancer data.

- We propose a new generic method to add dummies in order to mitigate frequency attacks that can incur when probabilistically encrypted data are transformed to deterministically encrypted data for the sake of enabling Boolean queries.

We summarize the notations used in this Chapter in Table 7.1.

Notation	Description
HBC	Honest-but-curious
MBC	Malicious-but-covert
DDT	Distributed Deterministic Tag
DVS	Distributed Verifiable Shuffling
DKS	Distributed Key Switching
K	Public key for the ElGamal encryption scheme
k	Secret key for the ElGamal encryption scheme
$E_K(m)$	ElGamal encryption of message m with public key K
$E_K(m) = (C_1, C_2)$	ElGamal ciphertext tuple
$\mathbb{X}$	Set of privacy-sensitive ontology concepts
(k_u, K_u)	ElGamal key pair (secret and public keys) for user u
s_i	Symmetric secret key for site i
$DT_s(m)$	Deterministic tag of message m under symmetric secret s
ϵ_u	Initial differential privacy budget for user u
ϵ_q	Differential privacy budget allocated for a given query q
f_i^j	Flag for patient j at site i indicating if the patient is “real” ($f_i^j = 1$) x or “dummy” ($f_i^j = 0$)
R_i	Patient count at site i satisfying the query
ϕ	Set of patients satisfying the query
$\hat{R}_i$	Obfuscated patient count at site i satisfying the query

Table 7.1: Notation used throughout the chapter.

7.2 MedCo Ecosystem

In this section, we introduce the ecosystem in which MedCo operates. We start by describing the system and threat models. We then define the functionality and privacy/security requirements that MedCo must satisfy.

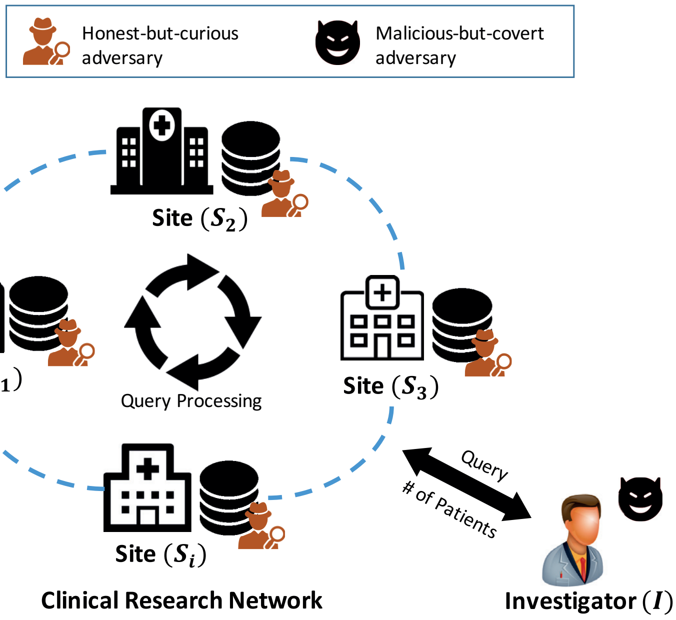


Figure 7.1: **MedCo Ecosystem.** System model, including several clinical sites and an investigator; threat model, including honest-but-curious adversaries at the clinical sites and a malicious-but-covert adversary at the investigator.

7.2.1 System Model

We consider the system model depicted in Figure 7.1, where medical institutions (or sites) are organized in a decentralized federation (or network) and collaborate to share clinical and genomic data without relying on any central third party or authority. This is the typical model of existing clinical networks such as the GA4GH Beacon Network [67] and Matchmaker Exchange [154], or most of the PCORNet Clinical Data Research Networks [180]. As opposed to the centralized model, our model ensures several advantages such as, for example, the absence of a single point of failure, increased transparency and local control. Each site's data are maintained separately and data access can be monitored by each different institution. In particular, the proposed system consists of the following entities:

- Clinical sites (S_i) such as research institutions, universities or hospitals that own clinical and genomic data and are willing to share them with internal (i.e., affiliated with one of the clinical sites in the federation) and external investigators. The sites are responsible for securely storing the data and for collectively processing incoming queries in a privacy-preserving way. We assume that each clinical site is internally organized into two main departments: (i) a clinical care department where clinical and genomic data are generated by patient encounters and stored in a private electronic health record (EHR) system and (ii) a clinical research department where a subset of data are imported from the EHR system into a research data warehouse that can be exposed to external researchers or investigators for the purpose of data sharing.

- A medical investigator (I) who is interested in exploring the distributed data stored at the different sites. Her main goal is to use MedCo for generating and validating research hypotheses or identifying cohorts of interest to then ask each site for the authorization for obtaining the patients' raw individual data for further analyses.

7.2.2 Threat Model

We consider two main types of threats: a *honest-but-curious (HBC)* adversary at the clinical sites and a *malicious-but-covert (MBC)* adversary impersonating the investigator:

- Clinical sites: We assume the clinical care department at each site S_i to be trusted as, in general, it is not directly exposed to external agents and their EHR systems can only be accessed by a limited number of authorized employees (e.g., physicians, nurses). However, we consider each site's research department to be HBC, as data stored in the research data warehouse must be exposed to queries from external parties for the sake of data sharing. These sites are trusted to store correct information in their data warehouses and to honestly follow the MedCo core protocol. Yet, they do not necessarily trust each other because they might be compromised by external or internal attackers willing to infer sensitive information about the individuals whose data are stored in their databases. For example, a hacker can enter into a research department's information system, by exploiting a vulnerability in the software or by a social-engineering attack, and illegitimately access the data stored in the clinical research data warehouse or infer other sites' sensitive information that is being processed during the MedCo protocol. Similarly, an insider with legitimate access to the clinical research data warehouse can try to steal sensitive information from its own site or from the others and then sell it to the best offer on the black market, or put it on the Web in order to ruin the reputation of renowned medical institutions.

- Investigator: We assume the investigator to be MBC. An authorized investigator might infer sensitive information stored at the different clinical sites by performing consecutive queries in order to exploit the information leaked by the end-results. For example, a malicious investigator can re-infer the presence of a known individual into a sensitive cohort (e.g., patients who are HIV-positive) or reconstruct a subset of the database itself.

We assume, however, that (i) all sites but one can collude or be compromised simultaneously, (ii) investigators have been identity proofed, hold a single account and do not collude with each other nor with any clinical site[1]; This last assumption appears reasonable in practice as, in order to collude, a user needs by definition to involve someone else. Finally we assume also that queries are logged into a distributed immutable ledger.

7.2.3 Requirements

To meet end-users expectations and be compliant with regulations, MedCo must satisfy the following requirements in terms of functionality and privacy/security features.

- Functionality. MedCo must provide at least the same functionalities as state-of-the-art systems for cohort exploration on distributed data (e.g., the SHRINE [213]) in order to enable the same use cases (e.g., feasibility studies or cohorts identification). In particular, an investigator must be able to run queries in MedCo by logically combining clinical and genomic concepts encoded by a medical ontology and obtain the number of patients per site satisfying the research criteria. Also, MedCo must enable different query breakdowns such as distribution of patient counts per age, gender, ethnicity. More formally, an investigator must be able to perform aggregate SQL queries such as "`COUNT(patients) FROM dataset WHERE * AND/OR * GROUP BY *;`" and selection SQL query such as "`SELECT(patients) FROM dataset WHERE * AND/OR * GROUP BY *;`" where '`*`' represents any possible concepts/codes in the ontology.

- Security/Privacy. MedCo must enable sites to protect the confidentiality of their sensitive data, such as identifying health information or genomic data at rest, in transit and during computation while avoiding a single point of failure in the system. Also, only the investigator issuing the query is allowed to obtain the query end-result. MedCo can also ensure unlinkability by providing a mechanism that prevents the investigator from tracing a query response back to its original site. Optionally, MedCo could enable the prevention of inferences from end-results of subsequent queries about the presence or absence of an individual in one of the databases, in order to guarantee, for instance, differential privacy.

Depending on the trustworthiness of the investigator querying the system, MedCo should enable for a modular enforcement of the above-mentioned privacy/security guarantees. For example, MedCo could release either obfuscated and unlinkable query results, exact query results, or individual patients' records.

[1] We note that this assumption permits an investigator to be an employee of one of the federated clinical sites but prevents her from having direct access to the clinical-research data warehouse.

7.3 MedCo Building Blocks

MedCo is the first operational system that combines established open-source technologies from both the biomedical informatics community (i2b2 [146] and SHRINE [213]) and the privacy and security community (UnLynx [89]) in order to enable privacy-preserving sharing of clinical and genomic distributed data. In this section, we provide a high-level description of these technologies and their main features that we use as MedCo building blocks.

7.3.1 Data Model from i2b2

Informatics for Integrating Biology and the Bedside (i2b2) [146] is the state-of-the-art clinical platform for enabling secondary use of electronic health records (EHR) [146]. It is designed to enable investigators to perform queries on an enterprise data-repository in order to find sets of patients that would be of interest for further clinical research studies.

We chose this platform for storing data and building queries in MedCo because of (i) its flexible data model, (ii) its popularity,[2] (iii) its extendability through the design of new plug-ins, and (iv) its intuitive end-user interface enabling the easy generation of clinical queries. Indeed, i2b2 consists of a simple and flexible relational data-model based on a "star schema" (see Fig. 7.2) and a set of server-side software modules, called "cells," which are responsible for the business logic of the platform and are organized in a "HIVe." The data model stores, in a narrow table called `observation_fact` table, clinical observations (or "facts") about patients such as diagnoses, medications, procedures, and demographics, along with a date, a patient identifier and an encounter identifier. Each observation is encoded by an ontology concept from a medical terminology, such as the International Classification of Disease (ICD) or the US National Drug Code (NDC). The use of extendable ontologies makes i2b2's model highly adaptable to site-specific coding and easily deployable on top of existing EHR systems. Besides the `observation_fact` table, there are four other "dimensions" tables that further describe patients' data and meta-data. Queries are built in a Web-based query system by combining ontology codes, organized in a hierarchical tree-based structure, with logical ORs and ANDs operators. Queries are executed as SQL statements by the data repository (or CRC) cell that returns the aggregate number of patients meeting the research criteria.

7.3.2 Interoperability Layer from SHRINE

The Shared Health Research Information Network (SHRINE) [213] is the state-of-the-art framework that enables investigators to search patients' data from the 'limited data set (LDS)' across multiple independent clinical sites. SHRINE is currently deployed in at least six networks in the United States. It is built on top of i2b2 and its purpose is to connect distributed i2b2 instances from various clinical sites through an interoperability layer based on a common ontology. Such a common ontology is translated into each local site's ontology at query time, thus hiding the complexity of the local databases from the rest of the network. SHRINE comprises three main components:

- The *Adapter*: It is a Web-service designed as an i2b2 cell in order to be fully integrated within the i2b2 HIVe. It translates the investigator's query made through the

[2] i2b2 is used by more than 200 institutions worldwide covering more than 250 millions patients data.

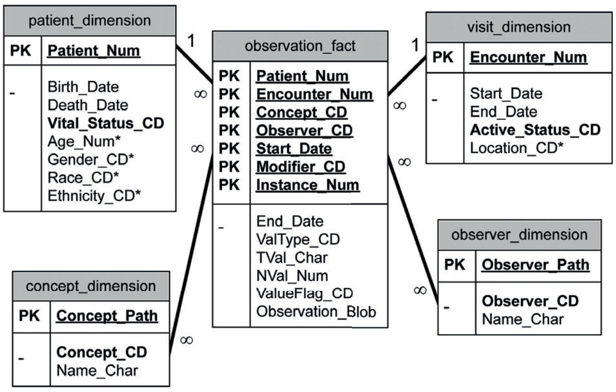


Figure 7.2: **i2b2 data model [146].** It consists of a central `observation_fact` table and four dimension tables. Each row of the `observation_fact` table stores one clinical observation for a given patient. 'PK' stands for 'primary key' while 'CD' stands for 'code'.

common SHRINE ontology into a format that matches the site's local databases. In a fully decentralized network, an adapter must be deployed at each clinical site that uses SHRINE.

- The *Query Aggregator*: It is responsible for (i) broadcasting the investigator's query to each of the adapters in the network and (ii) receiving from each clinical site the count of patients satisfying the query to be sent back to the investigator. At least one clinical site in the network must deploy a query aggregator that will serve as query entry-point in the system.

- The *Web Client*: It provides a Web-based user interface through which the investigator can access the system and build i2b2-type queries by using the common SHRINE ontology. The Web client must be deployed together with the Query Aggregator.

In our system model, we consider that each clinical site in the network is provided with all three SHRINE components so that MedCo is fully decentralized and each site can serve as a query entry point. We note that SHRINE does not provide any form of confidentiality protection for the data stored at the different sites as it only relies on basic access control and query-result obfuscation to mitigate the risk of re-identification.

7.3.3 Privacy-Preserving Distributed Protocols from UnLynx

UnLynx is the latest and most advanced general framework for privacy-conscious sharing of distributed sensitive data [89]. Its purpose is to enable a set of data providers to collectively protect the confidentiality of their sensitive data in the *anytrust* threat model [214], by encrypting them with a collective public key generated by a group of independent servers forming a collective authority (or "cothority"). Due to the use of additively homomorphic encryption (ElGamal on elliptic curves), users can still perform simple statistical queries directly on the encrypted data by relying on a set of secure distributed protocols run within the cothority. When a query comes to UnLynx, each data provider uploads the requested ciphertexts to the cothority, that securely processes them in order to obtain an encrypted query result. Such a result can eventually be decrypted only by the user who issued the initial query. UnLynx is designed to be modular, allowing the addition and removal of security/privacy features depending on the performance and security requirements. MedCo relies on three of the UnLynx main protocols. In the following, we assume elliptic curve notation where $\mathcal{E}$ denotes an elliptic curve over the prime field $\mathbb{GF}(p)$ and G designates its base point. We denote as $E_K(m)$ the ElGamal probabilistic encryption of a message m under a public key $K = kG$, where k is the secret key.

- Distributed Deterministic Tag (DDT) Protocol. The DDT protocol enables a set of n cothority servers to tag (with deterministically encrypted values) data probabilistically encrypted under the cothority collective key, without ever decrypting them. The purpose of this protocol is to enable equality-matching queries on probabilistically encrypted data that otherwise would not be possible. Let $E_K(m) = (C_1, C_2) = (rG, m + rK)$ be the encryption of a message m with the collective public key K. The DDT protocol consists of two cothority-rounds. In the first round, each server sequentially generates a fresh secret s_i and adds a value derived from its secret s_iG to C_2. After this first round, the resulting ciphertext is $(\tilde{C}_{1,0}, \tilde{C}_{2,0}) = (rG, m + rK + \sum_{i=1}^{n} s_iG)$. In the second round, each server partially and sequentially modifies this ciphertext. More specifically, when server S_i receives the modified ciphertext $(\tilde{C}_{1,i-1}, \tilde{C}_{2,i-1})$ from server S_{i-1}, it computes $(\tilde{C}_{1,i}, \tilde{C}_{2,i})$, where $\tilde{C}_{1,i} = s_i\tilde{C}_{1,i-1}$ and $\tilde{C}_{2,i} = s_i\left(\tilde{C}_{2,i-1} - \tilde{C}_{1,i-1}k_i\right)$. At the end of the second round, the deterministically encrypted tag is obtained by keeping only the second component of the resulting ciphertext $DT_s(m) = C_{2,n} = sx + \sum_{i=1}^{n} s_isB$, where $s = \prod_{i=1}^{n} s_i$ is a short-term collective secret corresponding to the product of each server's fresh secret.

- Distributed Verifiable Shuffling (DVS) Protocol. The DVS protocol enables a set of cothority servers to sequentially shuffle probabilistically encrypted data so that the outputs cannot be linked back to the original ciphertexts. More specifically, the DVS protocol uses the Neff shuffle [150]. It takes in input multiple sequences of ElGamal pairs $(C_{1,i,j}, C_{2,i,j})$ forming a $a \times b$ matrix and outputs a shuffled matrix of $(\bar{C}_{1,i,j}, \bar{C}_{2,i,j})$ pairs such that for all $1 \le i \le a$ and $1 \le j \le b$, $(\bar{C}_{1,i,j}, \bar{C}_{2,i,j}) = (C_{1,\pi(i),j} + r''_{\pi(i),j}B, C_{2,\pi(i),j} + r''_{\pi(i),j}P)$, where $r''_{i,j}$ is a re-randomization factor, π is a permutation and P is a public key.

- Distributed Key Switching (DKS) Protocol. The DKS protocol enables a set of cothority servers to convert a ciphertext generated with the collective public key of the cothority into a ciphertext of the same data generated under any known public key, without ever decrypting them. The DKS protocol never makes use of decryption. Let $E_K(m) = E_K(m) = (C_1, C_2) = (rG, m + rK)$ be the encryption of a message m with the collective public key K. The DKS protocol starts with a modified ciphertext tuple $(\tilde{C}_{1,0}, \tilde{C}_{2,0}) = (0, C_2)$. Then, each server partially and sequentially modifies this element by generating a fresh random nonce v_i and computing $(\tilde{C}_{1,i}, \tilde{C}_{2,i})$ where $\tilde{C}_{1,i} = \tilde{C}_{1,i-1} + v_iB$ and $\tilde{C}_{2,i} = \tilde{C}_{2,i-1} - rK_i + v_iU$. The resulting ciphertext corresponds to the message m encrypted under the public key U, $(\tilde{C}_{1,n}, \tilde{C}_{2,n}) = (vB, m + vU)$ from the original ciphertext (C_1, C_2), where $v = v_1 + \ldots + v_n$.

7.4 MedCo Core Architecture & Protocol

In this section, we provide a detailed description of MedCo (Figure 7.3). We begin by explaining the system initialization and the data ingestion phases in which clinical sites collectively encrypt their sensitive data and store them in the i2b2 data model. We then describe the secure query workflow that enables an investigator to efficiently query the encrypted data stored in independent i2b2 databases.

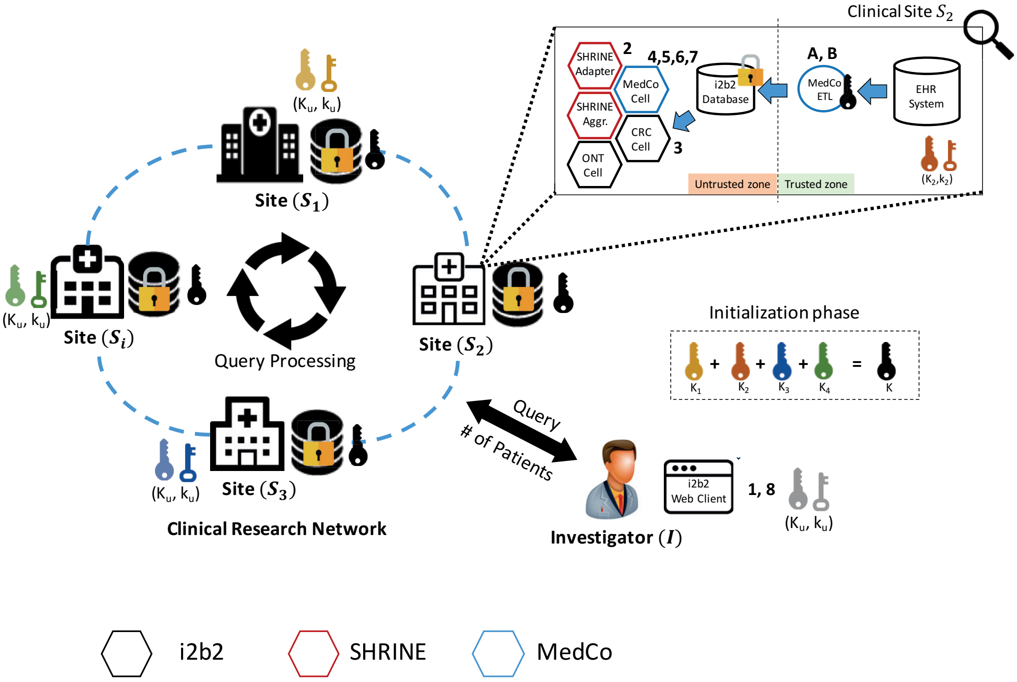


Figure 7.3: **MedCo Core.** High-level representation of MedCo core phases with 4 clinical sites including *system initialization*, *ETL phase* (steps A and B) and *secure query workflow* (steps 1-8).

7.4.1 System Initialization

Clinical sites store unencrypted patient-level clinical and genomic data in their private EHR systems and are willing to share these data by securely exposing them to internal/external investigators through an i2b2 research data warehouse. Therefore, we assume that each site has already deployed an instance of the i2b2 HIVe on top of its own data warehouse and that SHRINE is also installed in order to provide the interoperability layer necessary for connecting data encoded with local ontologies at the different sites. During this initial phase, each clinical site (S_i) generates a pair of ElGamal cryptographic key (k_i,K_i), where $K_i = Gk_i$, along with a symmetric secret s_i. Then, all sites combine their ElGamal public keys in order to generate a single collective public key $K = \sum_i K_i$ that will be used to encrypt the data.

7.4.2 Data Extraction Transformation and Loading

During the data-ingestion phase, a.k.a. *extraction transformation and loading (ETL)*, each site extracts patient-level data from its private EHR system and transforms them by following the i2b2 data model representation in Figure 7.2. In the i2b2 data model, the private information that must be protected from an untrusted third party consists of the set of clinical observations that are considered to be sensitive or identifying for a given patient. Those are usually represented by a subset, $\mathbb{X}$, of ontology concepts encoded with ontology codes (`Concept_CD`) in the `observation_fact` table. Hence, before loading the data into the i2b2 data warehouse, each site starts an encryption phase consisting of two steps:

A. Generation of dummy patients: Each site generates a set of dummy patients with plausible clinical observations specifically chosen so that the distribution of ontology codes across patients, in the `observation_fact` table, is as close as possible to the uniform distribution. We explain the rationale behind such a step in detail in Section 7.5. To distinguish the real patients from the dummies, each site also generates a binary flag to be appended to the demographic information in the `patient_dimension` table. Such flag is set to 1 for real patients and to 0 for dummy patients.

B. Data encryption: In order to protect patients' sensitive observations that are stored in the `observation_fact` table, each site deterministically encrypts the ontology codes in $\mathbb{X}$, by running on each of them a two-round UnLynx DDT protocol in which each site in the network uses its secret s_i. As a result of this protocol, the sites obtain the corresponding deterministic tag, $DT_s(x)$ for each code $x \in \mathbb{X}$, where $s = \sum_i s_i$. Along with the deterministic encryption of the sensitive ontology codes, each site also encrypts the patients' flags to be stored in the *patient_dimension* table, by using the probabilistic ElGamal encryption algorithm with the collective key K.

After this ETL phase, the i2b2 databases at the different sites contain "non-sensitive" ontology codes in cleartext, and "sensitive" ontology codes protected with deterministic encryption for both real and dummy patients. Probabilistically encrypted flags are stored to keep track of dummy patients and to make them indistinguishable from real patients. Clinical sites make use of dummies in order to thwart frequency attacks from honest-but-curious adversaries who aim at breaking the deterministic encryption when the distribution of ontology codes is not uniform.

7.4.3 Secure Query Workflow

We assume each investigator that uses MedCo has a pair of ElGamal cryptographic keys (k_u, K_u) and, optionally, an initial differential privacy budget ϵ_u. The purpose of such a budget is to limit the number of queries an investigator can run on the system so that ϵ_u-differential privacy can be guaranteed. The proposed query workflow is illustrated in Figure 7.3 and comprises the following steps:

1. Query Generation: The query generation takes place in the SHRINE Web client with an authenticated investigator who selects "sensitive" and "non-sensitive" concepts from the common SHRINE ontology and combines them with AND/OR logical operators in order to build a Boolean query. Once the query is built, the "sensitive" concepts are probabilistically encrypted by the Web client with the collective key K, whereas the "non-sensitive" ones are left in cleartext. The resulting query is sent along with K_u to the SHRINE query aggregator of the preferred clinical site.

2. Query Analysis: From the SHRINE query aggregator of the first clinical site, the query is broadcasted to all the SHRINE adapters installed at the other sites in the network. At each site, the query is translated into the local ontology by the SHRINE adapter and subsequently analyzed by a new MedCo cell that extracts the encrypted (hence "sensitive") codes from the query.

3. Query Processing: Once the encrypted ontology codes are extracted, the MedCo cell at each site runs an UnLynx DDT protocol on them in order to obtain the corresponding deterministic encrypted tags (as in the ETL phase). These tags, along with the unencrypted codes in the initial query, are then forwarded to the standard i2b2 Data Repository (CRC) Cell. The CRC cell uses them to fetch, from the i2b2 database, the set of patient numbers and probabilistically encrypted flags corresponding to the patients (real and dummy) that match the Boolean predicate in the initial query. Equality matching between the encrypted codes in the query and those in the `observation_fact` table is enabled by the deterministic nature of the encrypted tags that preserves the equality property in the ciphertext domain.

4. Result Aggregation (optional): Once the patient numbers and the encrypted flags are fetched from the local i2b2 database, the MedCo cell homomorphically aggregates the flags in order to obtain the encrypted local patient count $E_K(R_i)$ at each site. Because of the null contribution of their encrypted flags (i.e., $E_K(0)$), dummy patients are cancelled out from the local patient-count during the aggregation. Let $E_K(f_i^j)$ be the encrypted flag of the j-th patient in site S_i, then $E_K(R_i) = E_K(\sum_{j \in \phi} f_i^j) = \sum_{j \in \phi} E_K(f_i^j)$, where ϕ is the set of patients satisfying the query.

5. Result Obfuscation (optional): In order to guarantee differential privacy, the MedCo cell obfuscates the encrypted local patient-count by homomorphically adding noise sampled from a Laplacian distribution. More specifically, let ϵ_q be the privacy budget allocated for a given query q and μ be the noise value drawn from a Laplacian distribution with mean 0 and scale $\frac{\Delta f}{\epsilon_q}$, where the sensitivity Δf is equal to 1 due to R_i being a count. The encrypted obfuscated query result is obtained as $E_K(\hat{R}_i) = E_K(R_i + \mu) = E_K(R_i) + E_K(\mu)$. We note that the query result is released to the investigator only if the investigator's differential privacy budget is enough for such a query, i.e., if $\epsilon_u - \epsilon_q > 0$.

6. Result Shuffling (optional): In order to break the link between the encrypted obfuscated query results generated and the sites having generated them, the MedCo cell of the site that initially broadcasted the query starts an UnLynx DVS protocol on all the local encrypted and obfuscated patient counts. As a result of the protocol, each site receives back an encrypted obfuscated patient count, possibly generated by one of the other sites.

7. Result Re-Encryption: The local encrypted (shuffled and obfuscated) query results $E_K(\hat{R}_i)$ are computed at each site under the collective key K, so they must be re-encrypted under the investigator's public key K_u so that she can decrypt them. To this purpose, each site runs an UnLynx DKS protocol in order to obtain $E_{K_u}(\hat{R}_i)$. Then the SHRINE adapter at each site sends $E_{K_u}(\hat{R}_i)$ back to the initial SHRINE query aggregator.

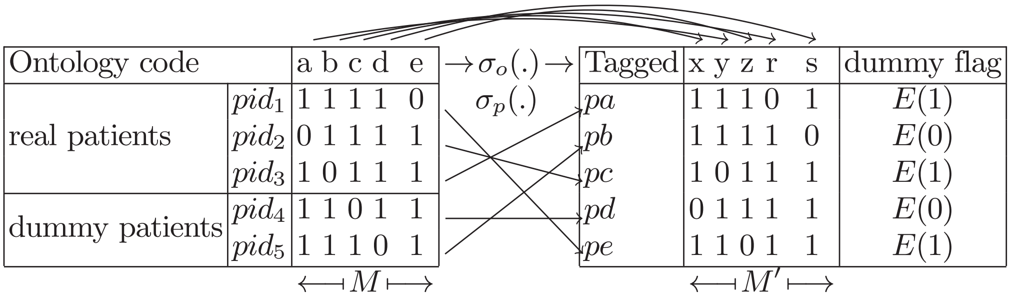
8. Result Decryption: Once the SHRINE query aggregator receives the encrypted query results from the different sites in the network, it sends them back to the Web client for decryption with the investigator's secret key k_u.

We note that, depending on the trustworthiness level of the investigator, steps 4, 5 and 6 can be skipped and patient numbers and encrypted flags can be directly released to the Web client. As such, the investigator will be able to rule out dummy patients from each site by checking the corresponding flags and use the real patient numbers for directly contacting sites and obtaining individual patient records.

7.5 Dummy-Addition Strategies

For cohort-exploration queries, the deterministic encryption of the ontology codes applied during the ETL phase (see Section 7.4.2) avoids dictionary attacks by any subset of colluding HBC sites due to the distribution of the secrets s_i used in the DDT protocol. Nevertheless, a *Dummy-Patients Generation* step is required prior to encryption in order to avoid the unintended leakage of (i) the ontology code distribution and (ii) the query result. In this section, we analyze the optimal dummy-generation strategy to achieve this goal.

We assume, without loss of generality, that each patient has a different set of observations; if there were equal patients in the database, fake ontology codes could be added to make them different. The leakage to HBC sites can be estimated by calculating (i) the adversary's equivocation (a.k.a. conditional entropy) on the ontology codes of the `observation_fact` table given their tagged versions, as an average measure, and (ii) the smallest anonymity set of the ontology codes, as a worst case measure. The higher the equivocation and the larger the anonymity set is, the lower the leakage is. For this exposition, we will focus only on the relation between patients and occurrences of ontology codes, leaving aside the temporal dimension, and we will follow the toy example shown in Figure 7.4. This figure represents the (horizontally) folded version of the (vertical) `observation_fact` table, therefore coding each patient as a row, each ontology code as a column, and each observed (resp. unobserved) code in a patient as a "1" (resp. "0") in the corresponding cell.



Ontology code		a	b	c	d	e
real patients	pid_1	1	1	1	1	0
	pid_2	0	1	1	1	1
	pid_3	1	0	1	1	1
dummy patients	pid_4	1	1	0	1	1
	pid_5	1	1	1	0	1

$\longleftarrow M \longrightarrow$
$\rightarrow \sigma_o(.) \rightarrow$
$\sigma_p(.)$
$\longleftarrow M' \longrightarrow$

Tagged	x	y	z	r	s	dummy flag
pa	1	1	1	0	1	$E(1)$
pb	1	1	1	1	0	$E(0)$
pc	1	0	1	1	1	$E(1)$
pd	0	1	1	1	1	$E(0)$
pe	1	1	0	1	1	$E(1)$

Figure 7.4: **Toy example.** Ontology code mapping to real and added dummy patients with pseudo-identifiers pid_i, and ontology codes a, b, c, d, e. pa, pb, pc, pd, pe are the randomly sorted version of the patient pseudo-identifiers, and x, y, z, r, s are the shuffled and deterministically encrypted (tagged) version of the ontology codes. The dummy flag is a probabilistic encryption of 1 for real patients and 0 for dummies.

More formally, let us define the matrix that associates ontology codes with patients as the tuple of a random binary matrix $\mathcal{M}$ where each row can be either a real or a dummy patient and each column represents one ontology code and two functions, σ_p and σ_o, that respectively map the patient pseudo-identifiers (pid_j in Fig. 7.4) to the rows (pa, pb, pc, pd, pe in Fig. 7.4) and the observed ontology codes (a, b, c, d, e in Fig. 7.4) to the columns (x, y, z, r, s in Fig. 7.4). These maps represent the shuffling applied to patients before they are assigned their pseudo-identifiers, and the shuffling and deterministic encrypted tag applied to ontology codes before they are loaded into the i2b2 database. In order to focus on the practical leakage of the deterministically encrypted database, let us assume that the tagging and the probabilistic encryption of the dummy flags do not leak anything about their inputs (their trapdoors cannot be broken), even if they are based on computational guarantees. Therefore, the adversary (each of the sites) observes the realization of the row- and column-permuted matrix: $\mathcal{A} \equiv [\mathcal{M}' = M']$, and her equivocation, with respect to the original information given $\mathcal{A}$, can be expressed as

$$H(\mathcal{M}, \sigma_o, \sigma_p|\mathcal{A}) = H(\mathcal{M}|\sigma_o, \sigma_p, \mathcal{A}) + H(\sigma_o|\sigma_p, \mathcal{A}) + H(\sigma_p|\mathcal{A}) \tag{7.1}$$

$$\overset{(a)}{=} H(\sigma_o|\sigma_p, \mathcal{A}) + H(\sigma_p|\mathcal{A}) \overset{(b)}{\leq} H(\sigma_o|\mathcal{A}) + H(\sigma_p) \overset{(c)}{\leq} H(\sigma_o) + H(\sigma_p). \tag{7.2}$$

Expression (7.1) can be divided in three terms: the first one represents the entropy of $\mathcal{M}$ conditioned to the two permutations and the observed contents of the cells, which is fully deterministic, hence zero-entropy (step (a) in (7.2)); the second term is the entropy of the ontology codes permutation conditioned to the observation of the matrix cells and the patient permutation, and the third term is the entropy of the patient permutation conditioned on the observed matrix contents. We aim at maximizing these two terms.

The last term of the equivocation can be maximized by making the dummy rows indistinguishable from the real patients; i.e., drawn from the same distribution. Empirically, this means that all the patients, real or dummy, have the same type of distribution, and the contents of the rows are independent of the position of the dummy patients in the list. This also makes the two permutations independent of each other even when conditioned on the contents of M' (step (b) in (7.2)). In our toy example in Fig. 7.4, all the real patients' rows belong to the same type (weight 4); by generating two new dummy patients with the same weight, they become indistinguishable from real patients in our simplified example.

In order to maximize the entropy of the ontology codes mapping σ_o conditioned on $\mathcal{A}$ (step (c) in (7.2)), all the permutations have to be equiprobable for the given M'. This is achieved by flattening the joint distribution of the observed ontology codes through the added dummies; the geometric interpretation of this flattening is that any column permutation can be cancelled out by a row permutation, such that it is not possible to univocally map any ontology code to any column in M'. In our toy example, it can be seen that due to the two added dummies, any fixed query yields the same number of patients independently of the permutation applied to the query terms, which gives a complete indistinguishability between all the tagged ontology codes even in light of the matrix M'. It must be noted that the unobserved codes do not have to be added to the table, as the adversary does not have a priori knowledge of which is the subset of observed codes, only its cardinality. Also, this strategy fully breaks the correlation between ontology codes; for example, if the site added only one dummy with codes a, b, e to the real patients in Fig. 7.4 the individual appearance rate of the codes would be flattened, but it would leak that there is a correlation between the codes c and d, that could be identified in the encrypted matrix through an l_p-optimization attack [149].

The last bound in (7.2) is the best that clinical sites can do with the dummy-patient addition strategy, knowing the matrix of real patients; it maximizes the uncertainty of the attacker about the original ontology concepts, for any real distribution of patients and ontology codes. The corresponding practical dummy-addition strategy can be described as follows: Real rows are grouped according to their weight (number of observations); if the whole set of observed ontology codes has n elements, for each group of rows of weight $k < n$, dummy rows are added to complete all the k-combinations of n elements, producing $\begin{pmatrix} n \\ k \end{pmatrix}$ rows (counting both real and dummies) per group. In our toy example, (considering independent codes) the equivocation goes from 3.58 bits with no dummies to 10.23 bits with the two dummies, while the minimum anonymity set raises from 2 to 5.

This strategy guarantees the maximum uncertainty for the adversary for an arbitrary real distribution of codes across patients, but it generates a combinatorial number of dummies, which is not feasible in general (unless the number of observed codes is very low); but if some assumptions can be made about the code joint distribution, we can simplify the strategy. If dependencies are only found within small groups of codes, being the groups mutually independent (that is the case for genomic information and dependencies found inside subsets of localized variants), it is possible to constrain the needed number of dummies by applying the same dummy-addition strategy in a restricted blockwise fashion. In order to flatten only the histogram of group weights, we group codes in independent blocks of size $n' \ll n$ and apply the dummy-generation permutation to the blocks (inter-block), but not to the contents of each block, until the block distribution is flat, therefore reducing the needed number of dummy rows. This trade-off strategy creates an "anonymity set" of ontology codes of size n/n' in such a way that the adversary cannot distinguish between the set of codes inside different blocks. The drawback is that the equivocation is reduced, as the resulting joint distribution of the ontology code observations is only flat across blocks, but not inside each block. In the worst case in terms of leakage (fully correlated codes within each block) the achievable adversary's equivocation becomes $H(\mathcal{M}, \sigma_o, \sigma_p|\mathcal{A}) = H(\sigma_o|\sigma_p, \mathcal{A}) + H(\sigma_p|\mathcal{A}) \leq H(\sigma_{o,n/n'}) + H(\sigma_p)$, where $\sigma_{o,n/n'}$ are the permutations of the n/n' blocks of n' codes each. This bound is achieved when the blocks are mutually independent, so the best partitioning strategy consists in keeping correlated codes inside the same block. If fully independence between codes can be assumed ($n' = 1$), it can be seen that flattening the observations histogram leads to the same maximum attacker equivocation as the complete permutation strategy (Eq. (7.2)), but with a much lower number of added dummies. In order to further reduce this number, it is possible to set a minimum anonymity set size m for the codes and add dummies to water fill the observation histogram (block-wise flat, instead of fully flat) until each code has at least other $m-1$ codes featuring the same number of observations.

Finally, it must be noted that whenever a site's i2b2 database is updated, dummies can be regenerated (and encryptions and tags re-randomized) when the ETL process (see Section 7.4.2) is run again for the whole updated database. The DDT protocol uses a different fresh randomness, so that the codes from the updated database cannot be linked back to the codes of the old one.

7.6 Privacy & Security Analysis and Extensions (MedCo+)

The main privacy and security requirements for MedCo are summarized in Section 7.2.3. In this section, we briefly discuss and analyze the fulfillment of these targets for MedCo, and we revisit possible extensions for more stringent requirements.

Security in MedCo is based on the cryptographic guarantees provided by the underlying decentralized data-sharing protocols (from UnLynx) and the adoption of well-established security practices when coding the interfaces with the i2b2 backend and the SHRINE interoperability layer. All input sensitive data are either deterministically (ontology codes) or probabilistically (dummy flags) encrypted with a collectively maintained key, such that they cannot be decrypted without the cooperation of all sites, thus guaranteeing confidentiality and avoiding single points of failure. For the full step-by-step security analysis of UnLynx, we refer the reader to [89]. Following this analysis, paired with the dummy strategy described in Section 7.5, it can be seen that MedCo covers the unlinkability requirement for the query results, thanks to the UnLynx DVS protocol; and it protects their confidentiality, as only the authorized investigator can decrypt the query results thanks to the UnLynx DKS protocol. Conversely, MedCo also enables the application of differentially private noise to the results to avoid membership inference attacks, and, thanks to the proposed dummy strategy, it guarantees confidentiality of the data also against all the clinical sites that participate in the system.

There are two extensions that can be applied to MedCo in order to satisfy additional confidentiality and integrity requirements: guaranteeing unlinkability among investigators' queries, and obtaining protection against malicious sites.

- Query confidentiality: In the basic MedCo system presented in Section 7.4, HBC sites can link the ontology codes used through different queries, as the applied deterministic tag is the same for all the queries. In the case that query confidentiality is also a requirement (e.g., investigators from pharmaceutical companies), it is possible to address it by probabilistically encrypting ontology codes during the ETL phase and by deterministically tagging the obtained ciphertexts with a fresh secret for each new query. The effective encryption key is different for each fresh run of the DDT protocol, so it is not possible to link the query terms between different runs of the shuffling-DDT. When this modified system (which we denote MedCo+) is paired with the proposed dummy-addition strategy, the terms between queries are indistinguishable and unlinkable, at the cost of transferring and tagging the subset of the encrypted database involved in the query.

- Malicious sites: MedCo's threat model assumes HBC sites to be credible and plausible assumption, based on the damage to reputation that a clinical site would suffer if it misbehaves in a collective data-sharing protocol. Nevertheless, it is possible to cope with malicious clinical sites by using UnLynx's proof generation protocols [89], which produce and publish zero-knowledge proofs for all the computations performed at the clinical sites, so that the proofs can be verified by any entity in order to assess that no site deviated from the correct behavior. UnLynx features zero-knowledge proofs for the DVS, DDT and DKS protocols, and the addition of differential privacy noise. Although this solution yields a hardened and resilient query protocol, the cost of producing all proofs causes a typically unacceptable burden in regular data sharing applications, for which the basic proposed MedCo covers all fundamental privacy requirements while yielding a very competitive performance, as shown in the next Section.

7.7 Deployment and Evaluation

We have deployed and tested MedCo in a real network of three sites in Switzerland: the École Polytechnique Fédérale de Lausanne (EPFL), the University of Lausanne (UNIL), and the Centre Hospitalier Universitaire Vaudois (CHUV). This section describes MedCo's performance for a clinical oncology use-case and shows its computational and storage overhead with respect to unprotected i2b2/SHRINE deployment.

7.7.1 Oncology Use-Case

Profiling somatic mutations is becoming increasingly common in oncology. Beside its role in diagnosis and prognosis, mutation profile is a system to guide treatment decision. It helps define the disease sub-types, quantify the level of intra-tumor heterogeneity, identify mutations driving tumor progression, and support clinical decision through targeted therapies. For a few genetic alterations that occur in a large number of patients, there are thousands of rare ones. In that context, being able to compare mutation profiles between patients across different clinics and identify those with a similar molecular profile is of critical importance for guiding treatment decision in oncology. For example, in case of a rare disease subtype or a bad prognosis after drug resistance, clinical decision recommendation can be inferred from patients sharing phenotypical and somatic mutations characteristics. Similarly, in clinical research, the ability to compare multiple patients with the same mutation profiles enables robust hypothesis generation and testing. The power of such an approach increases with the number of mutation profiles that can be queried. Therefore, being able to share somatic mutation information among hospitals and institutions is an absolute requirement. Yet, privacy and security concerns make such sharing extremely difficult, if not impossible. For these reasons, we tested MedCo on cancer genomic and clinical data from cBioPortal [57] by performing typical queries for oncogenomics, of which we report here two representative examples:

- Query A: *Number of patients with skin cutaneous melanoma AND a mutation in BRAF gene affecting the protein at position 600.* About half of melanoma patients harbor a mutation in the BRAF gene at position V600E or V600K and can be treated by the BRAF inhibitor *vemurafenib* [25]. The proportion of mutated BRAF melanoma is therefore an important benchmark for a clinic or hospital.

- Query B: *Number of patients skin cutaneous melanoma AND a mutation in BRAF gene AND a mutation in (PTEN OR CDKN2A OR MAP2K1 OR MAP2K2 genes).* This query is based on the fact that patients treated with *vemurafenib* develop resistance through mutations activating the *MAP kinase* pathways [217]. When facing drug resistance, finding another patient with a similar mutation profile could bring invaluable information for clinical decisions.

We used genomic and clinical datasets obtained from a skin cutaneous melanoma study [51, 106] of 121 patients with 9 clinical attributes and an average of 1,978 genetic mutations (239,286 observations in total).

7.7.2 Implementation

We developed MedCo as three components that fully integrate i2b2 [146], SHRINE [213] and UnLynx [89]: specifically (i) a new i2b2 server cell, developed in Java code, responsible for the MedCo business logic described in Section 7.4 and for interacting with UnLynx for the execution of secure distributed protocols, (ii) a new i2b2 Web-client plugin, developed in Javascript, enabling a user to generate queries involving somatic mutations through an annotation-based search engine, and to encrypt sensitive codes in the query directly in the browser, and (iii) an ETL system, developed in Go 1.8, responsible for extracting genomic and clinical data from a raw tab-separated file, encrypting them with the ElGamal collective key and loading them into the i2b2 data model.

Encrypted data were stored in the i2b2 data model with PostgreSQL [157]. In particular, somatic mutations were stored as deterministically encrypted observations in the `observation_fact` table encoded by the combination of their chromosomal position, reference allele and mutated allele. Genetic annotations (e.g., gene names) were stored in the `concept_dimension` table.

7.7.3 Experimental Setup

To avoid inconsistencies in the results, potentially caused by the heterogeneous setting of the EPFL-CHUV-UNIL network (i.e., different firewalls, servers, and access control mechanisms), and to obtain a fair comparison with the standard unprotected i2b2/SHRINE deployment, we ran our evaluation within an isolated environment. Such an environment comprises 3 servers interconnected by 10Gbps links and featuring two Intel Xeon E5-2680 v3 CPUs with a 2.5GHz frequency that support 24 threads on 12 cores, and 256GB RAM. We note, however, that bioinformaticians in the EPFL-CHUV-UNIL sites had similar user experiences. Each server hosted the SHRINE Web client, the i2b2 HIVe including the SHRINE adapter, query aggregator and the new MedCo cell, the i2b2 database, and the UnLynx server back-end. To set up our system and facilitate its deployment, we used Docker [143]. The default database contains the public dataset described in Section 7.7.1. We used UnLynx ElGamal encryption on the Ed25519 elliptic curve with 128 bit security.

To evaluate MedCo's performance, we considered four different experimental setups, with each measurement averaged over 10 independent runs:

1. Runtime for varying database size: For this setup, we ran query A (see Section 7.7.1) for different database sizes and measured the total runtime of MedCo, comparing it with: (i) The *insecure* i2b2/SHRINE implementation where all data are stored in clear, and (ii) the more *secure* version of MedCo, MedCo+, where we enhanced MedCo in order to protect also query confidentiality, as described in Section 7.6.

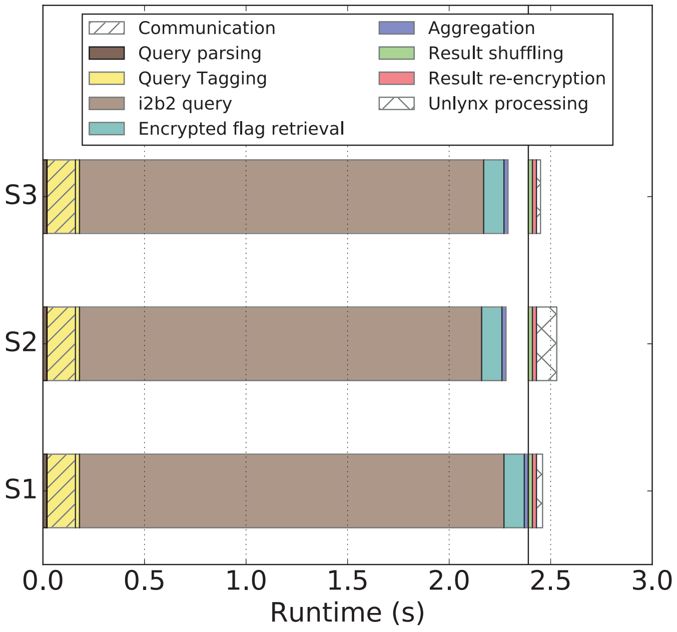
2. Runtime for varying number of sites: We studied MedCo's runtime for query A with varying number of sites in the network. We assumed that for each new site a new server is added to the system. We considered 3, 6, 9 and 10 sites with each site having a third of the original database (approximately 82.000 mutations or rows).

3. Network traffic for varying query size: In this setup, we assessed the amount of traffic incurred during a query in MedCo and MedCo+ by varying the number of queried ontology codes/mutations.

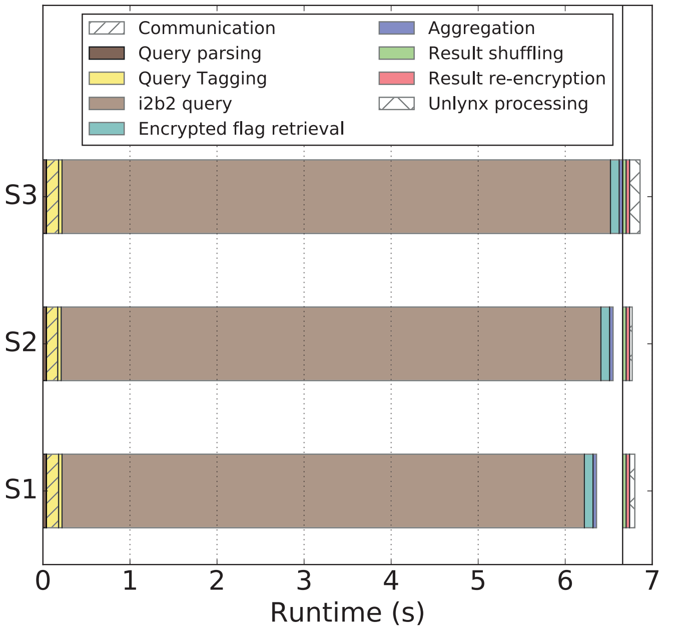
4. ETL runtime for varying database size: We studied the amount of time needed to extract, transform and load the data (pre-processing), which includes the formatting, initial deterministic encrypted tagging of codes, encryption of patients' flags and loading of the data in the i2b2 database.

7.7.4 Performance Results

Here we evaluate the raw overhead without dummies of the protocols featured in MedCo and MedCo+ with respect to the insecure i2b2; we analyze the impact of dummies in Section 7.7.5. Figure 7.5 provides query-workflow breakdowns for both query A and query B. Because they are negligible, we do not account for the query parsing and encryption/decryption times in the Web client, for the time to broadcast the query



(A) Query A.



(B) Query B.

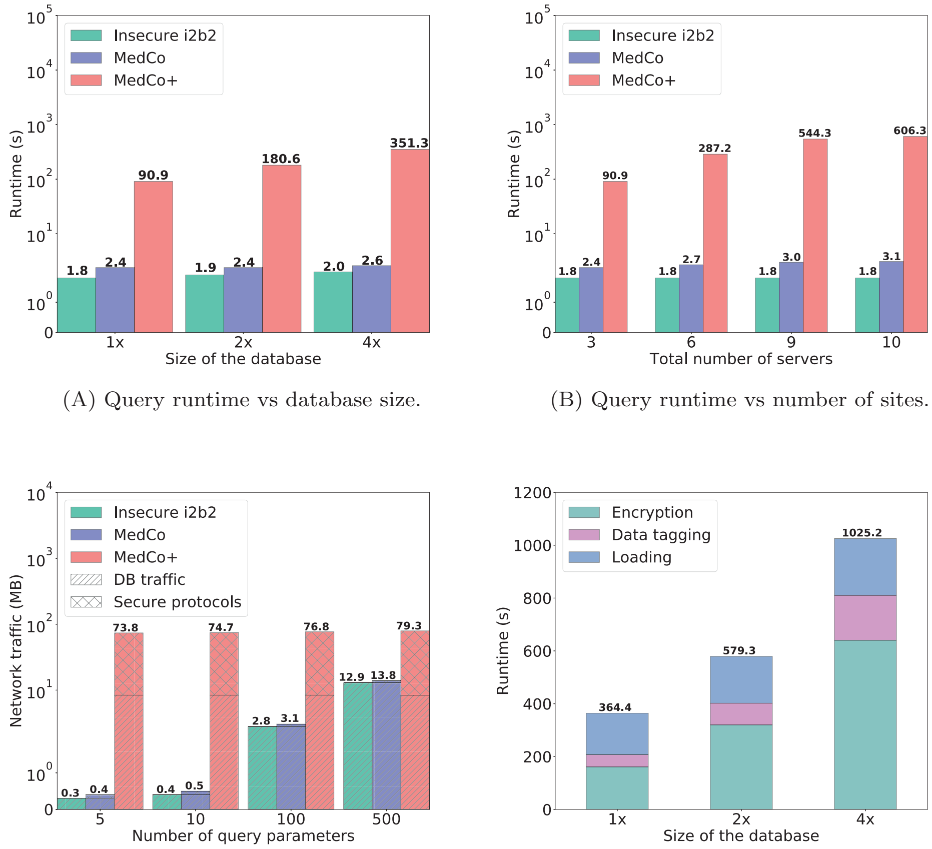
Figure 7.5: **Query-workflow breakdown.** Each site is represented as S1, S2, S3. The vertical black line signals the point where each node has to wait for the others before it can proceed.

from the SHRINE query aggregator to the different sites, and for the result obfuscation. Unexpectedly, results show that the i2b2 query to the `observation_fact` table is the most expensive operation in MedCo as it depends on the total number of observations. This time is also linear in the number of ontology codes in the query and it is inherent to the standard i2b2 database management for SQL-queries to the `observation_fact` table. Fetching the encrypted patients flags from the `patient_dimension` table, before homomorphic aggregation, can be also expensive as it depends on the number of patients satisfying the search criteria. The deterministic tagging of query encrypted codes is also linear in the number of ontology codes in the query, as each encrypted code has to be sequentially modified twice by each site in the network. Such a process takes more time for query B than for query A, as they consist of 79 (77 mutations and 2 clinical attributes) and 6 (4 mutations and 2 clinical attributes) query attributes, respectively. Differently, the homomorphic aggregation depends on the number of patients satisfying the query and it can be extremely fast for rare combinations of somatic mutations and clinical attributes. For queries A and B it takes around 0.68 and 0.40 milliseconds as only around 16 and 9 patients per site satisfy the research criteria. The remaining secure distributed operations introduced by MedCo depend on the number of sites in the network but they are negligible as they involve only one ciphertext, i.e., the encrypted query result.

Figure 7.6 shows the performance results for the four above-mentioned setups (1-4). The measurements are averaged out between servers. Subfigure 7.6A refers to the first setup and reports the time required to execute query A with different database sizes under different scenarios. Besides the normal database ('1x'), we chose two others with twice ('2x') and four times ('4x') more observations. These two additional databases were obtained by replicating the original one. In each case, the data were evenly distributed among the three sites, thus obtaining, for the three cases '1x', '2x' and '4x', around 80k, 160k and 320k observations over 40, 80 and 160 patients per site, respectively. Results show that MedCo is extremely efficient and comparable in terms of performance to the insecure version of the i2b2/SHRINE implementation. MedCo's overhead with respect to the insecure i2b2/SHRINE is almost constant when the database size increases as the privacy-preserving protocols introduced by MedCo depend mostly on the number of queried codes and the size of the resulting patient set. We can also observe that MedCo+ has a relatively higher runtime cost as a counterpart for achieving query unlinkability, because all the observations in the `observation_fact` table have to be deterministically tagged at runtime for each new query. However, we note that such a privacy enhancement might be necessary only under specific circumstances (e.g., when an investigator from a pharmaceutical company is using the system).

Subfigure 7.6B displays the results for the second setup, where we increase the number of sites in the network, to study the runtime scalability of MedCo. Results show that its overhead increases almost linearly with the number of participating sites. In other words, increasing the number of sites from 3 to 6 changes the MedCo's contribution, with respect to i2b2 from 0.4 to 0.7 seconds.

Subfigure 7.6C shows the network traffic incurred when increasing the number of queried ontology codes. The traffic was split between two main components: i2b2 database traffic, and traffic introduced by MedCo's secure protocols. As expected, network traffic increases linearly with the number of queried ontology codes for both MedCo and the insecure i2b2/SHRINE, whereas it is almost constant for MedCo+. Also, the database operations dominate the traffic in MedCo. Yet, the traffic caused by the



(A) Query runtime vs database size.

(B) Query runtime vs number of sites.

(C) Network traffic vs query size.

(D) ETL time vs database size.

Figure 7.6: **MedCo's performance results for setups 1-4.**

secure protocols is not negligible, as encrypted codes in the query are broadcasted to each site in the network. In MedCo+, the traffic of secure protocols becomes predominant with respect to the database traffic, as all observations in the database are broadcasted across the whole network in order to be deterministically tagged, regardless of the number of codes in the query.

Subfigure 7.6D shows the results for the pre-processing or ETL phase, including the encryption of the data (deterministic tagging for ontology codes and ElGamal encryption of patients' binary flags) and the data loading into the database. Results show that the ETL phase is costly and its time increases linearly with the amount of data in the database. However, it is important to mention that this phase is only executed once at each site, offline.

Finally, Figure 7.7 shows MedCo's ability to scale with respect to the insecure i2b2/SHRINE for database sizes of 100 and 1,000 times larger than the original database. Again, in both cases, patients were evenly distributed across sites thus obtaining for the two cases ('100x' and '1000x') around 8M and 80M observations over 4,000 and 40,000 patients, respectively. Results are impressive, as the total query runtime (for query A) for the '1000x' case, which can represent the standard size of a clinical site's database,

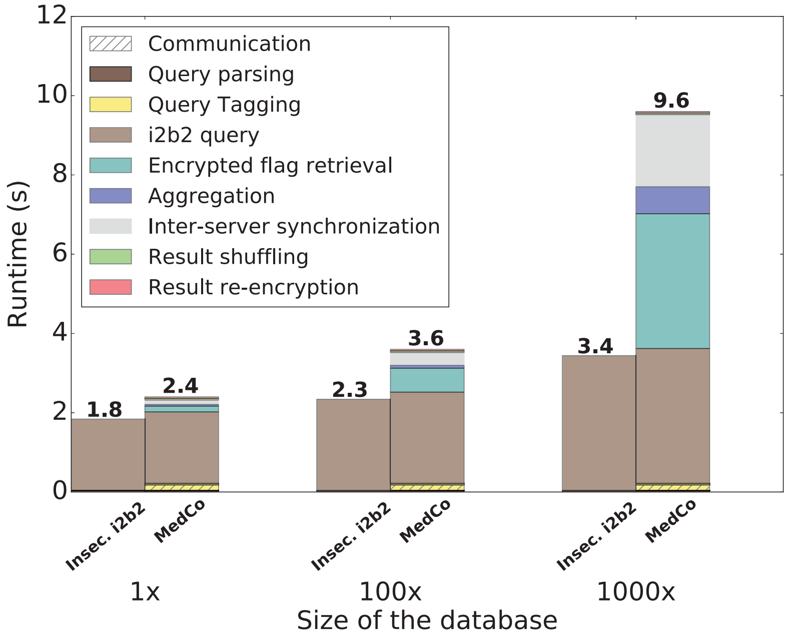


Figure 7.7: MedCo's scalability test: Query runtime vs database size (for a database up to 1,000 times larger the original database).

is still within 10 seconds. As expected, MedCo's overhead with respect to the insecure i2b2/SHRINE becomes more significant in the 1,000x case, as around 16,000 patients' encrypted flags need to be fetched from the `patient_dimension` table. This is due the artificial carbon-copy replication of the data and it is unlikely to happen for a typical oncology database with an equivalent size but without replicated patients. Similarly, also the time required for the homomorphic aggregation increases as the number of patients satisfying the query has increased by 1,000 times. However, we note that the most expensive operation still remains the i2b2 querying time that is independent of any additional privacy-preserving feature added by our solution.

7.7.5 Storage Overhead

In the unprotected i2b2 data model, ontology concepts are generally represented by 64-bit integers, whereas MedCo's deterministic encryption converts each code into a 32-bytes tag. Hence, the storage overhead introduced by MedCo's encryption is in the order of 4 times. Depending on the specific distribution of ontology codes across patients, a varying number of dummy patients must also be considered. In the tested oncology use-case, we assume independent codes and follow the dummy addition strategy described in Section 7.5. Because our original dataset is very sparse, with 238,363 ontology codes most of which (around 98%) are observed only once, the adversary's equivocation is already very high without the need of dummies (around 3.76 Mbits). We focus, therefore, on the minimum anonymity set for the observed codes. The extremely low number of common mutations and their small anonymity set (some of them having a unique rate) make common mutations easily identifiable by an adversary. Consequently, for maximizing both the equivocation and the minimum anonymity set (238,363 for all codes), the number of necessary dummy patients should be increased to 7,400, with 14 million new observations in total (approx. 62x increase factor in the number of rows of the `observation_fact` and the `patient_dimension` tables). Yet, by trading off the size of the smallest anonymity set, it is possible to reduce the number of needed dummy patients to 350 with roughly 614,000 observations, yielding a table increase factor of around 3.6x, which is practical. This guarantees a minimum anonymity set of 10,000 codes, and an adversary's equivocation close to the achievable upper bound. Due to space constraints, we do not show here the performance results of MedCo with dummies, but they can be assimilated to the '4x' case in Figure 7.6A.

7.8 Related Work

Among the operational systems for sharing of clinical or genomic information, SHRINE [213] and the GA4GH Beacon Network [87] are certainly the most advanced and widespread. However, as opposed to MedCo, they provide limited privacy guarantees (only ad-hoc result obfuscation) and no protection of data confidentiality besides standard access control, thus significantly restraining the amount of sharable information.

Besides that, and to the best of our knowledge, there are mainly two recent works dealing with privacy-preserving queries in distributed medical databases. The first one, PRINCESS [58], is based on trusted hardware: the sites encrypt all their data under AES-GCM (Advanced Encryption Standard - Galois Counter Mode) and send them to an enclave running in a central server, featuring an Intel SGX processor; this server decrypts and processes them, enabling the computation of statistical models. Compared to our work, PRINCESS can be more versatile in terms of allowed computations, but it presents a single point of failure (the central server), and centralizes all trust in the enclave and in the attestation protocol provided by Intel. Moreover, the memory restrictions of the enclave limit the scalability of the scheme, requiring compression and batching techniques to enable processing of large genomic data, for which MedCo scales much better.

The other recent approach, SMCQL [40], is based on secure two-party computation; it introduces a framework for private data network queries on a federated database of mutually distrustful parties. SMCQL features a secure query executor that implements different types of queries (e.g., merge, join, distinct) on the distributed database by relying on garbled circuits and Oblivious RAM (ORAM) techniques. Whereas this work features truly decentralized trust, it does not scale well to scenarios with more than two sites, which are likely to happen in medical contexts with a high number of collaborating hospitals.

7.9 Summary

In this chapter, we have presented MedCo, the first operational system that enables collective protection and privacy-preserving sharing of medical data across independent clinical sites. MedCo is based on widespread technologies from the biomedical informatics community, i2b2 and SHRINE, in order to be easily deployable on top of existing health information systems. Additionally, it relies on secure distributed protocols from UnLynx that enable different privacy/security vs. efficiency trade-offs, thus paving the way to the sharing of sensitive clinical and genomic information, which so far is not possible with existing operational systems. Finally, MedCo introduces a new general method for adding dummy patients (or records) in a database in order to conceal the distribution of deterministically encrypted ontology (or attributes) and to thwart frequency attacks. We have tested MedCo in a real operational environment by deploying it in a network of three institutions. Results on a clinical oncology use-case show small query-response times and good scalability with respect to the number of sites and amount of data. Therefore, we firmly believe that MedCo represents a concrete solution for enabling medical data sharing in a privacy-conscious and regulation-compliant way.

Chapter 8

GenoShare: Supporting Privacy-Informed Decisions for Sharing Exact Genomic Data

In the previous chapter, we have seen how cohorts of well-characterized patients can be identified across several clinical sites in a privacy-conscious way. The last step of the data-sharing process consists in sharing the individual records of these patients with investigators willing to perform more in-depth analyses. Yet, when individual genomic data need to be released, clinical sites are still hesitant as it is extremely difficult to fully understand the privacy risks that such a data-release entails. In this last chapter, we propose a new framework that supports privacy-informed decision for sharing exact genomic data by providing means to systematically reason about the risk of disclosing privacy-sensitive attributes (e.g., health status, kinship, physical traits).

8.1 Introduction

The privacy risks stemming from disclosing medical genomic data [79, 96, 148] are being increasingly amplified by the growing number of breaches occurring in healthcare organizations [197, 158, 105, 203]. This situation creates a complicated environment for health care practitioners and researchers trying to engage with citizens regarding the sharing of data for clinical research, as gaining their trust is becoming a major challenge. Currently, medical institutions and research centers address this problem by relying on a review board that decides whether disclosure is suitable. However, these decisions usually follow all-or-nothing policies, which provide little control on the inferences that can be made upon the shared data. Thus, they are of little help at conveying trust to users. The computer security community has made a remarkable effort to improve this situation, mainly focusing on solutions that perturb the data such that releases are differentially private [86, 116, 218], since anonymization approaches [189, 134, 129, 215] have been shown futile for privacy-preserving sharing of genomic data [99, 107].

Despite the demanding privacy needs of genomic data management, these solutions have not been adopted by practitioners so far. A main reason for this reluctance is that genomics applications usually require genomic data to be as exact as possible [79, 88, 151]. High accuracy is especially important in association studies aiming to identify significant correlations between particular genotypes and rare diseases, which are often weak signals. Moreover, differentially private solutions focus on safeguarding only the release of aggregates, and thus are not suitable for protecting individual's data, whose sharing is a common practice in research studies. In summary, *the need to release the **exact data values** precludes the use of state-of-the-art solutions that provide formal privacy guarantees in the presence of arbitrary external information.*

Yet, genomic data sharing is crucial to advance the state of the art in medicine. Thus, there is a high demand in the biomedical community for solutions that enable practitioners to reason about what exact data can be released while protecting individuals' privacy in clinical and research settings. Even though they cannot prevent inferences enabled by unforeseeable attack developments or data releases, such solutions would represent a great improvement over the current situation since they can effectively reduce the privacy risks based on the information available to the decision maker.

In this paper, we introduce GenoShare, whose goal is to assist practitioners in decision making by quantifying the risk of sensitive information leakage when sharing genomic data. Let us assume that an institution (e.g., hospital, research center) wishes to share genomic data, but is concerned about the privacy of the individuals who contributed their data. Upon reception of a request for genomic data sharing, such institution can use GenoShare to quantify the risk of sensitive attribute disclosure associated to revealing those data. To this end, GenoShare considers inference attacks relevant to the privacy concerns of the data contributors, and the information available to the adversary: i) the genomic statistics across populations [194], ii) the genomic association to sensitive information [104, 187], and iii) the correlations between genomic variants inside an individual's genome, and across related individuals' genomes. As opposed to prior works that consider only one type of inference attack at a time [107, 110, 175], GenoShare quantifies the risk of a privacy breach considering the joint effect of inference attacks, i.e., exploiting their interrelations, and can also consider partial adversarial knowledge – thus providing a more realistic risk estimation than the state-of-the-art approaches.

If the risk of sensitive attribute disclosure is deemed low, the institution can release the requested data in exact form, and otherwise it denies access to the data. GenoShare measures risk using novel intuitive metrics that, as opposed to current approaches based on inferences of raw genomic values [208], are directly related to the inference of tangible information, such as kinship or predisposition to a disease. Thus, they are well suited for modeling informed consent [123]. Furthermore, since denying access based on information secret to the adversary is known to leak information [121], GenoShare implements mechanisms to avoid this leakage.

To summarize, we make the following contributions:

- We present GenoShare, a framework that supports informed decision making regarding the sharing of *exact* genomic data by considering relevant inference attacks, and their joint effect on privacy.

- We introduce novel metrics that capture the risk of sensitive attributes disclosure, better suited to model informed consent than the state of the art. These metrics can be used to build privacy policies that model decision-making for genomic data sharing.

Notation	Description
aaf_i	Alternate allele frequency at position i
y	Disease susceptibility to be protected
$\Psi(y)$	Set of variants associated with disease y
$\omega^y = (\omega_1^y, \ldots, \omega_n^y)$	Set of effect-size coefficients for variants in $\Psi(y)$
$\phi_{A,B}$	Kinship coefficient for individual A and B
$\mathbf{g} = (g_1, \ldots, g_n)$	Set of genotypes for a real genome
$\mathbf{g_o}, \mathbf{g_u}$	Set of observed, unobserved genotypes used by genotype inference
R	Panel of reference individuals used by genotype inference
$\hat{\mathbf{g}} = (\hat{g}_1, \ldots, \hat{g}_n)$	Set of inferred genotypes
$\tilde{\mathbf{g}} = (\tilde{g}_1, \ldots, \tilde{g}_n)$	Set of genotypes for an avatar genome
$\dot{\mathbf{g}} = (\dot{g}_1, \ldots, \dot{g}_n)$	Set of most common genotypes in the population
$\mathbf{f} = (f_1, \ldots, f_n)$	Set of aggregated statistics for a real database
$\tilde{\mathbf{f}} = (\tilde{f}_1, \ldots, \tilde{f}_n)$	Set of aggregated statistics for an aggregated avatar
$q_g(\mathbf{g}_s)$	Query for the genotypes of a subset s of variants
$q_m(\mathbf{f}_s)$	Query for the aggregated statistics on a subset s of variants
$\mathbf{A_g}$	Genotypes revealed in previous queries
$\mathbf{A_m}$	Aggregated statistics revealed in previous queries
R^y	Risk of disclosing predisposition to disease y through phenotype inference attack
R^m	Risk of disclosing database membership through membership inference attack
R_d^k	Risk of disclosing familial relationship of degree d through kinship inference attack
α_m, β_m	Type I and II errors for database membership inference
α_d, β_d	Type I and II errors for kinship inference of degree d
$\boldsymbol{\rho}$	Set of thresholds on inference risks
ρ_y	Threshold on the risk of disclosing predisposition to disease y
ρ_m	Threshold on the risk of disclosing membership m to the database
ρ_k	Threshold on the risk of disclosing kinship
$\mathcal{B}$	Auxiliary information available to the adversary
$\mathsf{Priv}_{\tilde{\mathbf{g}}}$	Genome avatar's privacy
$\mathsf{Priv}_{\tilde{\mathbf{f}}}$	Aggregated avatar's privacy

Table 8.1: Notation used throughout the chapter. For all variables, bold indicates a set.

- We develop a novel method for preventing inferences based on genomic query denials. The idea is to internally use *avatars* (modified versions of individual's genomes) to decide upon data release, still releasing the original data when privacy is not at risk.

- We instantiate GenoShare with the three most relevant attacks on genomic privacy, advancing the state of the art by adapting them to consider partial information and considering their interrelations to amplify their inference power. We show GenoShare's effectiveness at detecting potential private information leaks using real data from the 1,000 Genomes Project [194].

8.2 The Genomic Sharing Scenario

We consider a scenario in which an institution holds a database $\mathbf{D}$ with genomic data of lots of individuals. We model an individual's genome in $\mathbf{D}$ as a vector $\mathbf{g} = (g_1, \ldots, g_n)$ formed by n variants on autosomal chromosomes (i.e., not sex chromosomes), where g_i denotes the value of variant i. We encode the value (or genotype) of a variant g_i at position i as $g_i \in \{0, 1, 2\}$, based on the number of alternate alleles it contains. We use a vector $\mathbf{f} = (f_1, \ldots, f_n)$ to model aggregated statistics on these variants. We summarize the notation in Table 8.1.

Institutions wish to share these data in one of two ways: i) as a subset of genotypes $\mathbf{g}_s$, in response to a *genotype request* for variants of a given individual, $q_g(\mathbf{g}_s)$; and ii) as a subset of aggregated statistics $\mathbf{f}_s$ for a specific group of individuals, in response to an *aggregated request* for variants, $q_m(\mathbf{f}_s)$. Because genomic-related applications are not tolerant to noisy data, institutions want to share them in their original, exact, form.

On the other hand, individuals whose genomes are in $\mathbf{D}$ could be concerned about the potential disclosure of their sensitive information. They express such concerns establishing a threshold, $\boldsymbol{\rho}$, that captures their tolerance to disclosure of sensitive information with respect to the risk of inference.

We assume an adversary who wants to learn some (sensitive) information about individuals in the database protected by GenoShare. Genome-based inference attacks can be categorized as follows: (i) *Phenotype inference attacks*, that aim at inferring an individual's predisposition to diseases (e.g., Alzheimer's disease, cancer, schizophrenia) [79], or her physical traits, from her genotype and known genotype-phenotype correlations [104]; (ii) *Membership inference attacks*, whose goal is to infer the presence of an individual of whom genomic information is available in a group for which aggregate statistics are known [107, 181], which can be very sensitive if such a group is associated with a sensitive attribute (e.g., HIV-positive patients, patients in a psychiatric institute, etc.); (iii) *Kinship inference attacks*, that aim at inferring familial relationships between know individuals' genomes; (iv) *Re-identification attacks*, that aim at inferring the identity (e.g., family name) [99] behind a known genome, or physical traits (e.g., height, eye color, etc.) that can lead to re-identification [61]; or (v) *Linking attacks*, that aim at linking anonymized sensitive phenotype data available to the adversary to a set of individuals for which their genotypes are known by exploiting genotype-phenotype correlations [112, 102].

To perform inferences, the adversary could have access to the following information :
–*Background information ($\mathcal{B}$)* such as public information about average individuals' genomes [194], and about genomic association to sensitive information [187, 104]; or information made public by individuals, e.g., on OpenSNP [152] that provides access to further genomic data, or on Facebook [81] that provides information about familial relationships [110],
–*Revealed variants* of the targeted individual and of her relatives ($\mathbf{A_g}$), and aggregated statistics ($\mathbf{A_m}$),
–*Potentially revealed variants*: information that would be revealed if a new request is granted ($\mathbf{g}_s$ or $\mathbf{f}_s$), or that could be inferred in case of GenoShare denying a high-risk query.

8.3 GenoShare

We design GenoShare to help institutions owning a database of genomic sequences to share exact genotypes ($\mathbf{g}_s$) or aggregated statistics ($\mathbf{f}_s$) in a privacy-conscious way. When GenoShare receives a data request (either $q_g(\mathbf{g}_s)$ or $q_m(\mathbf{f}_s)$), it quantifies the privacy risks regarding inferences stemming from the release of the data. If the risks are deemed too high with respect to given thresholds $\boldsymbol{\rho}$, GenoShare prevents any automatic release of data. It can further provide the institution with information that can be used to make a privacy-conscious decision regarding whether to share the requested data.

We note that when queries are granted, GenoShare *cannot protect the information that has already been released* from inferences that could be made possible by advances in the state of the art in genomics research. This limitation is inherent to the practitioners' need for clean and exact data.

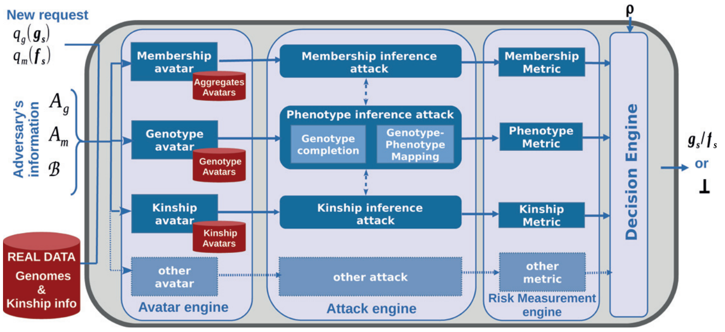


Figure 8.1: The four main blocks in GenoShare: i) an *Avatar Engine* that generates and stores avatar genomes which are used *internally* to avoid inferences based on query denials; ii) an *Attack Engine* that simulates the adversary's behavior in order to predict what information could be learned if the requested data are released; iii) a *Risk Measurement Engine* that quantifies the risk of disclosing sensitive information when releasing data; and iv) a *Decision Engine* that verifies whether the privacy risks are under the thresholds $\boldsymbol{\rho}$, and either outputs the **exact requested data** ($\mathbf{g}_s$ or $\mathbf{f}_s$) or nothing. The figure shows instantiations of the blocks for three inference attacks, and the bottom row illustrates how other attacks could be accommodated.

8.3.1 Architecture

GenoShare is conceptually divided in four main blocks, illustrated in Figure 8.1:

Attack Engine. This engine simulates the inference attacks that the adversary can perform given both the information already available to him ($\mathbf{A_g}$, $\mathbf{A_m}$, and $\mathcal{B}$), and what would be disclosed if the query was granted ($\mathbf{g}_s$ or $\mathbf{f}_s$).

Risk Measurement Engine. This engine computes the risk of sensitive attribute disclosure materializing if the data requested in the query is revealed.

Decision Engine. This engine checks if the risk computed by the Risk Measurement Engine exceeds the established thresholds for any individual in the database.

Avatar Engine. The Avatar Engine creates and stores avatars: modified versions of the stored genomic data. Avatars are used, *internally*, in the Attack Engine to simulate the inference attacks, and in the Risk Measurement Engine to quantify the inference risk. Their goal is to mitigate potential inferences on the true genome based on query denials. Note that, whenever the decision is to grant the queried data, the *true* data is released.

8.3.2 Using GenoShare

Initialization. GenoShare requires an initialization phase in which the attacks to be considered are instantiated in the Attack Engine (Sect. 8.4); the privacy metrics are set up in the Risk Measurement Engine (Sect. 8.5); the corresponding risk thresholds $\boldsymbol{\rho}$ are set up in the Decision Engine; and the Avatar Engine generates one avatar per individual per attack considered in the Attack Engine (Sect. 8.6).

Operation. Upon reception of a data request ($q_g(\mathbf{g}_s)$, or $q_m(\mathbf{f}_s)$), two steps are needed to run GenoShare:

1. Configuration. GenoShare needs to be configured to decide: (i) what background information $\mathcal{B}$ is assumed to be available to the adversary (e.g., only data released by tool, familial relationships, genome data obtained from other sources,...); and (ii) which attacks are of a concern for the given request.

2. Execution. To evaluate the query, GenoShare substitutes the requested data ($\mathbf{g}_s$ or $\mathbf{f}_s$) by the corresponding avatar genotypes ($\tilde{\mathbf{g}}_s$ or $\tilde{\mathbf{f}}_s$). Then, it runs *all* the attacks configured in the Attack Engine, whose output is then input to the Risk Measurement Engine. This engine computes the risk of a privacy breach, and the Decision Engine verifies whether it complies with all individuals' thresholds. If *any* of the risks exceeds the corresponding thresholds, GenoShare prevents the release of the data. If not, it releases the exact data requested in the query ($\mathbf{g}_s$ or $\mathbf{f}_s$).

8.4 GenoShare's Attack Engine

The Attack Engine runs all the inference attacks configured in GenoShare , to mimic the actions that an adversary would perform to learn sensitive information about individuals. As opposed to prior works that consider inference attacks independently, GenoShare considers them jointly, thus maximizing their effect. We now introduce the three most well-known attacks, namely *phenotype*, *membership*, and *kinship inference*, that we instantiate with state-of-the-art inference techniques. We note that GenoShare can accommodate other attacks, such as the *re-identification attack*, the *linking attack* or others, and it can be updated with better techniques each time there is a new proposal.

8.4.1 Phenotype inference attack

Phenotype inference attacks are aimed at learning genotype-related sensitive phenotypes about a target individual (and her relatives), such as predisposition to diseases or physical traits. For simplicity, in this paper we only consider predisposition to diseases inference.

Phenotype inference attacks run in two steps: (i) *genotype completion*, in which the adversary uses the target's known variants, i.e., those that he has already observed, to infer correlated unobserved variants both from the target and her relatives, and (ii) *genotype-phenotype mapping*, in which, given the recovered variants, the adversary computes the target's disease predisposition using publicly available information about genotype-phenotype correlations. We now provide details about these two phases that are relevant to understand the avatar generation algorithms in Sect. 8.6.

Genotype completion. Genotype completion enables the inference of unobserved variants $\mathbf{g}_u$ from the variants available to the adversary (i.e., previously revealed and to be revealed, $\mathbf{g}_o := \mathbf{A}_\mathbf{g} \cup \mathbf{g}_s$, and background information $\mathcal{B}$), and the genotype data in a public panel of reference individuals $\mathbf{R}$. It outputs a posterior probability distribution for each unobserved variant, $\hat{\mathbf{g}} = \Pr(\mathbf{g}_u | \mathbf{g}_o, \mathcal{B}, \mathbf{R})$. We instantiate this inference using a well-established statistical technique called *genotype imputation* [138] that makes use of a Hidden Markov Model (HMM) to model the target's genome. Simpler techniques could be used but, to the best of our knowledge, genotype imputation provides the most accurate genotype inference [175].

At a high level, genotype imputation works by using patterns of blocks of highly correlated variants, so-called *haplotypes*, in the reference panel to predict unobserved variants when only a subset $\mathbf{g}_o$ of $\mathbf{g}$ has been observed. By definition, a haplotype is a set of variants on a chromosome that tend to always occur together, i.e., that are statistically correlated. This process is illustrated in Figure 8.2.

Then, the adversary can infer the variants of the target's relatives from the completed genotype, i.e., the exact observed genotypes $\mathbf{g}_o$ and the probabilistically inferred

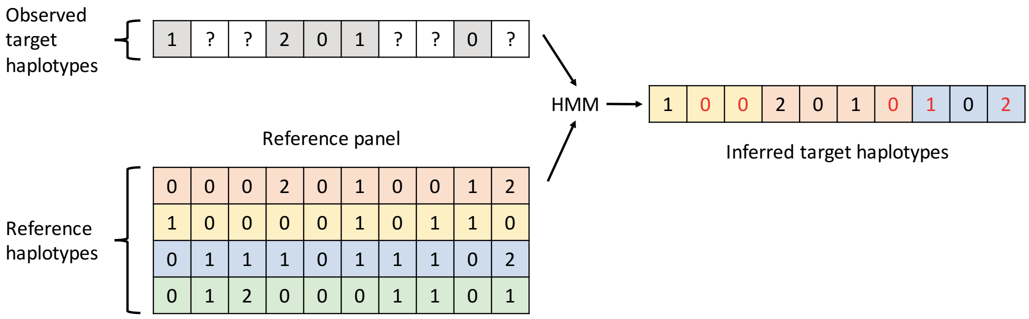


Figure 8.2: High-level representation of the genotype imputation technique.

ones $\hat{\mathbf{g}}$, and the knowledge on target's family tree. We model the target's family as a Bayesian network and the well-established *junction tree* algorithm [127] can be used to compute, for each family member, the set of marginal probability distributions over the unobserved variants conditioned on the target's information previously inferred ($\mathbf{g}_o \cup \hat{\mathbf{g}}$) (see Figure 8.3 for a high-level representation of the junction tree algorithm).

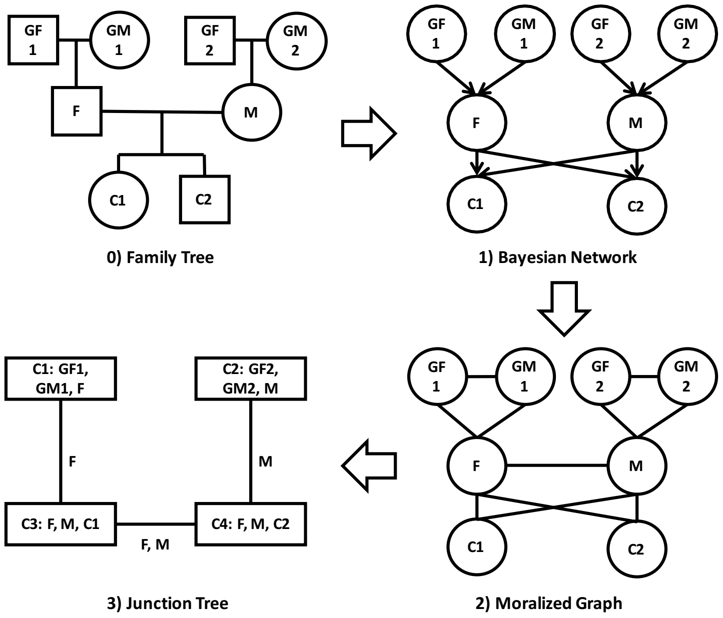


Figure 8.3: High-level representation of the junction tree algorithm: From a family tree (0), a Bayesian network is constructed (1) and then moralized through triangulation in order to obtain a moralized graph (2). Finally, The moralized graph is transformed into a clique tree (or junction tree) (3) by guaranteeing that for each pair of cliques U, V with intersection S, all cliques on path between U and V contain S.

Genotype-phenotype mapping. An individual's genetic predisposition to a disease can be inferred from her genotype at variants associated with the disease, and the strength of this genotype-phenotype association. As explained in Section 2.1, this strength is characterized by the effect size $\omega^y = \log(OR)$, where OR is the odds ratio. More formally, let $\Psi(y)$ be the set of variants associated with disease y. Then, the adversary computes the target's predisposition to disease y, denoted as P^y, as the linear combination of the target's inferred genotypes $\hat{g}_i$ of variants i in $\Psi(y)$ weighted by the strength of the genotype-phenotype association ω_i

$$P^y = P_0^y + \sum_{i \in \Psi(y)} \omega_i \sum_{k \in 1,2,3} k \Pr(\hat{g}_i = k)\,, \tag{8.1}$$

where P_0^y is the baseline predisposition to y. Note that for monogenetic diseases (e.g., breast cancer), $\Psi(y)$ consists of only one variant.

8.4.2 Membership inference attack

The membership inference attack enables the adversary to infer whether a target individual, for which variants are known, is present in a group of individuals for which genetic aggregated statistics are available [107, 210, 176, 207, 113, 220, 181]. We instantiate this attack using the technique proposed by Homer et al. [107] because it relies on less restrictive assumptions than other approaches, but any other membership inference technique could be used instead. We note that this technique is different from the one by Shringarpure and Bustamante described in Chapter 6 as it uses allele frequencies instead of allele presence/absence information. Homer's technique compares, for every target's observed variants $\mathbf{g}_o$, (i) the distance between the alternate allele frequency $\frac{g_i}{2}$ in the target's genotype and aaf_i (the alternate allele frequency in the population) with (ii) the distance between $\frac{g_i}{2}$ and f_i, the frequency of the same allele in the group of interest. Formally, using the L_1 distance:

$$D\left(\frac{g_i}{2}\right) = \left\|\mathsf{aaf}_i - \frac{g_i}{2}\right\| - \left\|f_i - \frac{g_i}{2}\right\|. \tag{8.2}$$

When the target is in the group, $E\left[D(\frac{g_i}{2})\right]$ is greater than zero because $\frac{g_i}{2}$ shifts f_i away from aaf_i. On the contrary, under the null hypothesis (the target is not present in the group of interest) $E\left[D(\frac{g_i}{2})\right]$ should approach zero. If $\frac{g_i}{2}$ is further away from the group than from the reference population, i.e., even less likely to be part of the group, $E\left[D(\frac{g_i}{2})\right]$ is negative.

If the number of released frequencies is sufficiently high, $E\left[D(\frac{g_i}{2})\right]$ converges to the normal distribution due to the central limit theorem. This enables the adversary to make use of a one-sample t-test to determine whether the target is part of the group or not. As we explain in Section 8.4.4, the adversary can make use of the output of the *genotype completion* ($\hat{\mathbf{g}}$) to further improve the membership attack. As the output of genotype completion is a probability distribution over the possible values of a variant, we adapt (8.2) such that it incorporates this knowledge. For $k \in \{0, 1, 2\}$:

$$D\left(\frac{\hat{g}_i}{2}\right) = \left\|\mathsf{aaf}_i - \sum_k \frac{k}{2} \Pr(\hat{g}_i = k)\right\| - \left\|f_i - \sum_k \frac{k}{2} \Pr(\hat{g}_i = k)\right\|. \tag{8.3}$$

8.4.3 Kinship inference attack

The kinship inference attack enables the adversary to infer the degree of kinship of a pair of target individuals, given a common set of their variants. We instantiate this attack, for the first time, with a technique that estimates the identical-by-descent (IBD) statistics representing the proportion of genomic variants co-inherited from a common ancestor [137, 199]. Similarly to the previous attacks, other techniques could be used. In particular, we use the kinship coefficient $\phi_{A,B}$ proposed in [137], defined as the probability that two alleles sampled at random from two individuals A and B are identical by descent. We compute $\phi_{A,B}$ as the average over all variants' coefficients $\hat{\phi}_{i,A,B}$ computed as:

$$\hat{\phi}_{i,A,B} = \frac{2\mathsf{aaf}_i(1-\mathsf{aaf}_i) - (g_{i,A} - g_{i,B})^2}{8\sum_{i \in M_{A,B}} \mathsf{aaf}_i(1-\mathsf{aaf}_i)}, \tag{8.4}$$

where $M_{A,B}$ is the set of observed variants for both individuals. If the number of observed variants is sufficiently high, the sum of $\hat{\phi}_{i,A,B}$ over all variants converges to the normal distribution due to the central limit theorem. Similarly to the previous attack, the adversary can infer the degree of kinship of the target individuals by using the inference criteria in [137] (see Table 8.2). As there are different kinship levels, in our experiments we use a closed testing procedure [139] of consecutive one-sample t-tests and choose the closest relationship that can be inferred.

Relationship	$\phi_{A,B}$	Inference criteria
Monozygotic twin	$\frac{1}{2}$	$> \frac{1}{2^{3/2}}$
Parent-offspring	$\frac{1}{4}$	$\left[\frac{1}{2^{5/2}}, \frac{1}{2^{3/2}}\right]$
Full siblings	$\frac{1}{4}$	$\left[\frac{1}{2^{5/2}}, \frac{1}{2^{3/2}}\right]$
2nd degree	$\frac{1}{8}$	$\left[\frac{1}{2^{7/2}}, \frac{1}{2^{5/2}}\right]$
3rd degree	$\frac{1}{16}$	$\left[\frac{1}{2^{9/2}}, \frac{1}{2^{7/2}}\right]$
Unrelated	0	$< \frac{1}{2^{9/2}}$

Table 8.2: Kinship inference criteria based on the estimated kinship coefficient $\phi_{A,B}$.

Note that the estimation of the kinship coefficient relies on the assumption that variants are independent hence contrary to membership inference, kinship inference cannot benefit from genomic imputation that relies on correlation.

8.4.4 Attacks interrelations

In order to best estimate the adversary's inference capabilities, the Attack Engine takes into account interrelations between the different attacks which can benefit from each other as shown in Figure 8.4. The genotype completion carried out within the phenotype inference attack can be used to improve the efficacy of the membership inference attack, because it increases the knowledge of the adversary about the target's genotype information: A larger number of the target's variants is made available to establish her membership to the database. On the contrary, the kinship inference attack cannot benefit from genotype completion. This is because genotype completion relies on correlations between variants, but the accuracy of the estimated kinship coefficient relies on independent variants. However, the kinship inference attack can improve the phenotype inference attack by informing about the familial ties which enable us to build the Bayesian network model. As a consequence, the kinship inference attack indirectly and positively influences the membership inference attack. Finally, the membership inference attack can also enhance the genotype completion of the phenotype inference attack if it reveals that the target is present in a database associated with a phenotype that is correlated to a particular genotype. We do not evaluate the latter in Section 8.7, as understanding the information gained by the adversary is straightforward.

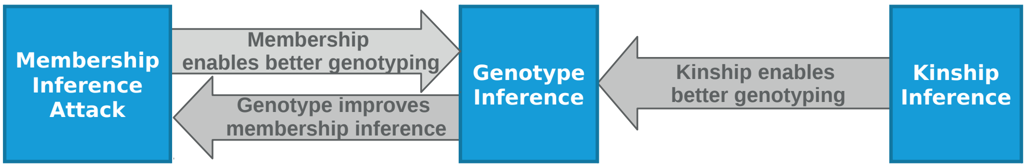

Membership Inference Attack
Membership enables better genotyping
Genotype improves membership inference
Genotype Inference
Kinship enables better genotyping
Kinship Inference

Figure 8.4: Interrelation among inference techniques

8.5 GenoShare's Risk Measurement Engine

GenoShare needs to have means to measure the risk of sensitive attribute disclosure when granting a request $q_g(\mathbf{g}_s)$ or $q_m(\mathbf{f}_s)$. Such risk needs to be understood by a large variety of users with extremely diverse knowledge related to genomics and/or medicine (e.g., patients, doctors, researchers). As such, we propose metrics to represent sensitive attributes that could be understood by the public at large [123]. It must be noted, however, that the Risk Measurement Engine could be instantiated with any other metrics deemed suitable for measuring the information leaked to the adversary in order to characterize other more-or-less-specialized concerns. For instance, metrics oriented to avoid bulk disclosure of data, e.g., by including a large percentage of variants in the risk computation, or metrics of interest for experts, such as genomic researchers, in which only specific variants are considered to be risky.

8.5.1 Phenotype inference risk

The phenotype inference risk aims at capturing how well the adversary can infer a target's phenotype, e.g., a physical trait or a predisposition P^y to a disease, regardless of the actual phenotype value, i.e., her inference error. For simplicity we focus our explanation on disease predisposition, but note that the metric can be easily adapted to other phenotype inferences.

Disease predisposition can be inferred through the *phenotype inference attack*, see Section 8.4.1. If the adversary had access to all the target's actual genotypes, he could perfectly compute her predisposition. However, when GenoShare is in place, the adversary only has access to the previously disclosed genotypes ($\mathbf{A_g}$), and to those that he can infer using the genotype completion ($\hat{\mathbf{g}}$). Recall from Section 8.4.1 that, for each $\hat{g}_i$, the adversary obtains a probability distribution over the three possible variant values $\{0, 1, 2\}$. Given the known and inferred variants, the predisposition P^y can be estimated using the genotype-phenotype mapping.

Wagner defined the per-variant success rate of the adversary as the probability that the adversary correctly infers the true variant value given the genotype inference output [208]. Inspired by this metric, we quantify the risk of inferring a given disease predisposition, denoted as R^y, by weighing the per-variant privacy metric by the per-variant strength of association ω_i between genotype and disease predisposition:

$$R^y = \frac{1}{\sum_i \omega_i} \sum_i \omega_i \Pr(\hat{g}_i = g_i), \quad i \in \Psi(y), \tag{8.5}$$

where $\Psi(y)$ is the set of variants associated with disease y and g_i the individual's true genotype of variants i in $\Psi(y)$. The metric R^y is an absolute measure of the adversary's success given the known and inferred genotypes. We note that, even if no information has been released, the adversary can use her prior information (i.e., the variants' alternate allele frequency in the population aaf). Thus, the mimimum risk faced by the individual is

$$R^{y,min} = \frac{1}{\sum_i \omega_i} \sum_i \omega_i \Pr(\hat{g}_{i,\mathsf{aaf}} = g_i), \quad i \in \Psi(y) \tag{8.6}$$

where $\hat{g}_{i,\mathsf{aaf}}$ denotes the adversary's estimation of g_i given his prior knowledge.

8.5.2 Membership and kinship inference risks

We consider membership and kinship inferences to be binary classification problems. These are based on a one-sample t-test used to test the null hypothesis of the individual not being in the dataset, in the case of membership (see Sect. 8.4.2), or not being related to anyone else in the database in the case of kinship (see Sect. 8.4.3). Thus, to quantify these inference risks, we use α, the significance level of the classification (i.e., the false positive rate), and $1 - \beta$, the test statistical power, where β denotes the false negative rate. Intuitively, the higher the power and the lower the significance level are, the more certain the adversary is about his classification. Therefore, an individual's privacy grows with α and β (i.e., when the number of false positives, resp. false negatives, grows).

We define the risk of membership inference as

$$R^m = (1 - \beta_m, \alpha_m), \tag{8.7}$$

and the risk of inferring kinship of degree d as

$$R^k_d = (1 - \beta_d, \alpha_d), \quad d \in \mathbb{N}^* . \tag{8.8}$$

where the subscripts of α and β refer to the membership (m), respectively the d-degree kinship (d), inference attacks.

8.6 GenoShare's Avatar Engine

Denying access to private data can leak information about these data, because decisions are based on information not available to the attacker [121]. Simulatable auditing [122] prevents such leakage by anticipating incoming queries, and only replying to those that do not enable unauthorized inferences. Unfortunately, existing solutions in this direction are limited to statistical queries different from aggregated requests in the genomic scenario, and not applicable to the case of individual genotype requests.

The key intuition in simulatable auditing is that, to prevent leakage, a decision to deny a query must be based exclusively on the information released by the system (including the potential answer to the current query), but *not* on the query itself nor other value in the database. Building on this idea, one may be tempted to use existing techniques to produce synthetic data [46], or perturb the data to make it differentially private [34], in order to obtain alternative data with similar statistical properties to those of the original data. These alternative data can be used as input for the decision process so that the query denial does not depend on the original sensitive data. However, by following this approach, GenoShare can take incautious decisions in particular instances, i.e., granting a query deemed safe for the alternative input, while for the original genomic data it would have raised an alarm. Such a risky behavior is not acceptable for medical institutions notably because of patients' privacy.

To mitigate this problem, we propose to use *avatars*, new modified versions of an individual's genome or a database's aggregates used *internally* by GenoShare. Avatars, as opposed to perturbed or synthetic data, always guarantee conservative decisions when used in GenoShare's decision process. They are used as input to the Attack and Risk Measurement Engines instead of the original genomes, thus ensuring that, given a denial, the adversary can, at most, recover the avatar. We note that when a query is granted, i.e., deemed safe, always the original data is released, not the avatars.

We define two types of avatars: genome avatars ($\tilde{\mathbf{g}}$), and aggregates avatars ($\tilde{\mathbf{f}}$) to substitute genotypes ($\mathbf{g}_s$) and aggregates ($\mathbf{f}_s$) real inputs, and construct them to ensure that decisions are never incautious. In terms of the risk metrics defined in Section 8.5, this implies that *phenotype inference attacks* on the avatar should result in a success rate, R^y, larger than on the real genome; and *membership* (resp. *kinship*) *inference attacks* should result in higher power, $1 - \beta_m$ (resp. $1 - \beta_d$) for a given significance level α_m (resp. α_d).

Avatar-based privacy. Guaranteeing safety of the decisions with respect to the real genomes inevitably leads us to generating avatars that are not fully independent from the real genomic data they represent. Thus, we cannot provide the provable protection guaranteed by simulatable auditing. Instead, we quantify the level of privacy provided by GenoShare's avatars. Let us consider genotype requests as an example. Given a query denial, the probability of the adversary inferring one true genotype is:

$$\Pr[g_i|\tilde{g}_i] \cdot \Pr[\tilde{g}_i|\text{denial}]\,, \tag{8.9}$$

where $\tilde{g}_i$ denotes the value of g_i's avatar, $\Pr[g_i|\tilde{g}_i]$ denotes the probability that the adversary succeeds at recovering true genotypes from the avatar, and $\Pr[\tilde{g}_i|\text{denial}]$ denotes the probability of learning the avatar from the denial. The latter strongly depends on the concrete sequence of queries and their replies, hence cannot be computed analytically. Thus, we choose to assume the worst-case scenario in which the adversary does recover the avatar and concentrates on computing the first probability, $\Pr[g_i|\tilde{g}_i]$. This worst-case scenario provides a lower bound on the privacy provided by avatars. If the adversary cannot correctly recover the avatar from the denial (i.e., $\Pr[\tilde{g}_i|\text{denial}] < 1$), the overall privacy increases.

Then, for a given individual, we compute her avatar's privacy as the average error of the adversary over all variants:

$$\mathsf{Priv}_{\tilde{\mathbf{g}}} = \frac{1}{n}\sum_i \left(1 - \Pr[g_i|\tilde{g}_i]\right), \;\; \mathsf{Priv}_{\tilde{\mathbf{f}}} = \frac{1}{n}\sum_i \left(1 - \Pr[f_i|\tilde{f}_i]\right). \tag{8.10}$$

We note that, depending on the use case, it could make sense to only consider variants that are deemed most sensitive for the individual.

In the following, we propose avatar-generation algorithms for the three families of techniques we instantiate in the Attack Engine. We note that the proposed avatar generation methods are not tied to any particular implementation of the attacks, but based on their fundamental operation principle. Thus, they are valid for any attack inside a family.

8.6.1 Genome avatar

Genome avatars $\tilde{\mathbf{g}}$ are used as input to the Attack Engine when GenoShare receives a genomic request $q_g(\mathbf{g}_s)$. Since the inference techniques are based on different principles, avatars must be technique-dependent to guarantee that, for all cases, GenoShare outputs a conservative decision.

Phenotype inference. GenoShare quantifies the phenotype inference risk stemming from a phenotype inference attack using R^y (as in (8.5)), dependent on the adversary's error. Hence, to trigger conservative decisions avatars need to reduce this error with respect to the case where the real genome would be used for the attacks. Phenotype inference attacks rely on genome completion to infer unknown variants before using a phenotype-genotype mapping to perform the inference (see Sect. 8.4.1). The working principle of genotype completion techniques is to infer unobserved variants using common patterns in a reference panel $\mathbf{R}$. This implies that inferred variants are likely to be equal to the most common variants in $\mathbf{R}$. Thus, setting the avatar to such common variants increases the probability that the inferred variants are equal to the avatar ($\Pr(\hat{g}_i = \tilde{g}_i)$), reducing the error in R^y.

Let us denote as $\dot{g}_i$ the most common value in the reference panel for variant i. Depending on the variant's aaf, we have that $\dot{g}_i = 0$, if $\mathsf{aaf} \leq 0.5$, and $\dot{g}_i = 2$ otherwise (variant's values encoded as 1 are never the most common, since they are split in two depending on which of the two chromosomes holds which allele). We compute the *genome avatar for phenotype inference*, denoted as $\tilde{\mathbf{g}}^{\mathbf{g}}$, using a privacy configuration parameter $p_g \in [0, 1]$:

$$\tilde{g}_i^g = \begin{cases} \dot{g}_i, & \text{if } g_i = \dot{g}_i, \\ \dot{g}_i, & \text{if } g_i \neq \dot{g}_i, \text{ with probability } \ p_g, \\ g_i, & \text{if } g_i \neq \dot{g}_i, \text{ with probability } \ 1 - p_g. \end{cases} \tag{8.11}$$

Given this creation mechanism, we compute the probability that the adversary succeeds at recovering true genotypes from the avatar considering the two possible avatar values. When $\tilde{g}_i^g \neq \dot{g}_i$, the adversary is certain that the value observed is the real genotype g_i (third case in (8.11)), thus succeeds with probability one. On the other hand, when $\tilde{g}_i^g = \dot{g}_i$, the choice that maximizes the adversary's success is to guess that $g_i = \dot{g}_i$. Her success probability is $1 - p_g(1 - \Pr[\dot{g}_i])$, where the second term captures the probability of failure, i.e., $\tilde{g}_i^g = \dot{g}_i$ was a consequence of the second case in (8.11). Therefore, the privacy level computed as in (8.10) is:

$$\mathsf{Priv}_{\tilde{\mathbf{g}}^{\mathbf{g}}} = 1 - \frac{\sum_i \mathbb{1}_{\tilde{g}_i^g \neq \dot{g}_i} + \mathbb{1}_{\tilde{g}_i^g = \dot{g}_i}(1 - p_g(1 - \Pr[\dot{g}_i]))}{n}, \tag{8.12}$$

Effectively, the parameter p_g balances the privacy and decision precision provided by the avatar. The larger p_g is, the larger the difference between avatar and real genome is (more privacy), but the more different are the decisions with respect to the real genome.

We note that, when related individuals are in the same system, their avatars must be consistent with the Mendelian inheritance laws to avoid inconsistencies when the junction tree algorithm is used for genotype inference of relatives. Given the two parents' avatars generated using the method in (8.11), we construct offspring avatars by "virtually mating" the parents' avatars. To ensure conservativeness, we choose the most conservative combination that is consistent with the parents for the offspring, instead of choosing at random as happens in reality. Given this creation mechanism, the offspring's avatar is independent from the real offspring genome, and thus does not leak information. It is only related to the parents' avatars which provide the privacy stated in (8.12).

Membership inference. The membership inference risk is measured in GenoShare as the power of a test establishing whether a statistical summary (e.g., allele frequencies) of a target individual's genome is more similar to the dataset of interest or to the reference population (see Sect. 8.4.2). Therefore, in order to trigger conservative decisions, an avatar should be more similar to the dataset than the real individual's genotypes.

To build the avatar, we first check which allele contributes more to the dataset aggregate for each variant. Then, with probability p_m, we replace the target's real value with such allele. Formally, we compute the *genome avatar for membership inference*, denoted as $\tilde{\mathbf{g}}^{\mathbf{m}}$ as follows. For each variant i that is in the dataset, given a privacy parameter $p_m \in [0, 1]$:

$$\tilde{g}_i^m = \begin{cases} \max(0, g_i - 1), & \text{with probability } \ p_m, \ \text{ if } \mathsf{aaf}_i \geq f_i \\ \min(g_i + 1, 2), & \text{with probability } \ p_m, \ \text{ if } f_i > \mathsf{aaf}_i \\ g_i, & \text{with probability } \ 1 - p_m. \end{cases} \tag{8.13}$$

The parameter p_m can be used to trade-off privacy and decision precision. Following the same reasoning as for the genome avatar, the success probability of the adversary is 1 when $\mathsf{aaf}_i \geq f_i$ and $\tilde{g}_i^m = 2$, and when $\mathsf{aaf}_i < f_i$ and $\tilde{g}_i^m = 0$. In the former case, she knows that $g_i = 2$, in the latter $g_i = 0$ (third case in (8.13)). Otherwise, the success of the adversary is $\max(p_m, 1 - p_m)$, depending on the probability of replacement. Then, the term $\Pr[g_i|\tilde{g}_i^m]$ in (8.10) is:

$$\Pr[g_i|\tilde{g}_i^m] = \begin{cases} 1, & \text{if } \mathsf{aaf}_i \geq f_i \wedge \tilde{g}_i^m = 2 \\ & \vee \mathsf{aaf}_i < f_i \wedge \tilde{g}_i^m = 0 \\ \max(p_m, 1 - p_m), & \text{otherwise .} \end{cases} \tag{8.14}$$

Kinship inference. Similar to membership inference, the risk of kinship inference depends on the power of a test measuring how similar the genomes of two individuals are (see Sect. 8.4.3). Essentially, the degree of relationship inferred by this test depends on the amount of overlap weighted by the allele frequency of the sampled variants. Hence, to ensure conservativeness, avatars should be more similar to the target's relative genome that the target itself.

For an individual A, we compute the *genome avatar for kinship inference* with respect to individual B, denoted as $\tilde{\mathbf{g}}^{\mathbf{k}}$, given a privacy parameter $p_k \in [0, 1]$ as:

$$\tilde{g}_i^k = \begin{cases} g_{i,B}, & \text{if } g_i = g_{i,B}, \\ g_{i,B}, & \text{if } g_i \neq g_{i,B}, \text{ with probability } \ p_k, \\ g_i, & \text{if } g_i \neq g_{i,B}, \text{ with probability } \ 1 - p_k, \end{cases} \tag{8.15}$$

where $g_{i,B}$ is B's genotype at variant i.

Since this avatar generation process is analogous to the one for genotype inference, privacy is computed in the same way:

$$\mathsf{Priv}_{\tilde{\mathbf{g}}^{\mathbf{k}}} = 1 - \frac{\sum_i \mathbb{1}_{\tilde{g}_i \neq g_{i,B}} + \mathbb{1}_{\tilde{g}_i = g_{i,B}}(1 - p_k(1 - \Pr[g_{i,B}]))}{n}, \tag{8.16}$$

where $\Pr[g_{i,B}]$ is genotype $g_{i,B}$'s prior probability in the population.

Each individual in the database needs to have one genome avatar for kinship inference per relative in the database. When simulating the kinship inference attack, the Attack Engine uses the avatar that corresponds to the closest relative known to the adversary (e.g., because of previous releases).

8.6.2 Aggregates avatar

Aggregates avatars $\tilde{\mathbf{f}}$ are used when GenoShare receives an aggregates' request $q_m(\mathbf{f}_s)$. As only the membership inference technique makes use of the dataset, one aggregated avatar is sufficient.

Intuitively, conservative decisions for membership-inference attacks should be triggered when the aggregates avatar is more similar to the genomic information to be tested by the adversary than to the population, so individuals would be found to be in the database and GenoShare would prevent the sharing. We construct the *aggregates avatar for membership inference*, denoted as $\tilde{\mathbf{f}}$, as follows. Given privacy parameters $\gamma, p_f \in [0,1]$, for all i for which $f_i \neq g_i/2$, we sample $\delta_i^{\tilde{f}_i}$ from $\mathcal{U}[0, \gamma|f_i - g_i/2|]$, and generate $\tilde{f}_i$ as follows:

$$\tilde{f}_i = \begin{cases} f_i - \delta_i^{\tilde{f}_i} & \text{if } f_i > g_i/2 \text{ with probability } p_f \\ f_i + \delta_i^{\tilde{f}_i} & \text{if } f_i < g_i/2 \text{ with probability } p_f \\ f_i & \text{with probability} (1 - p_f)\,. \end{cases} \tag{8.17}$$

If $f_i = g_i/2$, there is no way to construct a conservative avatar, and the avatar value will take the original value f_i. Therefore, the adversary's success probability is 1 in this (very rare) case. In other cases, the adversary will infer the frequency that maximizes his success between the original f_i (third case in (8.17)) and the modified ones (first or second cases in (8.17)) depending on the parameters. The resulting privacy of this avatar is given as:

$$\begin{aligned} &\mathsf{Priv}_{\tilde{\mathbf{f}}} = \qquad (8.18) \\ &1 - \frac{1}{n} \sum_i \mathbb{1}_{f_i = g_i/2} + \mathbb{1}_{f_i \neq g_i/2} \max\left(1 - p_f, p_f \frac{\epsilon}{\gamma|f_i - g_i/2|}\right), \end{aligned}$$

where the second term in the maximum function is derived from the following. As we rely on the uniform (continuous) distribution to generate $\tilde{f}_i$, the probability of being exactly at f_i is, in general, not defined. Instead, we compute the probability of being in a small interval (represented by ϵ) around f_i, knowing $\tilde{f}_i$:

$$\Pr[|F_i - f_i| \leq \epsilon | \tilde{f}_i] = \begin{cases} 1, & \text{if } \tilde{f}_i = f_i, \\ \frac{\epsilon}{\Delta}, & \text{if } (f_i > g_i/2 \wedge f_i - \Delta \leq \tilde{f}_i < f_i) \\ & \vee (f_i < g_i/2 \wedge f_i < \tilde{f}_i \leq f_i + \Delta), \\ 0, & \text{otherwise}, \end{cases} \tag{8.19}$$

where $\Delta = \gamma|f_i - g_i/2|$. In this case, p_f can be used to trade off privacy and accuracy in a coarse manner, and the parameter γ serves to fine-tune this trade-off. Note that, for the aggregate avatar, contrary to the genome avatar, the privacy value never depends on the specific avatar value $\tilde{f}_i$, mainly because we deal here with continuous values and not three possible discrete values that significantly constrain the space of possible avatars (in conjunction with the requirement to output a conservative query answer).

8.6.3 Using avatars

The avatar generation mechanisms described above depend on configuration parameters p_x, $x \in \{g, m, k, d\}$, that define the level of privacy provided by the avatars $\mathsf{Priv}_{\tilde{\mathbf{g}}}$, resp. $\mathsf{Priv}_{\tilde{\mathbf{f}}}$. As obtaining a closed expression that expresses the relation between the level of privacy and p_x is extremely complex, deriving analytically configuration values is not possible. However, computing avatars is extremely cheap. Thus, one can efficiently search for adequate parameters (e.g., using the bisection method).

Every time GenoShare is launched, it uses as many avatars as attacks its needs to consider. However, we note that an adversary cannot learn more than a single avatar for a given position by making multiple requests. Indeed, either there is no denial and the adversary learns nothing about any avatar, or there is a denial and this denial will always be based on the same avatar (the most conservative one) for later requests. In other words, there cannot be a denial based on more than one avatar.

8.7 Using GenoShare with Real Data

We now show how GenoShare can, in practice, support privacy-conscious decisions when sharing genomic data. We consider three use cases in which the adversary makes different requests and has different background knowledge. These use cases are chosen to illustrate how GenoShare reacts to the most likely combinations of requests and background knowledge, and how the interrelations between the attacks influence GenoShare's decision.

8.7.1 Experimental Setup

Real Data. We run our experiments on the genomes of 351 individuals with admixed American ancestries (AMR) from the 1,000 Genomes Project [194]. For each individual, we sample 270k variants across all autosomal chromosomes, and take this to be a representative sample of her genome. We use 250 individuals to build the public reference panel ($\mathbf{R}$), and the remaining 101 to simulate the institution's database ($\mathbf{D}$). We also build a "sensitive" dataset formed by 50 random individuals in $\mathbf{D}$ to simulate an HIV-related cohort $\mathbf{H}$ (any other sensitive disease could be alternatively used here as an example).

GenoShare's Initialization. We set up GenoShare's Attack Engine with the three inference attacks introduced in Section 8.4. To instantiate the *phenotype inference attack*, we implemented genotype completion using Brian L. Browning's BEAGLE implementation v4 [53, 52], and the junction tree algorithm using the Netica Bayesian network Software [66]. We take the disease-variant associations for the AMR population from the GWAS Catalog [104] for genotype-phenotype mapping. To instantiate *membership* and *kinship* inference attacks, we used our own implementations of the Homer [107] and kinship coefficient [137] inference techniques.

GenoShare's Avatar Engine is set up using the generation techniques in Section 8.6. For all individuals in $\mathbf{D}$, we generate (i) a genome avatar for phenotype inference, (ii) a genome and an aggregates avatar for membership inference, if they are also part of $\mathbf{H}$;

and (iii) a genome avatar for kinship inference, if they have relatives. Finally, the Risk Measurement Engine is initialized with the risk metrics in Section 8.5, and we instantiate the Decision Engine with risk thresholds particular to each use case.

8.7.2 Use Cases

In the following, we assume that the adversary always requests data about a single target individual, or summary statistics about a single disease-related cohort. To consider multiple individuals/cohorts, it suffices to replicate the experiments for all targets.

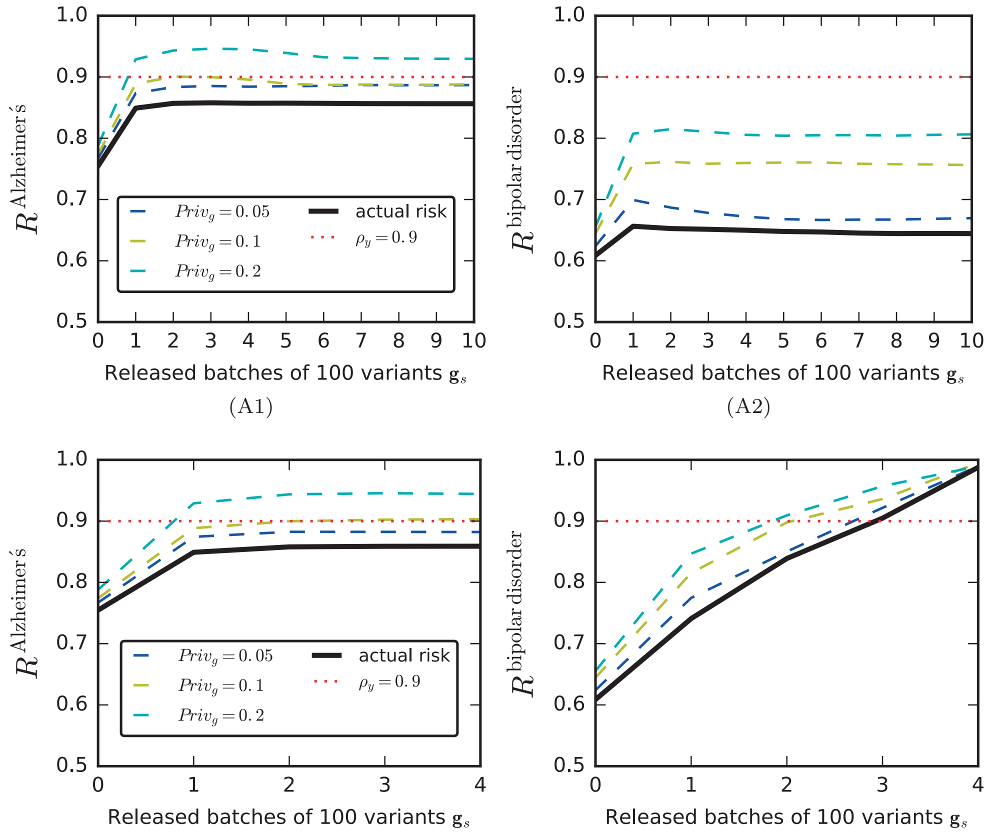
UC1: Genotype request – no background knowledge on D.
In this scenario, the institution receives consecutive requests for releasing batches of 100 variants of individual A in $\mathbf{D}$. We consider that A is part of the HIV-related cohort, $\mathbf{H}$, and one of her relatives, B, is also in $\mathbf{D}$ but not in $\mathbf{H}$. A's risk thresholds for phenotype inference risk, membership to the HIV-related cohort inference risk, and kinship inference risk are $\rho_y = 0.9, \rho_m = 0.7, \rho_k = 0.9$, respectively. We consider that the adversary's background knowledge ($\mathcal{B}$) consists of the publicly available reference panel $\mathbf{R}$.

Upon reception of a genotype request $q_g(\mathbf{g}_s)$, the institution configures GenoShare with $\mathcal{B}$ and $\boldsymbol{\rho}$ above, and launches it. Then, all the attacks in the Attack Engine are run on A's genome avatars, considering all their interdependencies.

Let us first consider the phenotype inference attack. To illustrate the evolution of A's phenotype inference risk, we consider the adversary's goal is to learn her predisposition to Alzheimer's disease and bipolar disorder. We stress, however, that GenoShare could be configured to consider disclosure of any other genomic-related clinical trait or phenotype. Figure 8.5 shows this risk's evolution as consecutive batches of variants are released. The solid line represents the risk for A's real genotype, and dashed lines for A's avatars offering privacy $\mathsf{Priv}_{\tilde{\mathbf{g}}\mathbf{g}} = \{0.05, 0.1, 0.2\}$ (maximum privacy is $\mathsf{Priv}_{\tilde{\mathbf{g}}\mathbf{g}}^{max} = 0.29$). The dotted red line represents A's threshold $\rho_y = 0.9$. The first point in each figure represents the risk before any variant is released, i.e., the prior risk for A computed as in (8.5) where the adversary's estimation of the target variants' is made according to the alternate allele frequency in the population $\mathsf{aaf}s$.

We consider two data-request patterns. The first pattern consists of a series of requests for arbitrary variants, that are not necessarily correlated with any sensitive disease. This case represents the behavior of a researcher looking for new genomic associations with a disease of interest at a genome-wide scale. The results of this experiment are shown in Figures 8.5A1 and 8.5A2. First, we observe that released data become part of the adversary's knowledge as $\mathbf{A}_{\mathbf{g}}$, and thus the risk never decreases. Second, as expected, releasing arbitrary (thus likely disease-unrelated) variants slightly affects A's predisposition inference risk for both considered diseases, when computed on the real genotypes. Yet, when avatars are used, as they contain more common variants, this growth is larger. This is because, due to the genotype completion technique that favors the estimation of common values in $\mathbf{R}$, the estimation of the avatar is better.

The second pattern consists of a series of requests for variants correlated with a specific disease, concretely schizophrenia. This represents a typical scenario in which a researcher studies variants of known significance. Results in Figures 8.5B1 and 8.5B2 show that releasing these variants does not have a particular impact on the risk of inferring A's predisposition to Alzheimer's disease as these two diseases are genetically uncorrelated [55]. Yet, because of the high genetic correlation between bipolar disorder



(A1) (A2)

(B1) (B2)

Figure 8.5: *UC1* – Disease predisposition inference risk, R^y, when releasing A's genotypes in batches of 100: effect of releasing arbitrary variants on the risk of inferring predisposition to Alzheimer's disease (A1) and bipolar disorder (A2); effect of releasing schizophrenia-related variants on the risk of inferring predisposition to Alzheimer's disease (B1), and bipolar disorder (B2).

and schizophrenia, the risk of inferring predisposition to bipolar disorder significantly increases as more schizophrenia-related variants are released.

Regarding GenoShare's decision, the larger is Priv$_{\tilde{\mathbf{g}}\mathbf{g}}$, the more conservative are the decisions based on the corresponding genome avatars, i.e., for a given inference risk level they allow to disclose less data than those based on the true genome. For instance, let us consider the risk of inferring predisposition to bipolar disorder (Figure 8.5(B2)). Given A's threshold, ρ_y, if computations were done on the real genome, GenoShare would allow to release up to 300 genotypes (intersection of the black line and the red dotted line), risking an information leak when the decision is to deny a query. This risk can be mitigated by using avatars, at the cost of releasing less data. The most protective avatar (Priv$_{\tilde{\mathbf{g}}\mathbf{g}} = 0.2$, cyan) enables the release of around 200 variants, while the less protective one (Priv$_{\tilde{\mathbf{g}}\mathbf{g}} = 0.05$, blue) permits the release of almost as many variants as with the real genome (black).

Given space constraints, in the following, we only show results for the consecutive releases of arbitrary variants. We obtained similar results for variants related to schizophre-

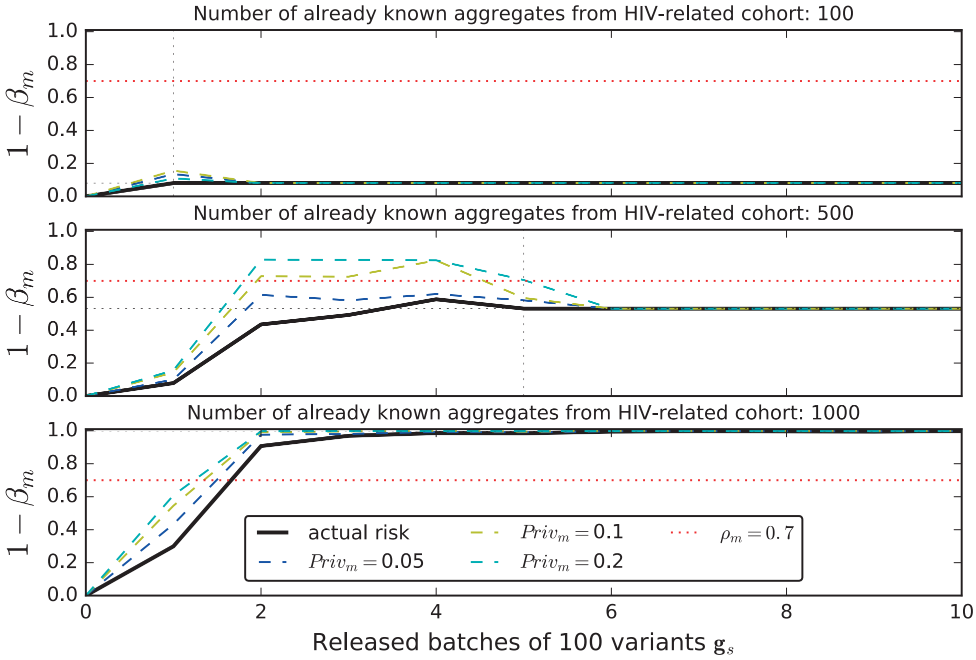
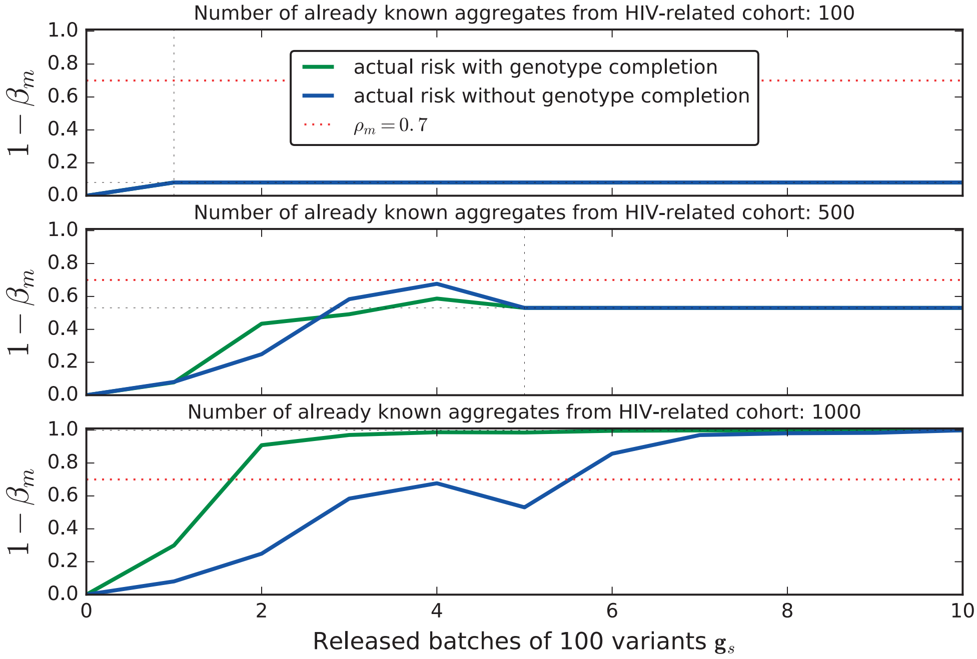


Figure 8.6: *UC1* – Power of membership inference for different levels of adversarial knowledge ($\alpha_m = 10^{-4}$).

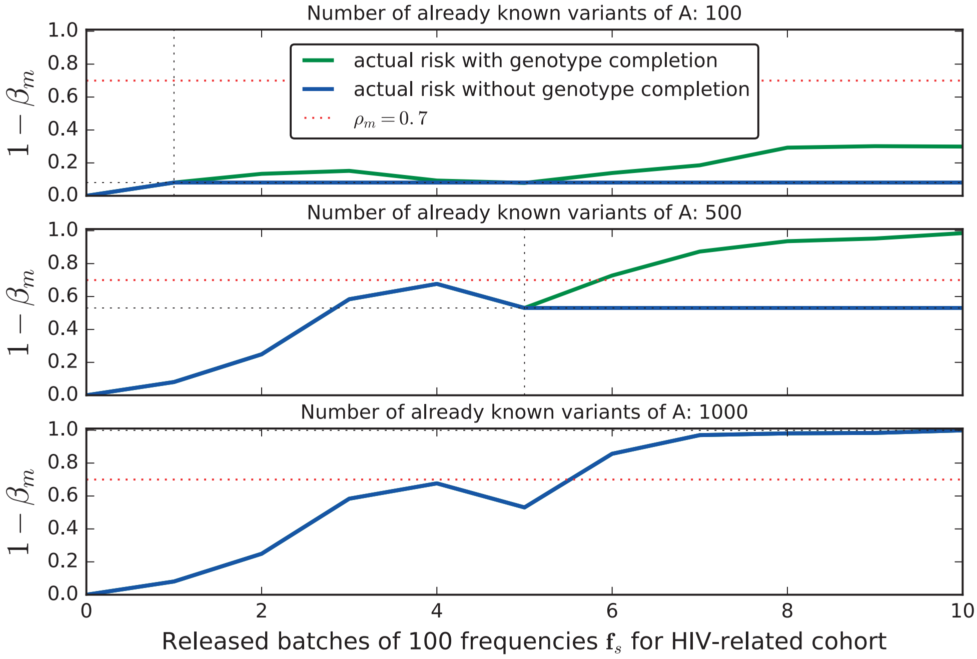
nia.

Releasing A's variants not only affects her phenotype inference risk but also her membership and kinship inference risks. We show the effect on the risk of inferring her membership to the HIV-related cohort $\mathbf{H}$ for a false positive rate $\alpha_m = 10^{-4}$ in Figure 8.6. In the different rows of Figure 8.6, we consider that the adversary has obtained an increasing number of aggregates from previous queries ($\mathbf{A_m}$) to be incorporated to his background knowledge $\mathcal{B}$. Unsurprisingly, the more genotypes are revealed, the stronger is the inference power, up to the point where the number of genotypes and aggregate statistics released are the same (vertical dotted line). Then the inference power remains constant because there are no more aggregate statistics to gain information from (horizontal dotted line). We observe that genotype completion helps membership inference by increasing the inference power before the vertical dotted line. Indeed, thanks to genotype intra-correlations, a larger number of genotypes than those made available to the adversary can be used in the attack (see Figure 8.7 for comparison with the membership inference power without the contribution of genotype completion). The behavior of the avatars (dashed lines) is similar to the previous case, privacy can be enhanced at the cost of releasing less information. We note that, the more variants are released, the less different is the avatar from the actual genotype. Hence, the inference power based on the avatar and real genome converges.

Finally, Figure 8.8 shows the effect of genotype release on risk of inferring A and B's kinship for different degrees of relatedness $d = \{1, 2, 3\}$ and a false-positive rate $\alpha_d = 10^{-4}$. Similarly to the previous case, each row assumes that the adversary has an increasing number of variants from B as part of his knowledge $\mathbf{A_g}$. We consider two cases where A and B are first and second degree relatives. In the first case (Fig. 8.8(A)),



(A) Releasing genotypes g_i when aggregates f_i are known



(B) Releasing aggregates f_i when genotypes g_i are known

Figure 8.7: Power of membership inference for $\alpha_m = 10^{-4}$: interleaving genomic and aggregated queries

we see that the kinship inference power is already maximized when 300 variants are disclosed for both individuals. Because of the closed test procedure, the power for $d > 1$ is also maximized. However, the kinship inference power for $d = 0$ (monozygotic twins) is negligible regardless of the number of variants used in the attack, as there is no possibility that this is the case. In the second case (Fig. 8.8(B)), few variants suffice to infer with significant power that the individuals are at least third degree relatives ($d = 3$). Yet, 1,000 variants are not enough to determine their real kinship, $d = 2$, with high certainty. In fact, up to 4,000 variants are necessary to maximize this inference power (see Fig. 8.9). Again avatars perform as expected, enabling a trade-off between amount of released data and privacy in case of query denials.

We recall that GenoShare denies a query as soon as any of the inference risks breaches its corresponding threshold. For instance, if we consider an avatar with $\mathsf{Priv}_{\tilde{\mathbf{g}}\mathbf{g}} = 0.05$, and adversarial background knowledge of 500 aggregates from $\mathbf{H}$ and 500 genotypes from B (second degree relative), GenoShare prevents the release of data at the fourth request because of the kinship inference risk being too high.

UC2: Genotype request – kinship background knowledge.

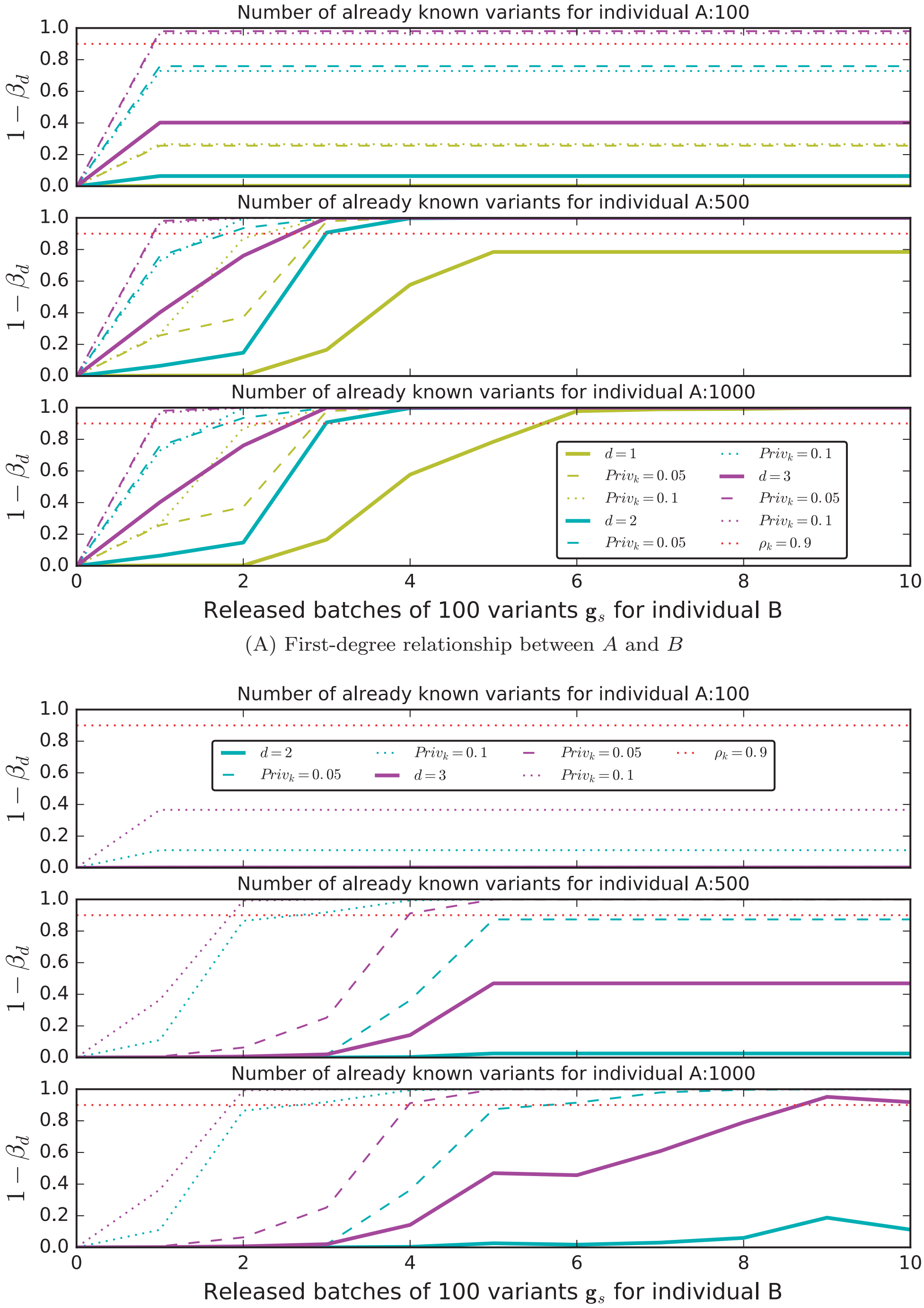
In this scenario, the institution holding $\mathbf{D}$ receives consecutive requests for releasing genotypes of arbitrary pathogenic variants (i.e., related to a disease) of individual A. A, and also her parents B and C, are part of the HIV-cohort $\mathbf{H}$. Their degree of kinship is already known by the adversary and it is part of his background knowledge $\mathcal{B}$. For example, he could have inferred this information through the kinship inference attack when A's, B's and C's genotypes were released by GenoShare , or if their kinship information was publicly available on Facebook [110]. Moreover, we assume that A, B, and C have different privacy concerns: A is not worried about any potential privacy breach and sets very permissive thresholds $\boldsymbol{\rho}_A = \{\rho_y = \rho_m = 0.95\}$, whereas B and C have more restrictive preferences $\boldsymbol{\rho}_B = \{\rho_y = \rho_m = 0.7\}$ and $\boldsymbol{\rho}_C = \{\rho_y = \rho_m = 0.8\}$ (Note that, as kinship is known, we disregard the kinship threshold.)

Upon reception of a genotype request $q_g(\mathbf{g}_s)$, the institution configures GenoShare with $\mathcal{B}$, $\boldsymbol{\rho}_A$, $\boldsymbol{\rho}_B$ and $\boldsymbol{\rho}_C$, and launches it. GenoShare runs both phenotype and membership inference considering their interrelation, and computes the risk of a privacy breach for the three individuals. Figure 8.10(A) illustrates the evolution of A's, B's, and C's predisposition to type 2 diabetes inference risk. Because of the kinship effect, releasing A's genotypes has an effect on B's and C's risk computed with both the real genotype (solid) and avatars (dashed). We note that, because the three individuals are involved, it is enough that at least one of their risk thresholds is exceeded to deny the request. For instance, if avatars are used, even though the first query would be deemed safe for A (blue), it would be denied because it implies that the risk for B (yellow) goes above her threshold.

Because it improves genotype completion, kinship information also significantly affects membership inference. Figure 8.10(B) illustrates the evolution of the risk of membership to the HIV-related cohort $\mathbf{H}$ inference for A, B, and C for $\alpha_m = 10^{-4}$ when 1,000 aggregates are already known to the adversary. Similarly to the previous use case, we observe how releasing A's genotypes also increases the membership inference risk for B and C. Also, observe that, as in the third row of Figure 8.6, genotype completion helps the adversary by significantly increasing his inference power (even if only 100 genotypes are available after the first query, the adversary can exploit the 1000 aggregates that are available).

UC3: Aggregate request – no background knowledge on D.

In this last scenario, the institution holding $\mathbf{D}$ receives consecutive requests for releas-



(A) First-degree relationship between A and B

(B) Second degree relationship between A and B

Figure 8.8: *UC1* – Power of kinship inference for different degrees ($\alpha_d = 10^{-4}$).

ing aggregate statistics $\mathbf{f}_s$ of the HIV-related cohort $\mathbf{H}$. We consider the same privacy preferences and background knowledge as in UC1. Upon reception of a request for aggregate data, the institution configures GenoShare with public background knowledge and launches it. In this particular case, GenoShare only runs the membership inference attack

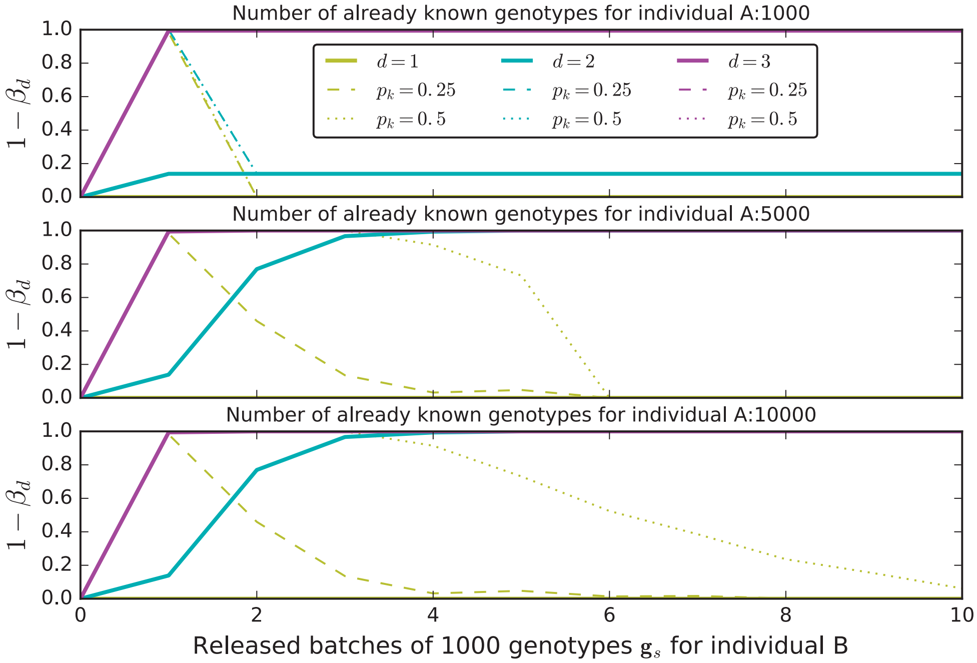
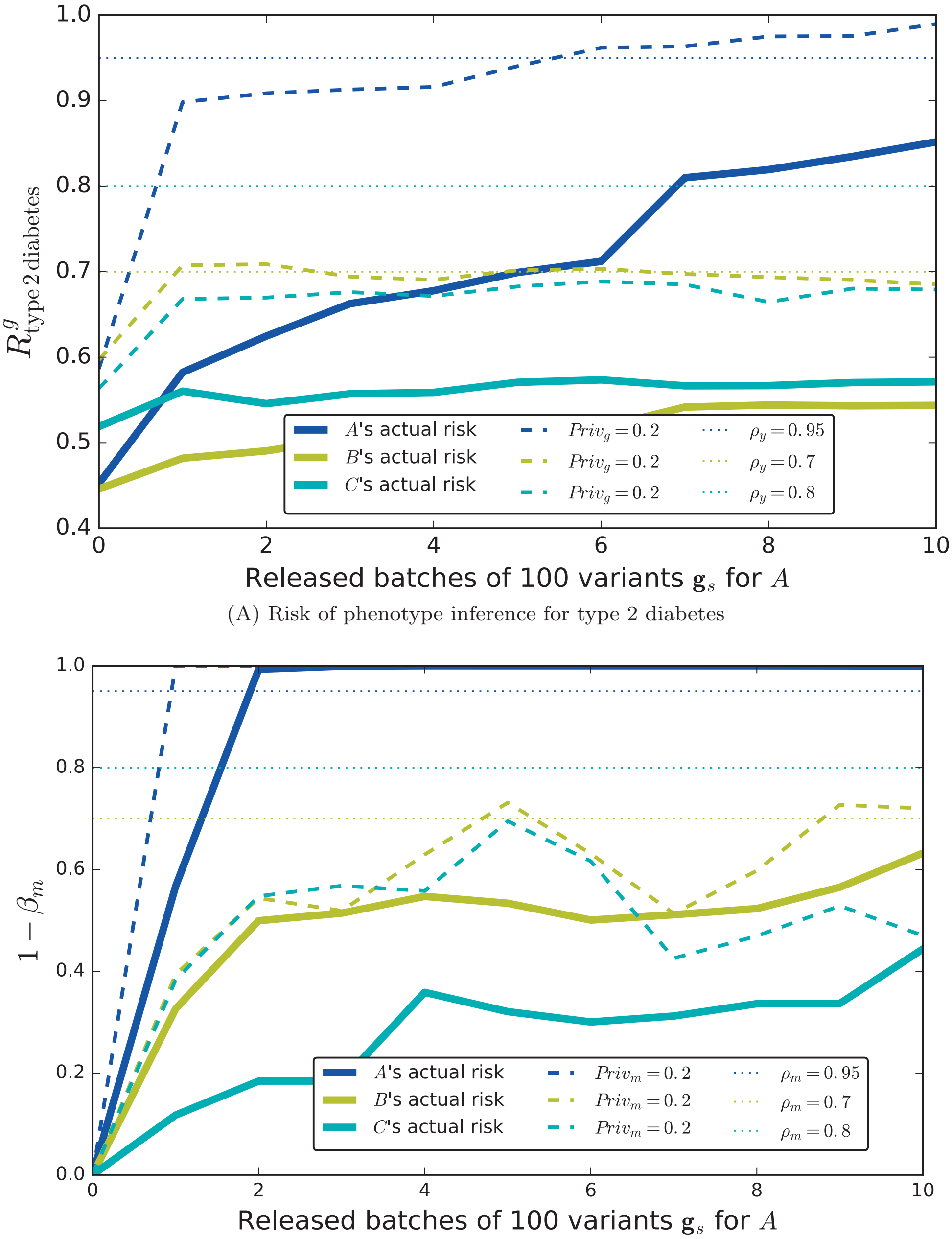


Figure 8.9: Power of kinship inference for different degrees and $\alpha_d = 10^{-4}$ in the case of second degree relationship between A and B.

because it is the only attack for which obtaining new aggregates helps the adversary.

Figure 8.11 shows the effect of a series of aggregate data releases on the risk of inferring A's membership to the HIV-related cohort $\mathbf{H}$, for a false-positive rate $\alpha_m = 10^{-4}$ and different amount of A's variants available to the adversary. Unsurprisingly, the more aggregates are released, the higher the inference power. It is important to note that, thanks to the *genotype completion* carried out in the phenotype inference, the adversary can use every new released aggregated frequencies even if the corresponding genotypes of A have not yet been revealed (i.e., after the black dotted lines cross). Thus, the risk of membership inference keeps growing with every extra frequency observed by the adversary. Without the contribution of the genotype completion, inference power stays constant since additional aggregate statistics cannot be used by the membership attack (see Figure 8.7(B)). As in previous cases, we observe how avatars enable to trade off privacy for amount of data released.

Finally, this use case also shows GenoShare's utility in the case where the adversarial background knowledge consists of the full genome of some individuals in $\mathbf{H}$. For example, individuals that put their genome online (e.g., on OpenSNP), or have been sequenced at a direct-to-customer genomic service (e.g., 23andMe) that keeps a copy of their genome. In these circumstances, as the genome is already known, the only attack that GenoShare can mitigate is the membership inference attack based on aggregate data requests. We emphasize that, for GenoShare to not underestimate the membership inference risk in such a case, it is the responsibility of the individuals to communicate to the institution holding their genomic data that other copies are available elsewhere. In this case, GenoShare's background knowledge should be set to account for the adversary's knowledge of the complete target's genome before its execution.



(A) Risk of phenotype inference for type 2 diabetes

(B) Risk of membership inference ($\alpha_m = 10^{-4}$) when the adversary already knows 1,000 aggregates from the HIV-related cohort

Figure 8.10: *UC2* – Disease predisposition and membership inference risk for A, B, and C

8.7.3 GenoShare's Performance

GenoShare is not intended to be a real time tool, but to be run offline. We conduct performance measurements on an 8-core Intel Xeon CPU E3-1270 V2 processor 3.50GHz, and 32 GB of memory, running Debian Linux. We measure the time required for computing the inference risks for the three considered attacks by executing GenoShare 10 times on 10 different requests of 100 arbitrary variants, and report the average over the 100

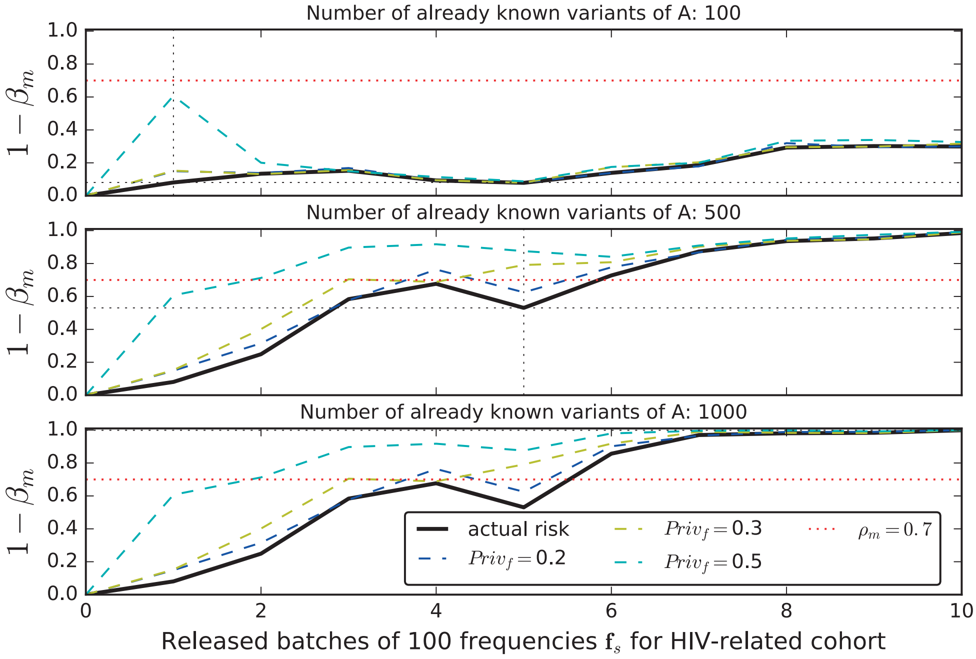


Figure 8.11: *UC3* – Effect of releasing batches of 100 aggregates on the risk of membership inference ($\alpha_m = 10^{-4}$, $\gamma = 0.25$).

experiments. Although the considered inference techniques are naturally parallelizable, our measurements are made on a single thread of execution and thus represent an upper bound for computation time.

We find that, despite its apparent complexity, the use of GenoShare entails very reasonable overhead. As expected, since they only require fast arithmetic operations, the computation time of both the kinship and membership inference attacks is within the second hence negligible, and grows linearly with the number of variants to be tested. Similarly, the generation of avatars, which is performed only once during GenoShare's initialization, is also negligible. On the other hand, the phenotype inference attack, that uses BEAGLE's Java implementation for *genotype completion*, takes on average around 8 minutes and 14 seconds at each new request. We note that this timing is strongly influenced by our choice of running BEAGLE with default parameters and limiting the amount of memory allocated for the Java virtual machine to 2GB, and could be reduced by optimizing the configuration (e.g., with a typical 22-node cluster it requires less than 23 seconds). We also note that the genotype completion computation time depends both on the number of variants to be considered across the genome (in our case 270k variants), and on the number of individuals in the reference panel $\mathbf{R}$ (in our case 250 individuals). We refer the reader to the original publication for more details on how BEAGLE scales [53, 52].

In terms of storage, for each individual in the database, GenoShare requires one avatar per kind of attack instantiated. The size of an avatar is at most the size of the set of human genetic variations, i.e., roughly 1% of a full genome. Thus, GenoShare's storage requirements are certainly practical.

8.8 Related Work

We revise attacks and defense mechanisms for genomic privacy which are most relevant to our work. We refer the reader to existing surveys [79, 148] for an exhaustive review of the state of the art.

Attacks. A first group of relevant attacks, referred to as *attribute disclosure*, can be subdivided into two categories. The first category, attribute disclosure via genomic completion, includes all techniques that rely on the intra- and inter-genome correlations to infer unobserved variants [128] to reduce genomic privacy of individuals [175] and relatives [111]. In this work, we improve previous inference techniques by taking both familial correlations and intra-genome correlations using *genotype completion*. The second category, attribute disclosure via membership inference, exploits knowledge of summary statistics of a given dataset to infer that a known genome is part of it [62, 107, 113, 176, 207, 210, 220]. As such datasets are typically associated to a disease of interest, inferring membership unveils very sensitive attributes. Commonly exploited statistics are allele frequency and genotype counts, or statistics about linkage disequilibrium. In this work, we explore for the first time the effectiveness of these attacks in presence of incomplete information, and the benefits of genotype completion on membership inference.

A second group of relevant papers deals with *kinship inference*. Previous works show how to exploit inter-genome correlations to infer familial relationships based on the amount of genomic data shared by individuals in genome-wide association studies with distinct subpopulations [137] or in admixed populations [199]. Recently, Arthur et al. have proposed a toolkit for analyzing large cohorts of whole-genome sequenced samples that includes kinship inference relying on state-of-the-art methods [24]. Our contribution is the framing of kinship inference as from a privacy perspective, both as an attack to reveal familial relationship and as complement to genotype completion to increase the amount of genetic information that can be inferred about an individual.

Finally, we revise *re-identification attacks* that de-anonymize genomic data by relying on auxiliary knowledge. Gymrek et al. have proposed an attack of this kind [99], which uses short tandem repeats on the Y chromosome (Y-STRs) to map anonymous genomes to those available in a recreational genetic genealogy database online to recover the surname of their (male) owner. Similarly, Humbert et al. link de-identified genotypes to online profiles, such as those from online social networks, by relying on genomic variants that influence phenotypic traits [112]. In this paper, we do not consider these attacks for space constraints and because they highly rely on the access to external background knowledge that may be difficult to access (e.g., the Y-STR database used by Gymrek et al. has been put offline after this attack was made public). Yet, we note that GenoShare's Attack Engine can easily accommodate re-identification attacks by integrating it in the different engines.

Protection mechanisms. In addition to de-identification, which is a necessary but not sufficient protection method as shown above, there exist two approaches for genomic privacy protection. The main idea behind the first approach is to properly apply noise, e.g., achieving differential privacy, on summary statistics for protecting a study participants' privacy and thus thwarting attribute disclosure attacks [116, 182, 201, 219]. Fredrikson et al. show that, however, using differential privacy in pharmacogenetics can lead to an unacceptable loss of utility, e.g., exposure of patients to an increased risk of stroke, bleeding events, and mortality [88]. Also, practitioners require exact genomic data to avoid false genotype-phenotype associations. Bhaskar et al. propose a noiseless version of differential privacy but their solution makes some statistical assumptions on the data that are too restrictive for the genomics setting [45]. Our solution does not perturb the released aggregated data at all, regardless of the data distribution. Moreover, the aforementioned protection mechanisms are not suitable for the release of an individual's variants. The second approach relies on secure storage and processing [36, 70, 109, 148, 211, 141] which are complementary to our proposed solution. In fact, secure processing protects only the data while processing, but GenoShare considers all the disclosed/shared information, including also the results of a computation.

8.9 Summary

Academic solutions for privacy-preserving sharing of genomic data have mostly focused on data perturbation. Such solutions, however, damage the utility of the data and thus have not been accepted by practitioners. In this work, we introduce GenoShare, a framework that supports privacy-informed decision-making when sharing genomic data. GenoShare quantifies the risk of sensitive attribute disclosure, and prevents the automatic sharing of data if the risk is deemed too high with respect to privacy thresholds encoded using novel meaningful sensitive attribute-oriented metrics. Otherwise, it releases *exact data* as requested by genomics research practitioners. Furthermore, GenoShare implements avatar genomes to protect individuals' real genotypes from inferences stemming from query denials.

To the best of our knowledge, GenoShare is the first framework to jointly consider relevant attacks in genomic privacy in presence of incomplete information. It provides a principled answer to the privacy concerns that have plagued the genomic community for the last decade, and thus it is a firm step forward to enable the responsible and privacy-respecting use of genomic data in research and medical environments. We hope that it will dramatically improve the current situation in institutions, thereby accelerating the slow and costly processes carried out by committees and lawyers by serving as support for more informed decisions.

Although we have focused on the protection of genomic data, the systematic principles underlying GenoShare make it suitable to deal with other data types where correlation with sensitive information can be detrimental for an individual's privacy, e.g., other 'omics' data such as transcriptomic. Furthermore, GenoShare can also be used to understand the risk incurred when voluntarily disclosing information to find others who have a similar rare disease and share experiences as on PatientsLikeMe [153] or to safely enjoy recreational genomics or direct-to-consumer genomics services. It is also worth noting that during operation, GenoShare creates evidence that can be associated with decisions related to the release of genomic data, hence providing support for accountability systems.

Chapter 9

Conclusion

In this thesis, we have investigated the problem of developing new and practical privacy-enhancing technologies (PETs) for the protection of medical and genomic data in order to foster their adoption in the medical field. In particular, we have proposed new solutions for storing, processing and sharing medical and genomic data in a privacy-preserving way by specifically addressing concerns that affect two main areas of precision medicine: clinical care and medical research.

In Part I, we began by showing how to securely outsource the storage of raw genomic data for use in clinical care for in-depth genomic analyses or for clinical trials. We proposed a new privacy-preserving architecture where the raw genomic data of patients is stored at a centralized biobank in encrypted form. Our scheme enables a medical unit (e.g., a hospital or a pharmaceutical company) to privately retrieve a subset of the patients' raw genomic data, without revealing to the biobank the nature of the analysis to be performed. Moreover, our proposed scheme enables the biobank to mask particular parts of the retrieved data that the patient does not want to disclose. We implemented the proposed scheme and demonstrated its practicality. We then proposed a new system for privacy-preserving testing in the clinic by using homomorphically encrypted genetic variants. In order to obtain approval from end users, we used DNA-based prediction of HIV-related outcomes as a model for exploring its use in a clinical operational environment. Our solution enables a medical unit to securely compute ancestry information and genetic risk scores from the encrypted genomic data stored at a storage and processing unit by exploiting the homomorphic properties of the cryptosystem at the expense of practical computational and storage overhead. We implemented the proposed model and deployed it for testing at five outpatient clinics of the the Swiss HIV Cohort Study. We evaluated the feedback from HIV specialists who tested our system on 230 HIV-positive patients genotyped at 4,149 genetic markers. This enabled us to gain unprecedented insights that can guide the design of new PETs for clinical genomics in the future.

In Part II, we first investigated the problem of securely re-using, in research, genomic and medical data initially generated for clinical care. As such, we developed a new privacy-preserving system that is built on top of the state-of-the-art framework for clinical research, namely *Informatics for Integrating the Biology and the Bedside* (i2b2). Our system improves the basic security guarantees of i2b2 by enabling the outsourcing of large genomic data to a central untrusted third party, such as a public cloud, and the exploration of this data under encryption. Our proposed solution makes use of lattice-based somewhat-homomorphic encryption to protect data confidentiality at rest and during computation and (optionally) makes use of result-obfuscation to achieve differential privacy and mitigate the re-identification risk. We implemented the proposed solution and tested it as a service of the clinical research data-warehouse of the Lausanne University Hospital. We demonstrated that it facilitates the access to sensitive genomic information that is otherwise prohibitive due to privacy and security concerns. The performance of our system is such that it can be efficiently used in standard feasibility studies, as it enables a researcher to simultaneously obtain aggregate information about 3,000 genetic variants, from a cohort of 5,000 patients in less than 5 seconds. Then, we studied the problem of privacy-conscious data sharing, which is crucial for assembling large amounts of data necessary for medical research, especially when the data cannot be stored at a central third party. In this context, we first evaluated the privacy risks that affect simple systems for the discovery of genetic variants (or "beacons") that enable researchers to obtain information about the presence of a given genetic variant in the database of each participating site. As a result, we proposed three practical strategies for thwarting the re-identification risks and discussed the different privacy/utility trade-offs they introduce. We demonstrated the effectiveness of these strategies on real data from the 1,000 Genomes Project in the context of the Beacon Project of the Global Alliance for Genomics and Health. Then, still in the context of medical data sharing, we proposed the first operational system that enables a federation of clinical sites to collectively protect their data against cyber attacks and to still be able to share it in a privacy-preserving way. Our system uses collective homomorphic encryption, deterministic encryption and a new method for generating dummy records in order to enable an investigator to explore cohorts of clinical and genomic data distributed across several clinical sites and find eligible patients for research studies. To be easily adopted and used in operational environments, our solution builds on top of widespread technology from the biomedical informatics community. We implemented and deployed our system in a real network of three institutions, and we demonstrated that it scales to millions of encrypted clinical records and hundreds of institutions. Finally, we proposed a new framework that complements the previous systems and enables practitioners and administrators to systematically reason about the risk of revealing privacy-sensitive attributes that patients are unwilling to share (e.g., health status, kinship, physical traits), caused by the disclosure of exact genomic data. In particular, in order to support privacy-informed decisions for exact genomic data sharing, our framework evaluates the potential loss of genomic privacy due to potential genomic-oriented inference attacks and their interactions. We demonstrated the capabilities of the proposed system by instantiating it with three of the most important inference attacks. We showed how it can be used in practice to detect leakage of sensitive attributes by using real data from the 1,000 Genomes Project.

In conclusion, we see that privacy protection of sensitive medical data, and particularly genetic data, is an important concern as the healthcare system increasingly suffers from data breaches. This concern must be addressed in order to fully realize the promise of personalized medicine. In this thesis, we propose concrete and practical technical solutions and demonstrate that these solutions can be deployed and have an impact in the real world. In particular, we show that, despite the general skepticisms around the apparent unpracticality and difficulty of these technologies, PETs-based solutions can be made efficient, deployable and usable in real operation settings, both for clinical care and research, and can represent concrete enablers to overcome the privacy and security concerns that are currently slowing down the advancement of personalized medicine. This thesis bridges the gap between the medical community and the information security community. Yet, an additional effort, especially on the engineering side, is required to make these new and sophisticated technologies be core components of next-generation health information systems. From the research perspective, an important lesson that can be drawn from this thesis is that we have to remove the unnecessary complexity of these tools and identify the best privacy, efficiency, and usability trade-offs in order to foster their acceptance and adoption by end-users.

Finally, we believe that this thesis represents a first systematic guide for future privacy and security researchers trying to build new PETs-based solutions for protecting medical and genomic data, as it shows the entire development process by spanning from the definition of the privacy and security requirements to the design of a potential privacy-preserving system and its deployment to operational environments. Addressing privacy issues in healthcare remains a great challenge that will increasingly require long-term and interdisciplinary collaboration among geneticists, healthcare providers, ethicists, lawmakers, and computer scientists.

Future Work

Despite the concrete solutions provided in this thesis for the secure management of medical and genomic data, a significant effort remains to be made on the technical side in order to fully realize the promise of a privacy-conscious personalized medicine. Future work will have to address unresolved problems such as (a) the long-term protection of genomic data (current solutions can provide protection for only a few decades), (b) the identification of optimal trade-offs between the risk of re-identification and the utility of the data (solutions based on statistical obfuscation are inherently difficult to achieve due to the high dimensionality of the data), (c) the lack of flexibility of the current cryptographic-based solutions and their computational cost, and (d) the quantification of secondary leakage stemming from inferences based on auxiliary meta-data.

Moreover, substantial work will be needed to increase the number of pilot studies that are relevant to current initiatives in collaborative medical and genomic research. Technology transfer is key for the widespread adoption of privacy-enhancing technologies that have the potential to enable applications that are difficult, or even impossible, at the moment due to of legal and policy restrictions (e.g., cross-border transfer of health and genomic data). Last but not least, the management of research consent is one of the major future challenges because of its confidential and dynamic nature. New models that ensure transparency, integrity and automatic enforcement of the consent will have to be developed and integrated with current privacy-enhancing technologies in order to establish a balanced participatory relationship between citizens and researchers.

Bibliography

[1] Ancestry® — Genealogy, Family Trees & Family History Records. https://www.ancestry.com/. Last Accessed: March 13, 2018. [cited at p. 10]

[2] Exac browser. http://exac.broadinstitute.org. Last Accessed: March 13, 2018. [cited at p. 66]

[3] Federal Act on Data Protection. https://www.admin.ch/opc/en/classified-compilation/19920153/index.html. Last Accessed: March 13, 2018. [cited at p. 56]

[4] i2b2 installations. https://www.i2b2.org/work/i2b2_installations.html. Last Accessed: March 13, 2018. [cited at p. 60]

[5] i2b2 software. https://www.i2b2.org/software/index.html. Last Accessed: March 13, 2018. [cited at p. 79]

[6] Igrs sample na06984. ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data/NA06984/. Last Accessed: March 13, 2018. [cited at p. 30, 31, 32]

[7] Medical data breaches raising alarm. http://articles.washingtonpost.com/2012-06-02/national/35462326_1_data-breaches-medical-data-social-security-numbers. Last Accessed. March 13, 2018. [cited at p. 22]

[8] The sam format. http://samtools.sourceforge.net/SAM1.pdf. Last Accessed: March 13, 2018. [cited at p. 20]

[9] SNP's of genetically inherited traits, conditions and diseases. http://www.eupedia.com/genetics/medical_dna_test.shtml. Last Accessed: March 13, 2018. [cited at p. 31]

[10] Swiss hiv cohort study (shcs). http://www.shcs.ch/. Last Accessed: March 13, 2018. [cited at p. 36]

[11] The variant call format specification - vcfv4.3 and bcfv2.2. http://samtools.github.io/hts-specs/VCFv4.3.pdf. Accessed: March 13, 2018. [cited at p. 63]

[12] Preventing Genetic Identity Theft — The Scientist Magazine®. http://www.the-scientist.com/?articles.view/articleNo/32796/title/Preventing-Genetic-Identity-Theft/, 2012. Last Accessed: March 13, 2018. [cited at p. 2]

[13] Remarks by the President in Precision Medicine Panel Discussion — whitehouse.gov. https://obamawhitehouse.archives.gov/the-press-office/2016/02/25/remarks-president-precision-medicine-panel-discussion, 2016. Last Accessed: March 13, 2018. [cited at p. 2]

[14] Epic. https://epic.org/privacy/reidentification, 2017. Last Accessed: March 13, 2018. [cited at p. 89]

[15] Exploring the Promise and Perils of Sharing Your DNA. https://undark.org/article/dna-ancestry-sharing-privacy-23andme/, 2017. Last Accessed: March 13, 2018. [cited at p. 2]

[16] Is privacy the price of precision medicine? — OUPblog. https://blog.oup.com/2017/03/privacy-precision-medicine/?utm{_}source=twitter{&}utm{_}medium=oupacademic{&}utm{_}campaign=oupblog, 2017. Last Accessed: March 13, 2018. [cited at p. 2]

[17] Largest Healthcare Data Breaches of 2016. https://www.hipaajournal.com/largest-healthcare-data-breaches-of-2016-8631/, 2017. Last Accessed: March 13, 2018. [cited at p. 2]

[18] The Health Data Conundrum - The New York Times. https://www.nytimes.com/2017/01/02/opinion/the-health-data-conundrum.html?{_}r=0, 2017. Last Accessed: March 13, 2018. [cited at p. 2]

[19] 23andMe Inc. 23andme. https://www.23andme.com, 2017. Last Accessed: March 13, 2018. [cited at p. 10]

[20] Nabil R Adam and John C Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515–556, 1989. [cited at p. 104]

[21] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Order preserving encryption for numeric data. *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 563–574, 2004. [cited at p. 12, 21]

[22] Carlos Aguilar-Melchor, Joris Barrier, Serge Guelton, Adrien Guinet, Marc-Olivier Killijian, and Tancrede Lepoint. NFLlib: NTT-based fast lattice library. In *Cryptographers' Track at the RSA Conference*, pages 341–356. Springer, 2016. [cited at p. 78]

[23] M Arab-Alameddine, J Di Iulio, Thierry Buclin, M Rotger, R Lubomirov, M Cavassini, A Fayet, LA Décosterd, Chin Bin Eap, J Biollaz, et al. Pharmacogenetics-based population pharmacokinetic analysis of efavirenz in HIV-1-infected individuals. *Clinical Pharmacology & Therapeutics*, 85(5):485–494, 2009. [cited at p. 46]

[24] Rudy Arthur, Ole Schulz-Trieglaff, Anthony J Cox, and Jared O'Connell. AKT: ancestry and kinship toolkit. *Bioinformatics*, 33(1):142–144, 2016. [cited at p. 154]

[25] Paolo A Ascierto, John M Kirkwood, Jean-Jacques Grob, Ester Simeone, Antonio M Grimaldi, Michele Maio, Giuseppe Palmieri, Alessandro Testori, Francesco M Marincola, and Nicola Mozzillo. The role of braf v600 mutation in melanoma. *Journal of translational medicine*, 10(1):85, 2012. [cited at p. 120]

[26] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30, Feb. 2006. [cited at p. 38]

[27] Brian D Athey, Michael Braxenthaler, Magali Haas, and Yike Guo. tranSMART: an open source and community-driven informatics and data sharing platform for clinical and translational research. *AMIA Summits on Translational Science Proceedings*, 2013:6, 2013. [cited at p. 105]

[28] E. Ayday, J. L. Raisaro, and J. P. Hubaux. Personal use of the genomic data: Privacy vs. storage cost. *Proceedings of IEEE Global Communications Conference, Exhibition and Industry Forum (Globecom)*, 2013. [cited at p. 17]

[29] E. Ayday, J. L. Raisaro, and J. P. Hubaux. Privacy-enhancing technologies for medical tests using genomic data. *(short paper) in 20th Annual Network and Distributed System Security Symposium (NDSS)*, 2013. [cited at p. 17]

[30] E. Ayday, J. L. Raisaro, P. J. McLaren, J. Fellay, and J. P. Hubaux. Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data. *Proceedings of USENIX Security Workshop on Health Information Technologies (HealthTech)*, 2013. [cited at p. 17]

[31] Erman Ayday, Emiliano De Cristofaro, Jean-Pierre Hubaux, and Gene Tsudik. The chills and thrills of whole genome sequencing. *Computer*, 2013. [cited at p. 1, 17]

[32] Erman Ayday, Jean Louis Raisaro, Urs Hengartner, Adam Molyneaux, and Jean-Pierre Hubaux. Privacy-preserving processing of raw genomic data. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 133–147. Springer, 2014. [cited at p. 7]

[33] Erman Ayday, Jean Louis Raisaro, Jacques Rougemont, and Jean-Pierre Hubaux. Protecting and evaluating genomic privacy in medical tests and personalized medicine. In *Proceedings of the 12th Workshop on Privacy in the Electronic Society*. ACM, 2013. [cited at p. 51]

[34] Michael Backes, Pascal Berrang, Anna Hecksteden, Mathias Humbert, Andreas Keller, and Tim Meyer. Privacy in epigenetics: Temporal linkability of microrna expression profiles. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium*, pages 1223–1240. USENIX Association, 2016. [cited at p. 139]

[35] Pierre Baldi, Roberta Baronio, Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. Countering GATTACA: Efficient and secure testing of fully-sequenced human genomes. *Proceedings of ACM CCS '11*, pages 691–702, 2011. [cited at p. 17]

[36] Pierre Baldi, Roberta Baronio, Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. Countering GATTACA: Efficient and secure testing of fully-sequenced human genomes. In *ACM Conference on Computer and Communications Security, (CCS)*, pages 691–702, 2011. [cited at p. 155]

[37] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. Recommendation for key management part 1: General (revision 3). *NIST special publication*, 800(57):1–147, 2012. [cited at p. 51]

[38] Ludovic Barman, Mohammed-Taha El Graini, Jean Louis Raisaro, Erman Ayday, and Jean-Pierre Hubaux. Privacy Threats and Practical Solutions for Genetic Risk Tests. In *2nd International Workshop on Genome Privacy and Security (GenoPri"15)*, 2015. [cited at p. 41]

[39] Pablo Barreiro, José Vicente Fernández-Montero, Carmen De Mendoza, Pablo Labarga, and Vincent Soriano. Pharmacogenetics of antiretroviral therapy. *Expert opinion on drug metabolism & toxicology*, 10(8):1119–1130, 2014. [cited at p. 36]

[40] Johes Bater, Gregory Elliott, Craig Eggen, Satyender Goel, Abel Kho, and Jennie Rogers. SMCQL: secure querying for federated databases. *Proceedings of the VLDB Endowment*, 10(6):673–684, 2017. [cited at p. 126]

[41] Linnea M Baudhuin, W Edward Highsmith, Jennifer Skierka, Leonard Holtegaard, Brenda E Moore, and Dennis J O'Kane. Comparison of three methods for genotyping the ugt1a1 (ta) n repeat polymorphism. *Clinical biochemistry*, 40(9):710–717, 2007. [cited at p. 47]

[42] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. *Advances in Cryptology-CRYPTO 2007*, pages 535–552, 2007. [cited at p. 12]

[43] Daniel J Bernstein. The salsa20 family of stream ciphers. *Lecture Notes in Computer Science*, 4986:84–97, 2008. [cited at p. 32]

[44] Julie Bertrand, Céline Verstuyft, Monidarin Chou, Laurence Borand, Phalla Chea, Kuy Huong Nay, François-Xavier Blanc, France Mentré, Anne-Marie Taburet, Thim Sok, et al. Dependence of efavirenz-and rifampicin-isoniazid–based antituberculosis treatment drug-drug interaction on cyp2b6 and nat2 genetic polymorphisms: Anrs 12154 study in cambodia. *The Journal of infectious diseases*, 209(3):399–408, 2013. [cited at p. 46]

[45] Raghav Bhaskar, Abhishek Bhowmick, Vipul Goyal, Srivatsan Laxman, and Abhradeep Thakurta. Noiseless database privacy. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 215–232. Springer, 2011. [cited at p. 155]

[46] Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. Plausible deniability for privacy-preserving data synthesis. *PVLDB*, 10(5):481–492, 2017. [cited at p. 139]

[47] Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'Neill. Order-preserving symmetric encryption. *Proceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques*, 2009. [cited at p. 13]

[48] David Botstein and Neil Risch. Discovering genotypes underlying human phenotypes: past successes for mendelian disease, future approaches for complex disease. *Nature genetics*, 33(3s):228, 2003. [cited at p. 9]

[49] Zvika Brakerski, Craig Gentry, and Shai Halevi. Packed ciphertexts in lwe-based homomorphic encryption. In *Public-Key Cryptography–PKC 2013*, pages 1–13. Springer, 2013. [cited at p. 62]

[50] Emmanuel Bresson, Dario Catalano, and David Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. *Proceedings of Asiacrypt 03*, pages 37–54, 2003. [cited at p. 38, 47, 51]

[51] Broad Institute. Skin Cutaneous Melanoma Datasets. `http://www.cbioportal.org/study?id=skcm_broad#summary`. Last Accessed: March 13, 2018. [cited at p. 120]

[52] Brian L Browning and Sharon R Browning. Genotype imputation with millions of reference samples. *The American Journal of Human Genetics*, 98(1):116–126, 2016. [cited at p. 144, 153]

[53] Sharon R. Browning and Brian L. Browning. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *The American Journal of Human Genetics*, 81(5):1084–1097, 2007. [cited at p. 144, 153]

[54] Liam R Brunham and Michael R Hayden. Whole-genome sequencing: the new standard of care? *Science*, 336(6085):1112–1113, 2012. [cited at p. 9]

[55] Brendan Bulik-Sullivan, Hilary K Finucane, Verneri Anttila, Alexander Gusev, Felix R Day, Po-Ru Loh, Laramie Duncan, John RB Perry, Nick Patterson, Elise B Robinson, et al. An atlas of genetic correlations across human diseases and traits. *Nature genetics*, 2015. [cited at p. 145]

[56] M. Canim, M. Kantarcioglu, and B. Malin. Secure management of biomedical data with cryptographic hardware. *Information Technology in Biomedicine, IEEE Transactions on*, 16(1):166–175, 2012. [cited at p. 61]

[57] Ethan Cerami, J Gao, U Dogrusoz, BE Gross, SO Sumer, BA Aksoy, A Jacobsen, CJ Byrne, ML Heuer, E Larsson, et al. The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. cancer discov. 2012; 2: 401–4. doi: 10.1158/2159-8290. *Nat Methods*, 7:92–93, 2012. [cited at p. 120]

[58] Feng Chen, Shuang Wang, Xiaoqian Jiang, Sijie Ding, Yao Lu, Jihoon Kim, S Cenk Sahinalp, Chisato Shimizu, Jane C Burns, Victoria J Wright, et al. Princess: Privacy-protecting rare disease international network collaboration via encryption through software guard extensions. *Bioinformatics*, 33(6):871–878, 2016. [cited at p. 126]

[59] Y. Chen, B. Peng, X. Wang, and H. Tang. Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds. *NDSS'12: Proceeding of the 19th Network and Distributed System Security Symposium*, 2012. [cited at p. 17]

[60] Yulia Cherdantseva and Jeremy Hilton. Information security and information assurance: Discussion about the meaning. *Organizational, Legal, and Technological Dimensions of Information System Administration*, 167, 2013. [cited at p. 2]

[61] Peter Claes, Denise K Liberton, Katleen Daniels, Kerri Matthes Rosana, Ellen E Quillen, Laurel N Pearson, Brian McEvoy, Marc Bauchet, Arslan A Zaidi, Wei Yao, et al. Modeling 3d facial shape from dna. *PLoS Genet*, 10(3):e1004224, 2014. [cited at p. 132]

[62] David Clayton. On inferring presence of an individual in a mixture: a bayesian approach. *Biostatistics*, 11(4):661–673, 2010. [cited at p. 154]

[63] Sara Colombo, Andri Rauch, Margalida Rotger, Jacques Fellay, Raquel Martinez, Christoph Fux, Christine Thurnheer, Huldrych F Günthard, David B Goldstein, Hansjakob Furrer, et al. The hcp5 single-nucleotide polymorphism: a simple screening tool for prediction of hypersensitivity reaction to abacavir. *The Journal of infectious diseases*, 198(6):864–867, 2008. [cited at p. 46]

[64] 1000 Genomes Project Consortium et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012. [cited at p. 9, 91, 97]

[65] International HapMap 3 Consortium et al. Integrating common and rare genetic variation in diverse human populations. *Nature*, 467(7311):52–58, 2010. [cited at p. 49, 91]

[66] Norsys Software Corp. Netica. `https://www.norsys.com/index.html`, 2017. Last Accessed: March 13, 2018. [cited at p. 144]

[67] Miroslav Cupak. Beacon network: A system for global genomic data sharing. Master's thesis, Masaryk University, 2016. [cited at p. 106, 108]

[68] Ivan Damgård, Martin Geisler, and Mikkel Krøigaard. Efficient and secure comparison for on-line auctions. *Information Security and Privacy*, 4586:416–430, 2007. [cited at p. 51]

[69] Ivan Damgård, Martin Geisler, and Mikkel Krøigaard. Homomorphic encryption and secure comparison. *Int. J. Appl. Cryptol.*, 1:22—-31, 2008. [cited at p. 38, 39]

[70] George Danezis and Emiliano De Cristofaro. Fast and private genomic testing for disease susceptibility. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. ACM, 2014. [cited at p. 155]

[71] Michael Dean, Mary Carrington, Cheryl Winkler, Gavin A Huttley, et al. Genetic restriction of hiv-1 infection and progression to aids by a deletion allele of the ckr5 structual gene. *Science*, 273(5283):1856, 1996. [cited at p. 46]

[72] Julia Di Iulio, Angela Ciuffi, Karen Fitzmaurice, Dermot Kelleher, Margalida Rotger, Jacques Fellay, Raquel Martinez, Sara Pulit, Hansjakob Furrer, Huldrych F Günthard, et al. Estimating the net contribution of interleukin-28b variation to spontaneous hepatitis c virus clearance. *Hepatology*, 53(5):1446–1454, 2011. [cited at p. 46]

[73] Julia di Iulio, Aurélie Fayet, Mona Arab-Alameddine, Margalida Rotger, Rubin Lubomirov, Matthias Cavassini, Hansjakob Furrer, Huldrych F Günthard, Sara Colombo, Chantal Csajka, et al. In vivo analysis of efavirenz metabolism in individuals with impaired cyp2a6 function. *Pharmacogenetics and genomics*, 19(4):300–309, 2009. [cited at p. 46]

[74] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008. [cited at p. 63]

[75] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006. [cited at p. 63, 84]

[76] Morris J Dworkin. Sp 800-38c. recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality. 2004. [cited at p. 47]

[77] Khaled El Emam, Elizabeth Jonker, Luk Arbuckle, and Bradley Malin. A systematic review of re-identification attacks on health data. *PloS one*, 6(12):e28071, 2011. [cited at p. 88]

[78] Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L. Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE Transactions on Information Forensics and Security*, 7(3):1053–1066, June 2012. [cited at p. 44]

[79] Yaniv Erlich and Arvind Narayanan. Routes for breaching and protecting genetic privacy. *Nature Reviews Genetics*, 15(6):409–421, 2014. [cited at p. 3, 60, 102, 129, 130, 132, 154]

[80] EU Parlament. The EU General Data Protection Regulation (GDPR). `http://www.eugdpr.org/`. Last Accessed: March 13, 2018. [cited at p. 56, 105]

[81] Facebook. `https://www.facebook.com/`, 2017. Last Accessed: March 13, 2018. [cited at p. 132]

[82] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012. [cited at p. 62, 67, 71]

[83] Jacques Fellay, Kevin V Shianna, Dongliang Ge, Sara Colombo, Bruno Ledergerber, Mike Weale, Kunlin Zhang, Curtis Gumbs, Antonella Castagna, Andrea Cossarizza, et al. A whole-genome association study of major determinants for host control of hiv-1. *science*, 317(5840):944–947, 2007. [cited at p. 46]

[84] Jacques Fellay, Alexander J Thompson, Dongliang Ge, Curtis E Gumbs, Thomas J Urban, Kevin V Shianna, Latasha D Little, Ping Qiu, Arthur H Bertelsen, Mark Watson, et al. Itpa gene variants protect against anaemia in patients treated for chronic hepatitis c. *Nature*, 464(7287):405–408, 2010. [cited at p. 46]

[85] S. E. Fienberg, A. Slavkovic, and C. Uhler. Privacy preserving GWAS data sharing. *Proceedings of the IEEE ICDMW '11*, Dec. 2011. [cited at p. 17]

[86] S.E. Fienberg, A. Slavkovic, and C. Uhler. Privacy preserving GWAS data sharing. In *ICDM*, pages 628–635, 2011. [cited at p. 129]

[87] Global Alliance for Genomics and Health. The beacon project. `https://beacon-network.org/#/`. Last Accessed: March 13, 2018. [cited at p. 66, 126]

[88] Matthew Fredriksen, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd USENIX Security Symposium*, 2014. [cited at p. 130, 155]

[89] David Froelicher, Patricia Egger, João Sá Sousa, Jean Louis Raisaro, Zhicong Huang, Christian Mouchet, Bryan Ford, and Jean-Pierre Hubaux. UnLynx: A decentralized system for privacy-conscious data sharing. In *Proceedings on Privacy Enhancing Technologies*, volume 4, pages 152–170, 2017. [cited at p. 106, 110, 111, 119, 120]

[90] Dongliang Ge, Jacques Fellay, Alexander J Thompson, Jason S Simon, Kevin V Shianna, Thomas J Urban, Erin L Heinzen, Ping Qiu, Arthur H Bertelsen, Andrew J Muir, et al. Genetic variation in il28b predicts hepatitis c treatment-induced viral clearance. *Nature*, 461(7262):399–401, 2009. [cited at p. 46]

[91] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. [cited at p. 13]

[92] Global Alliance for Genomics and Health. `https://genomicsandhealth.org`. Last Accessed: March 13, 2018. [cited at p. 56, 87]

[93] Global Alliance for Genomics and Health. GA4GH Privacy and Security Policy. `https://www.ga4gh.org/docs/ga4ghtoolkit/data-security/Privacy-and-Security-Policy.pdf`, 2015. Last Accessed: March 13, 2018. [cited at p. 87]

[94] Oded Goldreich and Rafail Ostrovsky. Software protection and simulation on oblivious RAMs. *J. ACM*, 43(3):431–473, May 1996. [cited at p. 22]

[95] Norman Göttert, Thomas Feller, Michael Schneider, Johannes Buchmann, and Sorin Huss. On the design of hardware building blocks for modern lattice-based encryption schemes. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 512–529. Springer, 2012. [cited at p. 79]

[96] Dov Greenbaum, Andrea Sboner, Xinmeng Jasmine Mu, and Mark Gerstein. Genomics and privacy: Implications of the new reality of closed data for the field. *PLoS Comput Biol*, 7(12), 12 2011. [cited at p. 88, 129]

[97] Monia Guidi, Giuseppe Foletti, Paul McLaren, Matthias Cavassini, Andri Rauch, Philip E Tarr, Olivier Lamy, Alice Panchaud, Amalio Telenti, Chantal Csajka, et al. Vitamin d time profile based on the contribution of non-genetic and genetic factors in hiv-infected individuals of european ancestry. *Antiviral therapy*, 20(3):261–269, 2014. [cited at p. 36, 46]

[98] Huldrych F Gunthard, Judith A Aberg, Joseph J Eron, Jennifer F Hoy, Amalio Telenti, Constance A Benson, David M Burger, Pedro Cahn, Joel E Gallant, Marshall J Glesby, et al. Antiretroviral treatment of adult hiv infection: 2014 recommendations of the international antiviral society–usa panel. *Jama*, 312(4):410–425, 2014. [cited at p. 36, 47]

[99] Melissa Gymrek, Amy L McGuire, David Golan, Eran Halperin, and Yaniv Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013. [cited at p. 3, 17, 60, 65, 88, 129, 132, 154]

[100] David W Haas, Heather J Ribaudo, Richard B Kim, Camlin Tierney, Grant R Wilkinson, Roy M Gulick, David B Clifford, Todd Hulgan, Catia Marzolini, and Edward P Acosta. Pharmacogenetics of efavirenz and central nervous system side effects: an adult aids clinical trials group study. *Aids*, 18(18):2391–2400, 2004. [cited at p. 46]

[101] Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 61–70. IEEE, 2010. [cited at p. 84]

[102] Arif Harmanci and Mark Gerstein. Quantification of private information leakage from phenotype-genotype data: linking attacks. *Nature methods*, 13(3):251–256, 2016. [cited at p. 132]

[103] Seth Hetherington, Arlene R Hughes, Michael Mosteller, Denise Shortino, Katherine L Baker, William Spreen, Eric Lai, Kirstie Davies, Abigail Handley, David J Dow, et al. Genetic variations in hla-b region and hypersensitivity reactions to abacavir. *The Lancet*, 359(9312):1121–1122, 2002. [cited at p. 46]

[104] Lucia A Hindorff, Heather A Junkins, PN Hall, JP Mehta, and TA Manolio. A catalog of published genome-wide association studies. `http://www.genome.gov/gwastudies`, 2011. Last Accessed: March 13, 2018. [cited at p. 130, 132, 144]

[105] HIPAA News. 480,000 Patients Notified of Radiology Regional Center PHI Exposure. `http://www.hipaajournal.com/480000-patients-notified-of-radiology-regional-center-phi-exposure-8322/`. [cited at p. 129]

[106] Eran Hodis, Ian R Watson, Gregory V Kryukov, Stefan T Arold, Marcin Imielinski, Jean-Philippe Theurillat, Elizabeth Nickerson, Daniel Auclair, Liren Li, Chelsea Place, et al. A landscape of driver mutations in melanoma. *Cell*, 150(2):251–263, 2012. [cited at p. 120]

[107] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 4(8), Aug. 2008. [cited at p. 3, 60, 65, 88, 90, 129, 130, 132, 136, 144, 154]

[108] Stefanie Hostettler and Esther Kraft. 36 175 médecins en exercice. *Bulletin Des Médecins Suisses*, 98(13):394–400, 2017. [cited at p. 52]

[109] Zhicong Huang, Erman Ayday, Jacques Fellay, Jean-Pierre Hubaux, and Ari Juels. Genoguard: Protecting genomic data against brute-force attacks. In *IEEE Symposium on Security and Privacy*, pages 447–462. IEEE Computer Society, 2015. [cited at p. 155]

[110] Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. Addressing the concerns of the lacks family: Quantification of kin genomic privacy. In *ACM Conference on Computer and Communications Security, (CCS)*, 2013. [cited at p. 130, 132, 149]

[111] Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. Reconciling utility with privacy in genomics. In *13th Workshop on Privacy in the Electronic Society (WPES14)*, pages 11–20. ACM, 2014. [cited at p. 154]

[112] Mathias Humbert, Kévin Huguenin, Joachim Hugonot, Erman Ayday, and Jean-Pierre Hubaux. De-anonymizing genomic databases using phenotypic traits. *Proceedings on Privacy Enhancing Technologies*, 2015(2):99–114, 2015. [cited at p. 3, 60, 65, 132, 154]

[113] Kevin B Jacobs, Meredith Yeager, Sholom Wacholder, David Craig, Peter Kraft, David J Hunter, Justin Paschal, Teri A Manolio, Margaret Tucker, Robert N Hoover, et al. A new statistic and its power to infer membership in a genome-wide association study using genotype frequencies. *Nature Genetics*, 41(11):1253–1257, 2009. [cited at p. 136, 154]

[114] Somesh Jha, Louis Kruger, and Vitaly Shmatikov. Towards practical privacy for genomic computation. *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 216–230, 2008. [cited at p. 17]

[115] Xiaoming Jia, Buhm Han, Suna Onengut-Gumuscu, Wei-Min Chen, Patrick J Concannon, Stephen S Rich, Soumya Raychaudhuri, and Paul IW de Bakker. Imputing amino acid polymorphisms in human leukocyte antigens. *PloS one*, 8(6):e64683, 2013. [cited at p. 47]

[116] Aaron Johnson and Vitaly Shmatikov. Privacy-preserving data exploration in genome-wide association studies. In *KDD*, pages 1079–1087, 2013. [cited at p. 129, 154]

[117] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. Dynamic searchable symmetric encryption. *Proceedings of the 2012 ACM Conference on Computer and Communications Security - CCS*, 2012. [cited at p. 22]

[118] Liina Kamm, Dan Bogdanov, Sven Laur, and Jaak Vilo. A new way to protect privacy in large-scale genome-wide association studies. *Bioinformatics*, 29(7):886–893, 2013. [cited at p. 61]

[119] M. Kantarcioglu, Wei Jiang, Ying Liu, and B. Malin. A cryptographic approach to securely share and query genomic sequences. *Information Technology in Biomedicine, IEEE Transactions on*, 12(5):606–617, 2008. [cited at p. 61]

[120] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014. [cited at p. 12]

[121] Krishnaram Kenthapadi, Nina Mishra, and Kobbi Nissim. Simulatable auditing. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 118–127. ACM, 2005. [cited at p. 130, 139]

[122] Krishnaram Kenthapadi, Nina Mishra, and Kobbi Nissim. Denials leak information: Simulatable auditing. *Journal of Computer and System Sciences*, 79(8):1322–1340, 2013. [cited at p. 139]

[123] Hyeoneui Kim, Elizabeth Bell, Jihoon Kim, Amy Sitapati, Joe Ramsdell, Claudiu Farcas, Dexter Friedman, Stephanie Feudjio Feupe, and Lucila Ohno-Machado. iconcur: informed consent for clinical data and bio-sample use for research. *Journal of the American Medical Informatics Association*, 24(2):380–387, 2016. [cited at p. 130, 138]

[124] Rex B Kline. *Principles and practice of structural equation modeling*. Guilford publications, 2015. [cited at p. 52]

[125] Bartha Maria Knoppers. Framework for responsible sharing of genomic and health-related data. *The HUGO journal*, 8(1):3, 2014. [cited at p. 87]

[126] Roman Kosoy, Rami Nassir, Chao Tian, Phoebe A White, Lesley M Butler, Gabriel Silva, Rick Kittles, Marta E Alarcon-Riquelme, Peter K Gregersen, John W Belmont, et al. Ancestry informative marker sets for determining continental origin and admixture proportions in common populations in america. *Human mutation*, 30(1):69–78, 2009. [cited at p. 47]

[127] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988. [cited at p. 135]

[128] Na Li and Matthew Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165(4):2213–2233, 2003. [cited at p. 154]

[129] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007. [cited at p. 73, 129]

[130] Christoph Lippert, Riccardo Sabatini, M Cyrus Maher, Eun Yong Kang, Seunghak Lee, Okan Arikan, Alena Harley, Axel Bernal, Peter Garst, Victor Lavrenko, et al. Identification of individuals by trait prediction using whole-genome sequencing data. *Proceedings of the National Academy of Sciences*, 114(38):10166–10171, 2017. [cited at p. 3]

[131] Rubin Lubomirov, Sara Colombo, Julia di Iulio, Bruno Ledergerber, Raquel Martinez, Matthias Cavassini, Bernard Hirschel, Enos Bernasconi, Luigia Elzi, Pietro Vernazza, et al. Association of pharmacogenetic markers with premature discontinuation of first-line anti-hiv therapy: an observational cohort study. *Journal of infectious diseases*, 203(2):246–257, 2011. [cited at p. 36, 46]

[132] Rubin Lubomirov, Chantal Csajka, and Amalio Telenti. Adme pathway approach for pharmacogenetic studies of anti-hiv therapy. *Future Medicine*, pages 623–633, 2007. [cited at p. 47]

[133] Rubin Lubomirov, Julia di Iulio, Aurélie Fayet, Sara Colombo, Raquel Martinez, Catia Marzolini, Hansjakob Furrer, Pietro Vernazza, Alexandra Calmy, Matthias Cavassini, et al. Adme pharmacogenetics: investigation of the pharmacokinetics of the antiretroviral agent lopinavir coformulated with ritonavir. *Pharmacogenetics and genomics*, 20(4):217–230, 2010. [cited at p. 46]

[134] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007. [cited at p. 73, 129]

[135] Bradley Malin and Latanya Sweeney. How (not) to protect genomic data privacy in a distributed network: Using trail re-identification to evaluate and design anonymity protection systems. *Journal of Biomedical Informatics*, 37(3):179–192, 2004. [cited at p. 60, 65]

[136] S Mallal, D Nolan, C Witt, G Masel, AM Martin, C Moore, D Sayer, A Castley, C Mamotte, D Maxwell, et al. Association between presence of hla-b* 5701, hla-dr7, and hla-dq3 and hypersensitivity to hiv-1 reverse-transcriptase inhibitor abacavir. *The Lancet*, 359(9308):727–732, 2002. [cited at p. 36, 46]

[137] Ani Manichaikul, Josyf C Mychaleckyj, Stephen S Rich, Kathy Daly, Michèle Sale, and Wei-Min Chen. Robust relationship inference in genome-wide association studies. *Bioinformatics*, 26(22):2867–2873, 2010. [cited at p. 136, 137, 144, 154]

[138] Jonathan Marchini, Bryan Howie, Simon Myers, Gil McVean, and Peter Donnelly. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature genetics*, 39(7):906–913, 2007. [cited at p. 134]

[139] Ruth Marcus, Peritz Eric, and K Ruben Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3):655–660, 1976. [cited at p. 137]

[140] Paul J McLaren and Mary Carrington. The impact of host genetic variation on infection with hiv-1. *Nature immunology*, 16(6):577–583, 2015. [cited at p. 36]

[141] Paul J. McLaren, Jean Louis Raisaro, Manel Aouri, Margalida Rotger, Erman Ayday, István Bartha, Maria B. Delgado, Yannick Vallet, Huldrych F. Günthard, Matthias Cavassini, Hansjakob Furrer, Thanh Doco-Lecompte, Catia Marzolini, Patrick Schmid, Caroline Di Benedetto, Laurent A. Decosterd, Jacques Fellay, Jean-Pierre Hubaux, Amalio Telenti, and the Swiss HIV Cohort Study. Privacy-preserving genomic testing in the clinic: a model using HIV treatment. *Genetics in Medicine*, 18(8):814–822, aug 2016. [cited at p. 7, 47, 155]

[142] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007. [cited at p. 84]

[143] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 2014(239):2, 2014. [cited at p. 121]

[144] Clint Mizzi, Brock Peters, Christina Mitropoulou, Konstantinos Mitropoulos, Theodora Katsila, Misha R Agarwal, Ron HN Van Schaik, Radoje Drmanac, Joseph Borg, and George P Patrinos. Personalized pharmacogenomics profiling using whole-genome sequencing. *Pharmacogenomics*, 15(9):1223–1234, 2014. [cited at p. 56]

[145] Shawn N Murphy, Vivian Gainer, Michael Mendis, Susanne Churchill, and Isaac Kohane. Strategies for maintaining patient privacy in i2b2. *Journal of the American Medical Informatics Association*, 18(Supplement 1):i103–i108, 2011. [cited at p. 60]

[146] Shawn N Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C Chueh, Susanne Churchill, and Isaac Kohane. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association*, 17(2):124–130, 2010. [cited at p. 60, 106, 110, 111, 120]

[147] Mina Namazi, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Dynamic privacy-preserving genomic susceptibility testing. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 45–50. ACM, 2016. [cited at p. 69]

[148] Muhammad Naveed, Erman Ayday, Ellen W Clayton, Jacques Fellay, Carl A Gunter, Jean-Pierre Hubaux, Bradley A Malin, and XiaoFeng Wang. Privacy in the genomic era. *ACM Computing Surveys (CSUR)*, 48(1):6, 2015. [cited at p. 3, 60, 129, 154, 155]

[149] Muhammad Naveed, Seny Kamara, and Charles V. Wright. Inference attacks on property-preserving encrypted databases. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 644–655, New York, NY, USA, 2015. ACM. [cited at p. 117]

[150] C Andrew Neff. Verifiable mixing (shuffling) of elgamal pairs. *VHTi Technical Document, VoteHere, Inc*, 2003. [cited at p. 112]

[151] Anna Nowogrodzki. Spiking genomic databases with misinformation could protect patient privacy. *Nature News*, 2016. [cited at p. 130]

[152] OpenSNP. `https://opensnp.org/`, 2017. Last Accessed: March 13, 2018. [cited at p. 3, 132]

[153] PatientsLikeMe. `https://www.patientslikeme.com`, 2017. Last Accessed: March 13, 2018. [cited at p. 155]

[154] Anthony A Philippakis, Danielle R Azzariti, Sergi Beltran, Anthony J Brookes, Catherine A Brownstein, Michael Brudno, Han G Brunner, Orion J Buske, Knox Carey, Cassie Doll, et al. The matchmaker exchange: a platform for rare disease gene discovery. *Human mutation*, 36(10):915–921, 2015. [cited at p. 108]

[155] Raluca Ada Popa, Frank H. Li, and Nickolai Zeldovich. An ideal-security protocol for order-preserving encoding. *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, 2013. [cited at p. 13, 33]

[156] Raluca Ada Popa, Catherine Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 85–100. ACM, 2011. [cited at p. 13, 32, 33]

[157] PostgreSQL Global Development Group. PostgreSQL 10. `https://www.postgresql.org/`. Last Accessed: March 13, 2018. [cited at p. 121]

[158] Premier Healthcare. Notice to Our Patients Regarding a Security Incident. `http://www.premierhealthcare.org/incident-2016-03.html`. [cited at p. 129]

[159] Alkes L. Price, Nick J. Patterson, Robert M. Plenge, Michael E. Weinblatt, Nancy A. Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, pages 904–909, 2006. [cited at p. 38, 43]

[160] Jean Louis Raisaro, Erman Ayday, and Jean-Pierre Hubaux. Patient privacy in the genomic era. *Praxis*, 103(10):579–86, 2014. [cited at p. 1]

[161] Jean Louis Raisaro, Gwangbae Choi, Sylvain Pradervand, Raphael Colsenet, Nathalie Jacquemont, Nicolas Rosat, Vincent Mooser, and Jean-pierre Hubaux. *Selected for publication in IEEE/ACM Transactions in Computational Biology and Bioinformatics*, pages 1–17, 2017. [cited at p. 7]

[162] Jean-Louis Raisaro, Paul J McLaren, Jacques Fellay, Matthias Cavassini, Catherine Klersy, and Jean-Pierre Hubaux. Are privacy-enhancing technologies for genomic data ready for the clinic? a survey of medical experts of the swiss hiv cohort study. *Journal of biomedical informatics*, 2018. [cited at p. 7]

[163] Jean Louis Raisaro, Florian Tramèr, Zhanglong Ji, Diyue Bu, Yongan Zhao, Knox Carey, David Lloyd, Heidi Sofia, Dixie Baker, Paul Flicek, Suyash Shringarpure, Carlos Bustamante, Shuang Wang, Xiaoqian Jiang, Lucila Ohno-Machado, Haixu Tang, XiaoFeng Wang, and Jean-Pierre Hubaux. Addressing Beacon re-identification attacks: quantification and mitigation of privacy risks. *Journal of the American Medical Informatics Association*, (0):1–8, 2017. [cited at p. 7]

[164] Jean Louis Raisaro, Carmela Troncoso, Mathias Humbert, Zoltan Kutalik, Amalio Telenti, and Jean-Pierre Hubaux. Genoshare: Supporting privacy-informed decisions for sharing exact genomic data. Technical report, EPFL infoscience, 2017. `https://infoscience.epfl.ch/record/225639`. [cited at p. 7]

[165] Jean Louis Raisaro, Juan Ramón Troncoso-Pastoriza, Mickaël Misbach, João Sá Sousa, Sylvain Pradervand, Edoardo Missiaglia, Olivier Michielin, Bryan Ford, and Jean-Pierre Hubaux. MedCo: Enabling Privacy-Conscious Exploration of Distributed Clinical and Genomic Data. In *4th International Workshop on Genome Privacy and Security (GenoPri'17)*, 2017. [cited at p. 7]

[166] Andri Rauch, D Nolan, A Martin, E McKinnon, C Almeida, and Simon Mallal. Prospective genetic screening decreases the incidence of abacavir hypersensitivity reactions in the western australian hiv cohort study. *Clinical infectious diseases*, 43(1):99–102, 2006. [cited at p. 36]

[167] John H. Relethford. Hardy-Weinberg equilibrium. In *Human Population Genetics*, pages 23–48. John Wiley & Sons, Inc., 2012. [cited at p. 42]

[168] M Rotger, H Tegude, S Colombo, M Cavassini, Hansjakob Furrer, L Decosterd, J Blievernicht, T Saussele, HF Günthard, M Schwab, et al. Predictive value of known and novel alleles of cyp2b6 for efavirenz plasma concentrations in hiv-infected individuals. *Clinical pharmacology & therapeutics*, 81(4):557–566, 2007. [cited at p. 46]

[169] Margalida Rotger, Cornelia Bayard, Patrick Taffé, Raquel Martinez, Matthias Cavassini, Enos Bernasconi, Manuel Battegay, Bernard Hirschel, Hansjakob Furrer, Andrea Witteck, et al. Contribution of genome-wide significant single nucleotide polymorphisms and antiretroviral therapy to dyslipidemia in hiv-infected individuals-a longitudinal study. *Circulation: Cardiovascular Genetics*, pages CIRCGENETICS–109, 2009. [cited at p. 46]

[170] Margalida Rotger and *et al.* Contribution of genetic background, traditional risk factors and HIV-related factors to coronary artery disease events in HIV-positive persons. *Clinical Infectious Diseases*, mar 2013. [cited at p. 45]

[171] Margalida Rotger, Tracy R Glass, Thomas Junier, Jens Lundgren, James D Neaton, Estella S Poloni, Angélique B Van't Wout, Rubin Lubomirov, Sara Colombo, Raquel Martinez, et al. Contribution of genetic background, traditional risk factors, and hiv-related factors to coronary artery disease events in hiv-positive persons. *Clinical infectious diseases*, 57(1):112–121, 2013. [cited at p. 36, 46, 55]

[172] Margalida Rotger, Thomas Gsponer, Raquel Martinez, Patrick Taffé, Luigia Elzi, Pietro Vernazza, Matthias Cavassini, Enos Bernasconi, Bernard Hirschel, Hansjakob Furrer, et al. Impact of single nucleotide polymorphisms and of clinical risk factors on new-onset diabetes mellitus in hiv-infected individuals. *Clinical infectious diseases*, 51(9):1090–1098, 2010. [cited at p. 36]

[173] Margalida Rotger, Patrick Taffé, Gabriela Bleiber, Huldrych F Günthard, Hansjakob Furrer, Pietro Vernazza, Henning Drechsler, Enos Bernasconi, Martin Rickenbach, and Amalio Telenti. Gilbert syndrome and the development of antiretroviral therapy–associated hyperbilirubinemia. *The Journal of infectious diseases*, 192(8):1381–1386, 2005. [cited at p. 46]

[174] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 765–774. ACM, 2010. [cited at p. 84]

[175] Sahel Samani, Zhicong Huang, Erman Ayday, Mark Elliot, Jacques Fellay, Jean-Pierre Hubaux, and Zoltán Kutalik. Quantifying genomic privacy via inference attack with high-order snv correlations. In *2nd International Workshop on Genome Privacy and Security (in conjunction with IEEE S&P; 2015)*, 2015. [cited at p. 130, 134, 154]

[176] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. *Nature Genetics*, 41(9):965–967, 2009. [cited at p. 88, 136, 154]

[177] Jennifer Schlesinger. Dark web is fertile ground for stolen medical records. `http://www.cnbc.com/2016/03/10/dark-web-is-fertile-ground-for-stolen-medical-records.html`. Last Accessed: March 13, 2018. [cited at p. 65]

[178] Franziska Schoeni-Affolter, Bruno Ledergerber, Martin Rickenbach, Christoph Rudin, Huldrych F Günthard, Amalio Telenti, Hansjakob Furrer, Sabine Yerly, and Patrick Francioli. Cohort profile: the swiss hiv cohort study. *International journal of epidemiology*, 39(5):1179–1189, 2009. [cited at p. 46]

[179] Heribert Schunkert, Inke R König, Sekar Kathiresan, Muredach P Reilly, Themistocles L Assimes, Hilma Holm, Michael Preuss, Alexandre FR Stewart, Maja Barbalic, Christian Gieger, et al. Large-scale association analysis identifies 13 new susceptibility loci for coronary artery disease. *Nature genetics*, 43(4):333–338, 2011. [cited at p. 46]

[180] Joe V Selby, Anne C Beal, and Lori Frank. The patient-centered outcomes research institute (pcori) national priorities for research and initial research agenda. *Jama*, 307(15):1583–1584, 2012. [cited at p. 105, 108]

[181] Suyash S Shringarpure and Carlos D Bustamante. Privacy risks from genomic data-sharing beacons. *The American Journal of Human Genetics*, 97(5):631–646, 2015. [cited at p. 60, 65, 88, 90, 98, 132, 136]

[182] Sean Simmons and Bonnie Berger. Realizing privacy preserving genome-wide association studies. *Bioinformatics*, 32(9):1293–1300, 2016. [cited at p. 154]

[183] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pages 44–55. IEEE, 2000. [cited at p. 22]

[184] Elizabeth K Speliotes, Cristen J Willer, Sonja I Berndt, Keri L Monda, Gudmar Thorleifsson, Anne U Jackson, Hana Lango Allen, Cecilia M Lindgren, Jian'an Luan, Reedik Mägi, et al. Association analyses of 249,796 individuals reveal 18 new loci associated with body mass index. *Nature genetics*, 42(11):937–948, 2010. [cited at p. 37]

[185] NIST-FIPS Standard. Announcing the advanced encryption standard (aes). *Federal Information Processing Standards Publication*, 197:1–51, 2001. [cited at p. 47]

[186] Emil Stefanov, Elaine Shi, and Dawn Song. Towards practical oblivious RAM. *NDSS'12: Proceeding of the 19th Network and Distributed System Security Symposium*, 2012. [cited at p. 22]

[187] Genome-Wide Association Studies. `http://www.genome.gov/20019523`, 2017. Last Accessed: March 13, 2018. [cited at p. 130, 132]

[188] The International HIV Controllers Study et al. The major genetic determinants of hiv-1 control affect hla class i peptide presentation. *Science (New York, NY)*, 330(6010):1551, 2010. [cited at p. 46]

[189] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002. [cited at p. 73, 129]

[190] Swiss Academies of Arts and Sciences. Swiss Personalized Health Network. `http://www.samw.ch/en/Projects/SPHN.html`. Last Accessed: March 13, 2018. [cited at p. 105]

[191] Amalio Telenti, Erman Ayday, and Jean Pierre Hubaux. On genomics, kin, and privacy. *F1000Research*, 3, 2014. [cited at p. 35]

[192] Jacob A Tennessen, Abigail W Bigham, Timothy D O'Connor, Wenqing Fu, Eimear E Kenny, Simon Gravel, Sean McGee, Ron Do, Xiaoming Liu, Goo Jun, et al. Evolution and functional impact of rare coding variation from deep sequencing of human exomes. *science*, 337(6090):64–69, 2012. [cited at p. 88]

[193] Sharon F Terry, Robert Shelton, Greg Biggers, Dixie Baker, and Kelly Edwards. The haystack is made of needles. *Genetic testing and molecular biomarkers*, 17(3):175–177, 2013. [cited at p. 88]

[194] The 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491:56–65, 2012. [cited at p. 43, 89, 130, 131, 132, 144]

[195] The Global Alliance for Genomics and Health. A federated ecosystem for sharing genomic, clinical data. *Science*, 352(6291):1278–1280, 2016. [cited at p. 105]

[196] The International HapMap 3 Consortium. Integrating common and rare genetic variation in diverse human populations. *Nature*, 467:52–58, 2010. [cited at p. 43]

[197] The Wall Street Journal. Anthem: Hacked Database Included 78.8 Million People. `http://www.wsj.com/articles/anthem-hacked-database-included-78-8-million-people-1424807364`. [cited at p. 129]

[198] David L Thomas, Chloe L Thio, Maureen P Martin, Ying Qi, Dongliang Ge, Colm O'hUigin, Judith Kidd, Kenneth Kidd, Salim I Khakoo, Graeme Alexander, et al. Genetic variation in il28b and spontaneous clearance of hepatitis c virus. *Nature*, 461(7265):798–801, 2009. [cited at p. 46]

[199] Timothy Thornton, Hua Tang, Thomas J. Hoffmann, Heather M. Ochs-Balcom, Bette Caan, and Neil Risch. Estimating kinship in admixed populations. *The American Journal of Human Genetics*, 91(1):122–138, 2012. [cited at p. 136, 154]

[200] Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, and Mehmet Celik. Privacy preserving error resilient DNA searching through oblivious automata. *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007. [cited at p. 17]

[201] Caroline Uhler, Aleksandra Slavkovic, and Stephen E. Fienberg. Privacy-preserving data sharing for genome-wide association studies. *Journal of Privacy and Confidentiality*, 5(1), 2013. [cited at p. 154]

[202] U.S. Department of Health & Human Services. The health insurance portability and accountability act (hipaa). `https://www.hhs.gov/hipaa/index.html`. Last Accessed: March 13, 2018. [cited at p. 56]

[203] U.S. Department of Health and Human Services . Breach portal: Notice to the secretary of hhs breach of unsecured protected health information. `https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf`. Last Accessed: March 13, 2018. [cited at p. 105, 129]

[204] GW Van Blarkom, JJ Borking, and JGE Olk. Handbook of privacy and privacy-enhancing technologies. *Privacy Incorporated Software Agent (PISA) Consortium, The Hague*, 2003. [cited at p. 3]

[205] T. Veugen. Improving the DGK comparison protocol. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 49–54, 2012. [cited at p. 44]

[206] Staal A Vinterbo, Anand D Sarwate, and Aziz A Boxwala. Protecting count queries in study design. *Journal of the American Medical Informatics Association*, 19(5):750–757, 2012. [cited at p. 84]

[207] Peter M Visscher and William G Hill. The limits of individual identification from sample allele frequencies: theory and statistical analysis. *PLoS Genetics*, 5(10):e1000628, 2009. [cited at p. 136, 154]

[208] Isabel Wagner. Evaluating the strength of genomic privacy metrics. *ACM Transactions on Privacy and Security (TOPS)*, 20(1):2, 2017. [cited at p. 130, 138]

[209] Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. Learning your identity and disease from research papers: Information leaks in genome wide association study. *Proceedings of ACM CCS '09*, pages 534–544, 2009. [cited at p. 17]

[210] Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. Learning your identity and disease from research papers: Information leaks in genome wide association study. In *ACM Conference on Computer and Communications Security, (CCS)*, pages 534–544, 2009. [cited at p. 60, 65, 136, 154]

[211] Rui Wang, XiaoFeng Wang, Zhou Li, Haixu Tang, Michael K. Reiter, and Zheng Dong. Privacy-preserving genomic computation through program specialization. In *ACM Conference on Computer and Communications Security, (CCS)*, pages 338–347, 2009. [cited at p. 155]

[212] Xiao Shaun Wang, Yan Huang, Yongan Zhao, Haixu Tang, XiaoFeng Wang, and Diyue Bu. Efficient genome-wide, privacy-preserving similar patient query based on private edit distance. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 492–503. ACM, 2015. [cited at p. 61]

[213] Griffin M Weber, Shawn N Murphy, Andrew J McMurry, Douglas MacFadden, Daniel J Nigrin, Susanne Churchill, and Isaac S Kohane. The shared health research information network (shrine): a prototype federated query tool for clinical data repositories. *Journal of the American Medical Informatics Association*, 16(5):624–630, 2009. [cited at p. 106, 109, 110, 120, 126]

[214] David I Wolinsky, Henry Corrigan-Gibbs, Bryan Ford, and Aaron Johnson. Scalable anonymous group communication in the anytrust model. Technical report, NAVAL RESEARCH LAB WASHINGTON DC, 2012. [cited at p. 111]

[215] Xiaokui Xiao and Yufei Tao. Dynamic anonymization: accurate statistical analysis with privacy preservation. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 107–120. ACM, 2008. [cited at p. 129]

[216] Wei Xie, Murat Kantarcioglu, William S Bush, Dana Crawford, Joshua C Denny, Raymond Heatherly, and Bradley A Malin. SecureMA: protecting participant privacy in genetic association meta-analysis. *Bioinformatics*, 30(23):3334–3341, 2014. [cited at p. 61]

[217] H Yang, DA Kircher, KH Kim, AH Grossmann, MW VanBrocklin, SL Holmen, and JP Robinson. Activated mek cooperates with cdkn2a and pten loss to promote the development and maintenance of melanoma. *Oncogene*, 36(27):3842–3851, 2017. [cited at p. 120]

[218] Fei Yu, Stephen E Fienberg, Aleksandra B Slavković, and Caroline Uhler. Scalable privacy-preserving data sharing methodology for genome-wide association studies. *Journal of biomedical informatics*, 50:133–141, 2014. [cited at p. 129]

[219] Fei Yu, Michal Rybar, Caroline Uhler, and Stephen E Fienberg. Differentially-private logistic regression for detecting multiple-snp association in gwas databases. In *International Conference on Privacy in Statistical Databases*, pages 170–184. Springer, 2014. [cited at p. 154]

[220] Xiao-yong Zhou, Bo Peng, Yong Fuga Li, Yangyi Chen, Haixu Tang, and XiaoFeng Wang. To release or not to release: Evaluating information leaks in aggregate human-genome data. In *ESORICS*, volume 11, pages 607–627. Springer, 2011. [cited at p. 17, 60, 65, 136, 154]

Index

JEAN LOUIS RAISARO

EPFL IC ISC LCA1, Station 14
1015 Lausanne, Switzerland
jeanlouis.raisaro@gmail.com
www.linkedin.com/in/jean-louis-raisaro
+41 (0)78 948 9861

23.11.1987
Married
Italian, French

PROFILE

- ✓ Expert in Medical/Genomic Data Privacy and Security, Applied Cryptography, Databases, Medical Informatics, Machine Learning and Blockchain technology
- ✓ Excellent communicator (English: full professional proficiency; Italian/French: Native speaker)
- ✓ Excellent team worker (efficient, responsible and capable to work under pressure)
- ✓ Good programming skills (Java, Go, Python, Matlab/Octave, Bash, SQL)
- ✓ Experience in leading and managing small teams of graduate/undergraduate students

WORK EXPERIENCE

École Polytechnique Fédérale de Lausanne (EPFL), Switzerland **09/2012 – 03/2018**

Research and Teaching Assistant at Laboratory for Communications and Applications

Teaching Assistant for: Privacy Protection, Information Computation Communication, Introduction to Programming, Practice of Object Oriented Programming, C Programming.

Harvard University, Boston, MA, United States **06/2017 – 09/2017**

Internship at Department of Biomedical Informatics, Harvard Medical School

Project: "System for secure sharing of i2b2 aggregate-level data in the cloud."
Host: Prof. Shawn Murphy

Lausanne University Hospital (CHUV), Switzerland **01/2012 – 06/2012**

Privacy and Security Consultant at Direction des Systèmes d'Information (DSI)

Project: "Privacy and security risk assessment of the clinical research data warehouse."

École Polytechnique Fédérale de Lausanne (EPFL), Switzerland **02/2012 – 09/2012**

Internship at Laboratory for Communications and Applications

Project: "Privacy-preserving genetic tests in the cloud."
Host: Prof. Jean-Pierre Hubaux

Aalborg University, Aalborg, Denmark **05/2011 – 09/2011**

Internship at Department of Biomedical Engineering and Informatics

Project: "Design and implementation of intelligent systems for Medical Terminology SNOMED CT."
Host: Prof. Stig K. Andersen

EDUCATION

École Polytechnique Fédérale de Lausanne (EPFL), Switzerland **09/2012 – 03/2018**

PhD in Computer, Communication and Information Sciences – Data Privacy and Security

Ph.D. Thesis: "Privacy-Enhancing Technologies for Medical and Genomic Data: From Theory to Practice"
Advisor: Prof. Jean-Pierre Hubaux (EPFL)

Università Degli Studi di Pavia, Pavia, Italy **09/2009 – 02/2012**

MSc in Biomedical Engineering and Medical Informatics

M.Sc. Thesis: "An Automatic SNOMED CT Encoder for Clinical Free-Text." (GPA=3.96/4)
Advisors: Prof. Silvana Quaglini and Prof. Riccardo Bellazzi

Università Degli Studi di Pavia, Pavia, Italy **09/2006 – 09/2009**

BSc in Biomedical Engineering

BSc Thesis: "A vocal system for the intelligent management of heart failure patients." (GPA=3.71/4)
Advisors: Prof. Mario Stefanelli and Prof. Silvana Quaglini

MAIN PROJECTS

Commission for Technology and Innovation (CTI) Project **03/2016 – 03/2018**
Title: "Privacy-Preserving Personalized Medicine Over Distributed Databases"
Academic Partner: Laboratory for Communications and Applications (EPFL)
Industrial partner: Sophia Genetics SA
Total budget: CHF 800K

Contrat de Mandat de l'Etat de Vaud **01/2014 – 03/2018**
Title: "Protection of Genomic and Medical Data of Lausanne Institutional Biobank (BIL)"
Partners: CHUV and Laboratory for Communications and Applications (EPFL)
Main achievements:

- Development of a prototype of a privacy-preserving solution for personalized medicine based on homomorphic encryption (in Java)
- Deployment of a secure i2b2 plugin for the exploration of genetic cohorts based on homomorphic encryption and differential privacy (in Java and C++)
- Development of a system for privacy-preserving sharing of distributed medical and genomic data
- 4 scientific publications, 1 patent

Total budget: CHF 444K

Swiss HIV Cohort Study (SHCS) Project **01/2013 – 12/2013**
Title: "Data protection in the SHCS – a pilot study using pharmacogenetics applications"
Partners: SHCS and Laboratory for Communications and Applications (EPFL)
Main achievements:

- System (back-end and front-end) for pharmacogenetics tests on encrypted genomes (deployed for test at 7 cantonal hospitals) (in Java)
- 1 scientific publication, 1 patent

Total budget: CHF 25K

Commission for Technology and Innovation (CTI) Project **03/2013 – 03/2015**
Title: "Privacy-Compliant Genomic Data Storage, Retrieval, and Processing"
Academic Partner: Laboratory for Communications and Applications (EPFL)
Industrial partner: Sophia Genetics SA
Main achievements:

- Prototype system for secure management of raw genomic data (BAM/SAM files) (in Java)
- 1 scientific publication, 1 patent

Total budget: CHF 800K

PATENTS

Method for Privacy-Preserving Medical Risk Tests. **J. L. Raisaro**, A. Erman, P. J. McLaren, J.-P. Hubaux, A. Telenti, Publication Nr. EP3016011 / US20160125141, Publication Date: 05.05.2016

Method To Manage Raw Genomic Data in a Privacy Preserving Manner in a Biobank. J.-P. Hubaux, E. Ayday, **J. L. Raisaro**, U. Hengartner, A. Molyneaux, Z. Xu, J. Camblong, P. Hutter, Publication Nr. WO/2014/202615, Publication Date: 24.12.2014

Privacy-Enhancing Technologies for Medical Tests Using Genomic Data. E. Ayday, J.-P. Hubaux, **J. L. Raisaro**, A. Telenti, J. Fellay, P. J. McLaren, J. Rougemont, M. Humbert, Publication Nr. WO/2014/040964, Publication Date: 20.03.2014

OTHER ACTIVITIES

Active member of Global Alliance for Genomics and Health (GAGH)

Reviewer for scientific conferences and journals
USENIX Security'14, IEEE Euro S&P'16, GenoPri'15'16'17, PoPETs'16'17'18, ACM Transactions on Privacy and Security , Nature Biotechnology, Computing Surveys, Bioinformatics

COMPUTING

Operating Systems: Windows, Mac OS, Linux
Scientific Programming: Python, Java, Matlab, Bash, Go
Web/Database development: HTML, CSS, JSP, SQL

MISC

Languages:
Italian and French: mother tongues
English: full professional proficiency

Interests:
Technology, Sports (volleyball, ski, tennis, hiking), Music, Travelling;

INVITED TALKS

"*Privacy-Enhancing Technologies for Protecting Genomic and Medical Data,*" TriNetX, Cambridge, MA, United States, Aug.'17

"*Addressing Beacon Re-Identification Attacks: Quantification and Mitigation of Privacy Risks,*" Global Alliance for Genomics and Health (GA4GH) Plenary Meeting, Vancouver, Canada, Oct.'16

"Protecting Genomic Data: Why and How," Molecular Medicine Working Group Meeting TMF, Germany, Sep.'16

"Design and Deployment of an i2b2 plugin for privacy-preserving exploration of genetic cohorts," The 4th i2b2 European Academic User Group Meeting, Pavia, Italy, Sep.'16

"*Addressing Beacon Re-Identification Attacks: Quantification and Mitigation of Privacy Risks,"* GA4GH – Elixir Security Group Meeting, EMBL-EBI, Wellcome Genome Campus, Hinxton, UK, Jun.'16

HONORS & AWARDS

Student Travel Award for Privacy-Aware-Computational-Genomics Workshop (PRIVAGEN), Tokyo, Japan, 2015

Student Travel Award for 1st International Workshop on Genome Privacy and Security (GenoPri'14), Amsterdam, Netherlands, 2014

PUBLICATIONS

J.L. Raisaro, C. Troncoso, M. Humbert, Z. Kutalik, A. Telenti and J.-P. Hubaux, *GenoShare: Supporting Privacy-Informed Decisions for Sharing Exact Genomic Data,* paper under submission

J.L. Raisaro, J.G. Klann, K.B. Wagholikar, H. Estiri, J.-P. Hubaux and S. N. Murphy, *Feasibility of Homomorphic Encryption for Sharing i2b2 Aggregate-Level Data in the Cloud,* AMIA Informatics Summit, 2018

J.L. Raisaro, P.J. McLaren, J. Fellay, M. Cavassini, C. Klersy and J.-P. Hubaux, *Are Privacy-Enhancing Technologies for Genomic Data Ready for the Clinic? A Survey of Medical Experts of the Swiss HIV Cohort Study*, Journal of Biomedical Informatics, 2018

J.L. Raisaro, J.R. Troncoso-Pastoriza, M. Misbach, J. Sa Sousa, S. Pradervand, E. Missiaglia, O. Michielin, B. Ford and J.-P. Hubaux, *MedCo: Enabling Privacy-Conscious Exploration of Distributed Clinical and Genomic Data,* 4th International Workshop in Genome Privacy and Security (GenoPri'17) at GA4GH annual meeting, Orlando, FL, Oct. 2017

J.L. Raisaro, G. Choi, S. Pradervand, R. Colsenet, N. Jacquemont, N. Rosat, V. Mooser, J.-P. Hubaux, *Protecting Privacy and Security of Genomic Data in i2b2,* to appear in IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2017.

D. Froelicher, P. Egger, J. Sá Sousa, **J.L. Raisaro**, Z. Huang, C. Mouchet, B. Ford and J.-P. Hubaux, *UnLynx: A Decentralized System for Privacy-Conscious Data Sharing*, Proceedings on Privacy Enhancing Technologies, 2017.

J. Sá Sousa, C. Lefebvre, Z. Huang, **J.L. Raisaro**, C. Aguilar-Melchor, M.-O. Killijian and J.-P. Hubaux, *Efficient and Secure Outsourcing of Genomic Data Storage*, BMC Medical Genomics, 2017.

J.L. Raisaro, F. Tramer, Z. Ji, et al. *Addressing Beacon Re-Identification Attacks: Quantification and Mitigation of Privacy Risks,* Journal of the American Medical Informatics Association, 2017

G. Choi*, **J.L. Raisaro*,** S. Pradervan, et al. *Privacy-Preserving Exploration of Genetic Cohorts with i2b2 At Lausanne University Hospital,* 3rd International Workshop in Genome Privacy and Security (GenoPri'16) at AMIA annual Symposium, Chicago IL, Nov. 2016. *(co-first authors)

P.J. McLaren***, J.L. Raisaro***, M. Aouri, et al., *Privacy-preserving genomic testing in the clinic: a model using HIV treatment,* Genetics in Medicine, Springer Nature, vol. 18, no. 8, pp. 814–822, 2016. *(co-first authors)

L. Barman, M.-T. Elgraini, **J.L. Raisaro**, E. Ayday and J.-P. Hubaux, *Privacy Threats and Practical Solutions for Genetic Risk Tests,* 2nd International Workshop in Genome Privacy and Security, Security and Privacy Workshops (SPW), 2015 IEEE, pp. 27-31.

J.L. Raisaro, E. Ayday, P. J. McLaren, J.-P. Hubaux, A. Telenti, *Privacy-Preserving HIV Pharmacogenetics: A Real Use Case of Genomic Data Protection,* 1st International Workshop in Genome Privacy and Security (GenoPri'14), Amsterdam, Neth., Jun. 2014

J.L. Raisaro, E. Ayday, and J.-P. Hubaux, *Patient Privacy in the Genomic Era,* Praxis (Bern. 1994)., vol. 103, no. 10, pp. 579–586, 2014.

J. Fellay, **J.L. Raisaro**, Z. Huang, M. Humbert, E. Ayday, P. J. McLaren, J.-P. Hubaux, and A. Telenti, *Practical Solutions for Protecting Individual Genomic Privacy, In American Society of Human Genetics (ASHG)*, 2014.

E. Ayday, **J.L. Raisaro**, P. J. Mclaren, J. Fellay, and J. Hubaux, *Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data,* USENIX Workshop on Health Information Technologies (HealthTech'13), 2013.

E. Ayday, **J.L. Raisaro**, J. Rougemont, and J.-P. Hubaux, *Protecting and Evaluating Genomic Privacy in Medical Tests and Personalized Medicine,* 12th ACM workshop on Workshop on privacy in the electronic society (WPES '13), 2013, pp. 95–106.

E. Ayday, **J.L. Raisaro**, U. Hengartner, A. Molyneaux, and J.-P. Hubaux, *Privacy-preserving processing of raw genomic data,* Data Privacy Management and Autonomous Spontaneous Security, Springer, 2014, pp. 133–147.

E. Ayday, **J.L. Raisaro**, and J.-P. Hubaux, *Personal use of the genomic data: Privacy vs. storage cost,* IEEE Global Communications Conference (GLOBECOM), IEEE 2013, pp. 2723-2729