UNIVERSITÉ DU LUXEMBOURG

PhD-FSTM-2022-100
Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 26/08/2022 in Luxembourg
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG EN INFORMATIQUE

by

Túlio Albuquerque Pascoal
Born on 21 July 1992 in Recife, Pernambuco, Brazil

Secure, privacy-preserving and practical collaborative Genome-Wide Association Studies

Dissertation defence committee

Dr. Marcus Völp, Dissertation supervisor
Associate Professor, Université du Luxembourg

Dr. Reinhard Schneider, Chairman
Full Professor, Université du Luxembourg

Dr. Gabriele Lenzini, Vice-chairman
Associate Professor, Université du Luxembourg

Dr. Yves Moreau, Member
Full Professor, University of Leuven

Dr. Erman Ayday, Member
Assistant Professor, Case Western Reserve University

"Success consists of going from failure to failure without loss of enthusiasm."

Acknowledgments

I would like to thank all the people who directly or not contributed to the development and achievements of the present thesis work:

- Thank you, Prof. Dr. Paulo Esteves-Veríssimo, for the opportunity to join the CritiX group at the University of Luxembourg and for the trust placed on me to be able to thrive in a Ph.D. Your experience, advice, and lessons are priceless, and I will undoubtedly carry them throughout my life.
- Thank you, Prof. Dr. Jerémie Decouchant, for the invaluable support, helpful lessons learned, and substantial guidance that led to the accomplishments obtained during my Ph.D.
- Thank you, Prof. Dr. Marcus Völp and Prof. Dr. Antoine Boutet for the useful inputs and suggestions that surely improved the overall quality of the works developed during my Ph.D.
- Thank you, Prof. Dr. Yves Moreau, Prof. Dr. Erman Ayday, Prof. Dr. Reinhard Schneider, and Prof. Dr. Gabriele Lenzini, for accepting to be juries of my thesis committee.
- Thanks to all members of my family, especially my parents and little brothers, for the support, positive thoughts, and understanding of not only my physical but also social absence in important moments during these last four years.

Declaration

I, Túlio Albuquerque Pascoal, declare that this thesis titled, "Secure, privacy-preserving and practical collaborative Genome-Wide Association Studies" and the work presented therein are my own. I confirm that:

- This work was done wholly or mainly while in candidature for the degree Docteur de l'Université du Luxembourg;
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated;
- Where I have consulted the published works of others, these are clearly attributed;
- Where I have quoted from the works of others, the sources are always given;
- Where the work presented in this thesis is based on work done by myself jointly with others, I have clearly outlined what was done by others and what I contributed;
- With the exception of such quotations, this is entirely my own work; and
- I have acknowledged all main sources of help.

Signed: _____

Date: _____

Abstract

Understanding the interplay between genomics and human health is a crucial step for the advancement and development of our society. Genome-Wide Association Study (GWAS) is one of the most popular methods for discovering correlations between genomic variations associated with a particular phenotype (i.e., an observable trait such as a disease). Leveraging genome data from multiple institutions worldwide nowadays is essential to produce more powerful findings by operating GWAS at larger scale. However, this raises several security and privacy risks, not only in the computation of such statistics, but also in the public release of GWAS results. To that extent, several solutions in the literature have adopted cryptographic approaches to allow secure and privacy-preserving *processing* of genome data for federated analysis. However, conducting federated GWAS in a secure and privacy-preserving manner is not enough since the public releases of GWAS results might be vulnerable to known genomic privacy attacks, such as recovery and membership attacks.

The present thesis explores possible solutions to enable end-to-end privacy-preserving federated GWAS in line with data privacy regulations such as GDPR to secure the public release of the results of Genome Wide Association Studies (GWASes) that are dynamically updated as new genomes become available, that might overlap with their genomes and considered locations within the genome, that can support internal threats such as colluding members in the federation and that are computed in a distributed manner without shipping actual genome data. While achieving these goals, this work created several contributions described below.

First, the thesis proposes **DyPS**, a Trusted Execution Environment (TEE)-based framework that reconciles efficient and secure genome data outsourcing with privacy-preserving data processing inside TEE enclaves to assess and create private releases of dynamic GWAS. In particular, **DyPS** presents the conditions for the creation of safe dynamic releases certifying that the theoretical complexity of the solution space an external probabilistic polynomial-time (p.p.t.) adversary or a group of colluders (up to all-but-one parties) would need to infer when launching recovery attacks on the observation of GWAS statistics is large enough. Besides that, **DyPS** executes an exhaustive verification algorithm along with a Likelihood-ratio test to measure the probability of identifying individuals in studies. Thus, also protecting individuals against membership inference attacks. Only safe genome data (i.e., genomes and SNPs) that **DyPS** selects are further used for the computation and release of GWAS results. At the same time, the remaining (unsafe) data is kept secluded and protected inside the enclave until it eventually can be used. Our results show that if dynamic releases are not improperly evaluated, up to 8% of genomes could be exposed to genomic privacy attacks. Moreover, the experiments show that **DyPS**' TEE-based architecture can accommodate the

computational resources demanded by our algorithms and present practical running times for larger-scale GWAS.

Secondly, the thesis offers **I-GWAS** that identifies the new conditions for safe releases when considering the existence of overlapping data among multiple GWASes (e.g., same individuals participating in several studies). Indeed, it is shown that adversaries might leverage information of overlapping data to make both recovery and membership attacks feasible again (even if they are produced following the conditions for safe single-GWAS releases). Our experiments show that up to 28.6% of genetic variants of participants could be inferred during recovery attacks, and 92.3% of these variants would enable membership attacks from adversaries observing overlapping studies, which are withheld by **I-GWAS**.

Lastly yet importantly, the thesis presents **GenDPR**, which encompasses extensions to our protocols so that the privacy-verification algorithms can be conducted distributively among the federation members without demanding the outsourcing of genome data across boundaries. Further, **GenDPR** can also cope with collusion among participants while selecting genome data that can be used to create safe releases. Additionally, **GenDPR** produces the same privacy guarantees as centralized architectures, i.e., it correctly identifies and selects the same data in need of protection as with centralized approaches. In the end, the thesis presents a homogenized framework comprising **DyPS**, **I-GWAS** and **GenDPR** simultaneously. Thus, offering a usable approach for conducting practical GWAS.

The method chosen for protection is of a statistical nature, ensuring that the theoretical complexity of attacks remains high and withholding releases of statistics that would impose membership inference risks to participants using Likelihood-ratio tests, despite adversaries gaining additional information over time, but the thesis also relates the findings to techniques that can be leveraged to protect releases (such as Differential Privacy). The proposed solutions leverage Intel SGX as Trusted Execution Environment to perform selected critical operations in a performant manner, however, the work translates equally well to other trusted execution environments and other schemes, such as Homomorphic Encryption.

Keywords: Federated GWAS, Privacy-preserving GWAS, Collusion-tolerance, Interdependent privacy, Genomic privacy, Distributed multi-party computation.

Contents

Abstract **v**

1 Introduction **1**
1.1 Problem statement 4
1.2 Contributions 5
1.3 Outline 6
1.4 List of publications 9

2 Background **11**
2.1 Genomics 101 12
2.2 Genome-wide Association Study (GWAS) 15
2.3 GWAS aggregate statistics 17
2.4 GWAS test statistics 17
2.5 Towards large-scale genomics 18
2.6 Towards collaborative and practical GWAS environments 20
2.7 Privacy-preserving *processing* of GWAS 23
2.7.1 Trusted Execution Environments (TEE) 27
2.8 Genomic privacy attacks on GWAS releases 32
2.8.1 Recovery attacks 33
2.8.2 Membership attacks 35
2.9 Privacy-preserving *releasing* of GWAS 36
2.10 Enabling practical federated GWAS 38

3 Related Work **40**
3.1 Federated GWAS 40
3.2 Solutions for privacy-preserving *processing* of GWAS 41
3.3 Solutions for privacy-preserving *releasing* of GWAS 45
3.4 Issues of interdependent GWAS releases 47
3.5 Overview and current stage of federated GWAS 48

4 Dynamic, Privacy-Preserving and Secure Federated GWAS (DyPS) 52
4.1 Conditions for safe GWAS releases . . . 53
4.1.1 Protecting recovery attacks . . . 54
4.1.2 Protecting membership attacks . . . 55
4.2 Detailing SecureGenome . . . 56
4.3 DyPS' system and threat models . . . 58
4.4 Overview of DyPS . . . 60
4.4.1 TEE-based architecture . . . 60
4.4.2 Workflow diagram . . . 61
4.5 Request selection to address test statistics releases . . . 63
4.5.1 Pseudocode of DyPS requests selection mechanism . . . 64
4.5.2 Scaling the GWAS over number of SNPs . . . 66
4.5.3 Composition property and proofs for genome requests selection to protect test statistics releases . . . 67
4.5.4 Membership tests to address aggregate statistics releases . . . 69
4.5.5 Pseudocode of DyPS exhaustive verification mechanism for aggregate statistics . . . 71
4.5.6 Verification for pairwise statistics releases . . . 72
4.6 Experimental evaluation . . . 72
4.6.1 Experiments setup . . . 72
4.6.2 Bandwidth, CPU and memory consumption . . . 74
4.6.3 Naïve dynamic release vs. DyPS . . . 75
4.6.4 Impact of dynamic SNP-set scaling . . . 76
4.6.5 Impact of colluding biocenters . . . 79
4.6.6 SNP selection for aggregate statistics . . . 80
4.6.7 DyPS vs. static release of aggregate statistics . . . 81
4.6.8 DyPS over a large-scale GWAS . . . 84

5 Privacy-preserving Interdependent GWASes (I-GWAS) 86
5.1 I-GWAS' system and threat models . . . 87
5.2 Safety conditions for interdependent GWASes . . . 89
5.2.1 Recovery attack mappings . . . 90
5.2.2 Singlewise allele frequencies search space analysis . . . 91
5.2.3 Pairwise allele frequencies search space analysis . . . 92
5.2.4 Test statistics search space analysis . . . 92
5.2.5 Protecting interdependent GWASes against recovery attacks 93
5.2.6 Sequential releases of GWASes . . . 93
5.2.7 Scaling with the number of GWASes . . . 95
5.2.8 Allowing safe genome removals . . . 96
5.3 Membership attacks on interdependent GWASes . . . 97
5.4 Protecting interdependent GWASes against membership attacks . . . 98

5.5 Experimental evaluation . . . 102
5.5.1 Privacy and data utility . . . 103
5.5.2 Comparison to Differential Privacy . . . 105
5.5.3 Running time and complexity . . . 107

6 Genome Distributed Private Release (GenDPR) 109
6.1 GenDPR' system and threat models . . . 111
6.2 Genome Distributed Private Release (GenDPR) . . . 113
6.2.1 Architecture and overview . . . 113
6.2.2 Verification for mitigating recovery attacks . . . 114
6.2.3 Workflow . . . 114
6.2.4 MAF analysis (Phase 1) . . . 115
6.2.5 LD analysis (Phase 2) . . . 116
6.2.6 LR-test analysis (Phase 3) . . . 117
6.2.7 Collusion-tolerant GenDPR . . . 121
6.3 Experimental evaluation . . . 123
6.3.1 Bandwidth, memory and CPU usage . . . 123
6.3.2 Running time . . . 124
6.3.3 Correctness . . . 126
6.3.4 Collusion-tolerant GenDPR . . . 128

7 A Holistic Approach (combining DyPS, I-GWAS and GenDPR) 130
7.1 Holistic architecture . . . 130
7.2 Holistic framework . . . 131
7.3 Comparison between SecureGenome, Differential Privacy and the solutions offered in this thesis . . . 133
7.4 Assessing the limitations of the proposed solutions . . . 136

8 Conclusions and Future Work 139
8.1 Conclusions and outcomes of the thesis . . . 139
8.2 Future work . . . 142

Terminology 148

References 149

List of Figures

2.1 Genomics 101 - background and representation. 13
2.2 Moving from a stand-alone to a federated GWAS setting. 21
2.3 A typical setting of SGX-based solutions. 30
2.4 Illustration of a recovery attack on the observation of GWAS statistics. 34
2.5 Illustration of a membership attack on the observation of GWAS statistics. 36

4.1 Observable data for privacy attacks on GWAS. 53
4.2 DyPS' system and threat models. 59
4.3 DyPS' federated architecture. 61
4.4 DyPS' workflow diagram. 62
4.5 SNPs set dynamic scaling. 66
4.6 Successive releases of test and aggregate statistics as new genome addition or removal requests are executed. 70
4.7 Running time of the brute force and DyPS request selection approach for test statistics over 5,000 SNPs ($LtoN(L) = 38,040$) ($B = 4, f = 0$). 74
4.8 Running time of DyPS request selection approach for a test statistics over 5,000 SNPs ($LtoN(L) = 38,040$) inside the SGX enclave ($B = 4, f = 0$). 75
4.9 Comparison between the naïve release approach and DyPS under different scenarios (r rounds, and L SNPs) for ($B = 4, f = 0$). . . . 76
4.10 DyPS with or without dynamic SNP-set scaling - round delays for a GWAS consisting of 3,000 SNP positions ($LtoN(L) = 21,600$) ($B = 4, f = 0$). 77
4.11 DyPS without and with dynamic scaling of the SNPs set for a GWAS consisting of 3,000 SNP positions ($LtoN(L) = 21,600$) ($B = 4, f = 0$). 78
4.12 ($B - f$) DyPS: Time to release in rounds for genome requests for a GWAS consisting of 300 SNP positions ($LtoN(L) = 1,598$) during 1,000 rounds, and different number of possibly colluding biocenters. 80

4.13 Running time for the different steps of DyPS execution for a GWAS studying 10 SNP positions ($B = 5, f = 0$). Reference group size: 1,000. Total number of real genomes used: 2,000. 81
4.14 Comparison between a static approach and DyPS for releases of aggregate statistics for a GWAS studying 1,000 SNP positions ($B = 5$, $f = 0$). Reference group size: 1,000. Total number of real genomes used: 2,000. 82
4.15 Running time for the different steps of DyPS for a GWAS studying 1,000 SNP positions ($B = 5$, $f = 0$). Reference group size: 1,000. Total number of real genomes used: 2,000. 83
4.16 Running time for $f = 0$ and $f = 4$ with aggregate statistics computed over 1,000 SNP positions. Reference group size: 1,000. Total number of real genomes used: 2,000. 83
4.17 Running time for the different steps of DyPS execution for a GWAS studying 5,000 SNP positions ($B = 5, f = 0$). Reference group size: 13,035. Total number of real genomes used: 27,895. 84

5.1 I-GWAS system and threat model. 89
5.2 Illustration of two overlapping GWASes. 90
5.3 Smallest number of genomes N_2 that a GWAS that overlaps with a previous GWAS should use for a safe release depending on their overlapping SNP-set size (L_{ovl}) and genomes set size (N_{ovl}). 95
5.4 Exhaustive verification process to protect interdependent GWASes releases against membership attacks. 99
5.5 Vulnerable SNPs and release coverage when protecting interdependent GWASes against membership attacks. 104
5.6 Results of GWAS releases using ϵ-DP releases using ($\epsilon = 0.1$ and $pvb = 0.12$). Cut-off of the first 200 of 1,000 SNPs. 107
5.7 Running times of the brute force and I-GWAS approach for protecting recovery attacks on interdependent GWASes. 108

6.1 GenDPR system and threat model. 112
6.2 GenDPR architecture components. 113
6.3 GenDPR workflow. 114
6.4 GenDPR distributed LR-test phase scheme. 118
6.5 Running time comparison (1,000 SNPs). 125
6.6 Running time comparison (10,000 SNPs). 126

7.1 Holistic multi-enclave system architecture. Steps (1), (2), (3) and (4) are presented in Figure 7.2 131
7.2 Holistic framework for federated practical GWASes. 132

List of Tables

1.1 List of publications. 10

2.1 Genome encoding for GWAS. 16
2.2 A singlewise contingency table for phenotype p. 16
2.3 A GWAS pairwise contingency table for two variants. $\mathbf{SNP}_i$ and $\mathbf{SNP}_j$, where $i, j \in \{1, \cdots, L\}$. 17
2.4 Overall comparison of privacy-preserving *processing* approaches. . . 27
2.5 Overall comparison of the performance of privacy-preserving *processing* approaches. 28

3.1 Overview of existing federated GWAS solutions. **STPC:** Secure Two-Party computation; **CPs:** Computing parties; **SS:** Secret-Sharing mechanism; **GC:** Garbled circuit; **MCMC:** Markov Chain Monte Carlo; ***:** Releases *yes/no* answers, which can be vulnerable to similar Beacon's privacy attacks [AAC21; Rai+17a; SB15a; Al +17; VAC19]. 49

4.1 Release conditions for GWAS aggregate or test statistics computed over L SNPs and N individuals. 54
4.2 Average number of processed addition and removal requests, number of GWAS releases, and average round delays for addition and removal requests depending on the number of colluding biocenters. 79
4.3 DyPS' average memory consumption inside the enclave depending on several controls group sizes and 5,000 SNPs. 85

5.1 I-GWAS protection against membership attacks with four GWASes. 104
5.2 Comparison between I-GWAS and the standard ϵ-DP using Laplace mechanism under several settings. The results represent the average of 100 repetitions. 106

6.1 Average resource demand of GenDPR. 124
6.2 Comparison of the selected SNPs after each phase of the privacy-protecting verification. 127

6.3 Collusion-tolerant **GenDPR** results considering 10,000 SNPs and 14,860 genomes. 128

7.1 Comparison between SecureGenome, Differential Privacy and this work solutions for practical GWAS. 135

Chapter 1

Introduction

The discovery of the deoxyribonucleic acid (DNA) as a carrier of genetic information was one of the most important steps in science for understanding how living beings are formed and how their characteristics are passed through generations. Several years after its discovery, DNA was fully sequenced and digitized in 2003. Such a scientific breakthrough enabled scientists to better understand and study the human DNA in more detail. Understanding the human DNA and how genetic information impacts on health, abilities, and lifespan of individuals, to name a few, is extremely important for contributing to a healthier and prosperous society.

With the advancement of DNA research, new findings appeared. For instance, it allowed the discovery that humans share almost 99.9% of their genetic code. Such genetic loci, where humans do not share the same genomic information, are what make us unique as human beings. These genetic variations are frequently used in broader studies, such as Genome-Wide Association Study (GWAS). GWAS is a popular type of statistical genomic study that has been developing medicine by allowing researchers to identify genetic variants associated with a particular phenotype (i.e., a specific trait, such as a disease). In fact, GWAS has been adopted to find disease susceptibility and predisposition to risk factors (e.g., drug or alcohol addiction) and to improve personalized medicine.

Over the years, the development of bioinformatics has enabled the creation of more sophisticated, powerful, and cheaper DNA sequencing machines. As a result, individuals' DNA is now being sequenced more rapidly and cheaply. Thus, enabling accessible DNA sequencing for the masses. This fact has increased the number of individuals being sequenced and willing to participate in studies, which directly helps the progress and confidence in GWAS since it can now be conducted relying on a more significant number of individuals. Notwithstanding this fact, the digital format of human DNA is enormous in size. A regular Variant Call Format (VCF) file usually used in GWAS might take 125 GB considering the entire genome. Such a characteristic motivates sequencing companies and genomic data holders to

outsource the storing of genome data to third-party service providers, such as cloud servers, to reduce operational costs, for example. Besides that, to produce higher precision results, especially in terms of statistical confidence, and to remove biased findings related to using homogeneous populations instead of considering multiple population ancestries around the globe, the research community has been adopting larger-scale GWAS by combining genome data from several institutions, usually geographically dispersed. This phenomenon leads to the creation of collaborative environments to perform such large-scale (collaborative) GWAS. This setting is usually referred to as federated analysis, which is known as federated GWAS for Genome-Wide Association Studies.

Additionally, the benefits of GWAS can be ameliorated by open-access releases of its results. Indeed, public releases of GWAS statistics would contribute to a faster and broader access to medical/health research findings, and therefore benefiting society as a whole. As a result, nowadays, biocenters are encouraged to conduct federated GWASes and release their results publicly.

Despite the above, due to its importance when identifying and carrying humans' genetic code, DNA is very sensitive data. Hence, individuals need to be assured of trust and motivation before accepting to share their genomes. Indeed, the leakage of DNA entails high privacy risks because:

1. It cannot be revoked.

2. It infringes not only the donor's privacy but also of their family since DNA is inherited from parents, and

3. It might bring forward a variety of unethical activities from malicious players who may access leaked genomic data.

Therefore, systems that manage and operate genome data must ideally comply with the highest privacy and security standards. In addition, when considering a federated setting, where each data holder is responsible for their sequencing costs, enforcing local security and privacy of donor's genomic data is not enough. In particular, genome data repositories need to assert that the privacy of their data cannot be breached during their local data are being outsourced and processed in the GWAS federation. For instance, being aware of the potential presence of collusion among members of the federation aiming to attack others' data. If such data privacy protection cannot be enforced, genomic data centers will not be willing to participate in collaborative environments.

In this context, solutions in the literature have been relying on cryptographic methods to enable secure outsourcing and privacy-preserving *processing* of genomic data while aggregating and computing GWAS statistics for federated analysis.

Commonly used mechanisms to assist in preserving privacy include: Homomorphic Encryption (HE), Secure Multiparty Computation (SMC), Differential Privacy (DP), and Trusted Execution Environments (TEE). Unfortunately, whereas it is true that GWAS plays an essential role by producing studies that allow the identification of genotype-phenotype correlations, it has been shown that the publication of GWAS statistics can be subject to genomic privacy attacks (even if the released GWAS statistics were computed leveraging privacy-preserving *processing* schemes like the ones cited above). In fact, in 2008, Homer et al's. attack [Hom+08] showed that individuals could have their participation linked to a specific GWAS by the observation of its released GWAS statistics. Such a privacy breach is very critical since it might reveal to an adversary whether a victim has a particular disease or not. For example, insurance companies may misuse this improperly acquired information to accept or reject individuals' applications. The disclosure of this attack has lead the National Institute of Health (NIH) to revoke access to all open-access GWAS' results in their database [ZN08]. Nowadays, only authorized researchers can access GWAS releases, limiting access to new studies' findings and consequently decelerating the spreading and benefits that GWAS brings to society.

Concluding from the above, enforcing only secure and privacy-preserving *processing* of genomic data is not enough. Currently, reconciling secure and privacy-preserving *processing* with privacy-preserving *releasing* of GWAS is a crucial challenge that has not been tackled by the research community yet. On the one hand, the computation of GWAS must preserve genomic privacy against attacks mounted during the sharing and processing of federated GWAS. On the other hand, releases of GWAS results must preserve genomic privacy against known attacks, such as recovery and membership attacks.

Motivated by these facts, the present thesis, for the first time, simultaneously addresses:

- Secure and privacy-preserving *processing* of genome data;
- Privacy-preserving *releasing* of GWAS results, impeding the disclosure of secret and private data of both donors and data holders;

On top of reconciling secure and privacy aspects of federated GWAS, this work not only anticipates the dissemination and popularity of GWAS but also aims to obey current data-privacy regulations, which brings new issues. More specifically, the present thesis envisions to enable the following properties for allowing *practical GWAS* under collaborative environments:

- Public access to dynamic releases of GWAS results, i.e., allowing GWAS to be updated over time as soon as genomes are added or removed (a desirable

requirement imposed by data-privacy regulations, such as GDPR), while enforcing that only safe releases (i.e., protected against known genomic privacy attacks) are exposed.

- Collusion-tolerance, i.e., considering the presence of honest-but-curious institution(s) in the GWAS federation that might collude to combine their knowledge (aggregated genome data) to facilitate or make genomic privacy attacks possible even if safe releases conditions are being enforced.

- Interdependent genomic privacy, i.e., enforcing the genome data privacy of donors and institutions under the presence of potentially overlapping studies that might share genome and studied genome locations, which becomes present when multiple GWASes are conducted by one or more federations.

- Data locality, i.e., keeping genome data as most as possible at the premises of the institutions responsible for sequencing them while conducting federated analysis. For instance, allowing the generation of private releases without requiring actual genome data outsourcing.

In summary, the current thesis aims at offering mechanisms that acknowledge and enable the properties presented above in a homogenized form. Thus, presenting approaches to make *practical GWAS* a reality. In particular, the thesis designs and develops frameworks to allow dynamic releases or updates of interdependent GWASes will never infringe genomic privacy of individuals donating their genome data or withdrawing consent of participation while securing data of each data holder even when suffering collusion attacks from other's parties in the federation while granting open-access releases of GWASes.

The hypothesis investigated in this thesis is therefore:

It is possible to enforce secure and privacy-preserving dynamic releases (to allow individuals the ability to safely withdraw or give consent of participation at any time, as required by data-privacy regulations, such as GDPR) of interdependent (overlapping) GWASes in federated environments, as well as to deal with internal threats, such as collusion attacks being launched by malicious federations members, using a secure and practical system architecture, while achieving low degradation in terms of data-utility and accuracy-loss so that large-scale and practical federated GWAS can be conducted in a scalable and end-to-end privacy-aware manner.

1.1 Problem statement

Due to the particularities of human DNA (huge size, sensitiveness, vulnerability to inference attacks from released GWAS results, etc.), genome data needs to be managed with proper care. Additionally, when relying on such data to perform GWAS

in cooperative environments, new system-related, privacy and security issues arise. In particular, federated GWAS has been conducted by relying on existing privacy-preserving approaches that are mainly based on cryptographic methods, such as HE, SMC, and TEE. However, processing and producing federated GWAS in a privacy-preserving and secure manner is not enough. Several works have shown that publicly shared GWAS results might be subject to privacy attacks [Hom+08; Wan+09; Jac+09; Cai+15; Im+12; SB15b; Cra+11]. In particular, some works have evaluated and offered safety conditions for GWAS releases [San+09a; USF13; Zha+14; Tra+15; SSB16; Jia+14; SBS19; AAU20a; Hum+14]. Nevertheless, these works have considered only static GWAS, leaving it questionable whether genome data remains secure under the statistics updates (assuming the presence of addition and removal requests). For instance, when new individuals are sequenced or when some participants would like to withdraw their participation from studies in order to be compliant with data privacy regulations [BLR19; Des+].

This thesis evaluates and identifies that new conditions for keeping dynamic GWAS releases safe arise and must be enforced. It shows that removal operations can undermine the privacy of new and older participants' genomic data if not assessed with precaution. Furthermore, the present thesis also shows that the presence of overlapping data such as genomes and SNPs shared among multiple studies might also compromise the privacy of participants.

Besides, existing works have not considered possibility of collusion among federation members to attack others' data (e.g., because of economic or conflict of interest reasons). In particular, this thesis identifies that such colluding parties can conjointly aggregate their data and so isolate small enough data of the victim party so that privacy attacks can be successfully launched on it.

Lastly, yet significantly, the thesis extends its solutions so that the proposed genomic privacy-protection mechanisms can be conducted in a distributed manner and without demanding the outsourcing of individuals' genome data. In other words, allowing federation members to jointly conduct the privacy analysis while not shipping genome data across their premises.

In summary, this work presents and evaluates solutions to enable secure and end-to-end privacy-preserving GWAS while addressing aspects of *practical GWAS*, e.g., dynamic releases, data privacy regulation-compliant, collusion-tolerance, and interdependent private releases.

1.2 Contributions

This thesis offer solutions to first address a remaining challenge on the state-of-the-art of federated GWAS. Namely, to reconcile privacy-preserving *sharing*, *processing* and *releasing* of federated GWASes. On top of that, this thesis assumes novel real-

life and contemporary requirements that have not been addressed so far, which lead to a more practical perspective for GWASes, which are defined as *practical GWAS* properties. In particular, the solutions developed in this thesis support:

- The production of safe open-access releases of dynamic GWASes updates while protecting them against privacy attacks also assuming the presence of multiple (interdependent) studies;
- Consent withdrawal from donors while safely updating GWAS results;
- The production of safe releases by enabling parties to securely aggregate their data even when parties in the federation are colluding to leak others' data (i.e., collusion-tolerance);
- Distributed verification of privacy-preserving releases of GWAS. More specifically, it extends the offered protocols to allow them to be performed in a distributed fashion and without relying on the outsourcing of actual genome data from federation members.

1.3 Outline

Chapter 2 provides the background used to support the work. Firstly, it introduces the basics of genomics, while describing the main concepts and the features of the human genome. Next, it presents the main issues and challenges when digitally managing such sensitive data. Secondly, it describes how a Genome-Wide Association Study (GWAS) is conducted by retrieving and encoding genomic data from individuals and computing statistics over them to find observations that might correlate genetic variations to a particular trait (e.g., a disease). It then describes the types of statistics GWAS produces and explains the needed data to support these studies. Posteriorly, it discusses the challenges of performing GWAS in a federated setting, where several data holders (e.g., biocenters usually geographically separated) contribute with genomic data from individuals sequenced in their perimeter. Next, it describes and compares existing privacy-preserving approaches to enable secure and private computation over data in a multiparty setting. Nevertheless, as enforcing privacy-preserving *processing* of genomic data is not enough, section 2.8 introduces the existing attacks on GWAS results releases and current countermeasures. Finally, the section summarizes the open challenges for enabling *practical GWAS*, which copes with collusion among federation members, complies with existing data privacy regulations, ensures safe releases under a dynamic setting where GWAS are updated over time and removes the need for the genome data outsourcing while conduction privacy-protection mechanisms.

Next, Chapter 3 first presents the existing works in the literature that enables the participation of several genome data holders to conduct federated GWAS. It then details solutions that allow privacy-preserving *processing* of genome data. The following sections discuss the issues of releasing GWAS results as they might be vulnerable to recovery and membership attacks, which might compromise the privacy of individuals participating in studies. Later, it introduces existing mitigation mechanisms to protect GWAS releases. For instance, presenting existing solutions that rely on Differential Privacy (DP) and statistical inference methods such as Likelihood-ratio tests (LR-tests). Finally, it describes new privacy concerns that arise under the presence of multiple releases of interdependent GWASes, i.e., studies that eventually use the same genome(s) and consider the same SNP position(s). As identified later in the thesis, such a scenario might compromise the genomic privacy of individuals even if releases are produced considering existing privacy-protection.

Chapter 4 introduces the first solution created to address some of the open challenges to enable *practical GWAS* described previously. Namely, this section presents **DyPS**, which offers: (i) fully private federated GWAS by combining privacy-preserving *processing* and *releasing*; (ii) privacy-aware in the sense of being compliant with data privacy regulations (such as GDPR) and hence allowing participants to withdraw consent at any time; (iii) safe releases of dynamic GWAS, which have statistics updated over time given the addition and removal of participants; and (iv) collusion-tolerant algorithm that enables GWAS federations to select a safe batch of genome requests so that federation members' data cannot be subject to privacy attacks by up to all-but-one colluding players while still granting donor's privacy against external adversaries. **DyPS** leverages a TEE architecture. More specifically **DyPS** uses Intel SGX as its privacy-preserving enabler to allow secure outsourcing and processing of genomic data from federation members. In addition, **DyPS** extends statistical inference methods to protect dynamic releases of GWAS against membership and recovery attacks. At the same time, enabling collusion-tolerance. To the best of author's knowledge, **DyPS** is the first approach that reconciles privacy-preserving *processing* and *releasing* of GWAS. The experimental results show that **DyPS** updates releases with a reasonable additional processing delay (11% longer) while protecting genomic privacy. In particular, even though a naïve approach is able to produce more releases, it compromises the privacy of up 8% of participating genomes. Furthermore, **DyPS** does not decrease the amount of aggregate statistics considered in releases, while being able to produce multiple releases when compared to a static release that can release only once. **DyPS** shows practical and scalable performance, presenting reasonable running time and communication costs under a variety of scenarios. It was experimented with assuming different federated GWAS settings, considering from 3

to 7 biocenters, up to 300,000 SNP positions, 6 million simulated genomes, and approximately 35,000 real genomes.

Moving forward, Chapter 5 extends the protocol and algorithms introduced in DyPS to support the creation of safe releases under the presence of multiple interdependent GWASes. This chapter first evaluates and identifies the new conditions to enable safe releases of GWASes statistics considering the presence of multiple and potentially overlapping studies. In particular, Section 5.2.1 and Section 5.3 show that enforcing dynamic single-GWAS releases is not safe as adversaries might take advantage of overlapping data to decrease the solution space they have to infer when launching recovery attacks or to increase the identification power of individuals participating in studies. As a result, the chapter introduces **I-GWAS**, a novel framework that allows privacy-preserving *releasing* of interdependent GWASes. **I-GWAS** is also able to incorporate DyPS features such as dynamic releases and consent withdrawal to participants while not allowing genomic privacy attacks on the releases. **I-GWAS** evaluation shows its performance when protecting overlapping releases. For instance, it was found that up to 92.3% of genomes might be vulnerable to membership inference, and 28% can be subject to recovery attacks when adversaries are able to observe several GWASes releases from multiple sources that share genomes. **I-GWAS** also presents a better release utility when compared to dynamic Differential Privacy-based scheme for GWASes releases. Even though **I-GWAS** needs to withhold the disclosure of statistics over some SNPs, it does not apply any data perturbation to the results. In contrast, DP-based releases suffer data perturbation, which impacts the accuracy of the results. In addition, unfortunately, no scheme able to offer DP guarantees over continuous releases of GWAS statistics has been provided so far, which limits its usability in such a setting. Therefore, **I-GWAS** represents the first step towards dynamic and privacy-preserving interdependent GWASes.

Next, Chapter 6 introduces a novel distributed workflow for the assessment of private GWAS releases. In particular, we offer Genome Distributed Private Releases (GenDPR), which aims to remove the need for genome data outsourcing while being able to correctly perform the privacy-protection analyses to create safe releases. In particular, GenDPR executes the privacy-protection statistical analyses to select data over which releases can be computed while not allowing membership inference attacks in a distributed manner. Since the verification to evaluate which genome data can be used for the creation of safe releases is performed jointly by the members of the federation, GenDPR removes the need for genome data outsourcing and storage in a centralized location. Moreover, GenDPR can also cope with the presence of colluding parties aiming to isolate other institutions' data to leak their data. The experiments show that GenDPR imitates the same outputs of the LR-test when compared to a centralized version while needing slightly longer

running times due to extra coordination and aggregation tasks, which shows its correctness.

Finally yet importantly, Chapter 7 provides an overview of a holistic scheme to accommodate DyPS, I-GWAS and GenDPR solutions simultaneously and in a homogenized form. Next, it provides a comparison between statistical inference methods (SecureGenome), Differential privacy and the solutions of this work. In particular, correlating the limitations, advantages and properties that each mechanism allows, before discussing and addressing the limitations of the methods proposed in the thesis.

Lastly, Chapter 8 presents the conclusions, outcomes and planned future work of the thesis.

1.4 List of publications

Following the targeted goals discussed in Section 1.2, the approaches developed during this thesis generated several contributions. More specifically, three main manuscripts have been written, where several of them have already been published in high-ranked international journals and/or conferences of the privacy-enhancing technologies and genomic privacy specialities.

Table 1.1 summarizes the publications accomplished by the work developed during the thesis.

Table 1.1: List of publications.

Year	Title	Short description	Status	Contributions
2020	DyPS: Dynamic, Private and Secure GWAS	A novel framework to allow privacy-preserving *processing* and *releasing* of federated GWAS while enabling continuous and GPDR-aware releases.	**Published** and presented at PETS 2021 (July 12, 2021).	Partially conceived and designed the idea, partially collected the data, fully carried implementation and experiments, partially conducted analysis and validation, partially wrote the paper.
2021	Towards dynamic federated GWAS	A summary of current challenges to pave the way for dynamic GWAS. DyPS is also presented.	**Published** and presented at GenoPri 2021 (August 22, 2021).	Partially conceived and designed the idea, fully collected the data, fully carried implementation and experiments, partially conducted analysis and validation, partially wrote the paper.
2022	Towards practical GWASes: Overview and Challenges	A position paper that discusses current issues and unsolved challenges for the development of practical GWAS elaborating on interdependent and multi-party privacy aspects.	**Published** and presented at PETS'22 Workshop on Interdependent and Multi-party Privacy (July 11, 2022).	Fully conceived and designed the idea, fully collected the data, fully carried implementation and experiments, partially conducted analysis and validation, fully wrote the paper.
2022	I-GWAS: Privacy-preserving Interdependent Genome-Wide Association Studies	A novel framework to assess and enforce privacy-preserving continuous releases of multiple overlapping GWASes under federated environments.	**Accepted** at PETS 2023 (July 10-14, 2023).	Partially conceived and designed the idea, fully collected the data, fully carried implementation and experiments, partially conducted analysis and validation, partially wrote the paper.
2022	Distributed and secure assessment of privacy-preserving releases of GWAS	A multi-enclave distributed workflow for the assessment of privacy-preserving GWAS releases without genome data outsourcing and centralized data.	**Accepted** at Middleware 2022 (November 7-11, 2022)	Fully conceived and designed the idea, fully collected the data, fully carried implementation and experiments, partially conducted analysis and validation, partially wrote the paper.

Chapter 2

Background

This chapter familiarizes the reader with the basic knowledge and aspects that support this work. The chapter has the following outline. The first section introduces the history, advancements, and details the nature and particularities of the human genome. Next, it presents the importance, contributions, benefits, and how a Genome-Wide Association Study (GWAS) is conducted. The following section describes the concept and the advantages of conducting large-scale GWAS in collaborative environments (i.e., federated GWAS). Next, it discusses approaches, challenges, and trade-offs to enable fully privacy-preserving cooperative GWAS environments, especially considering the existing genomic privacy attacks from GWAS releases observation.

At the same time, this chapter contemplates the new assumptions targeted in this work to enable *practical GWAS*, which assumes new functionalities, such as dynamic updates of GWAS statistics, GDPR-aware studies, interdependent privacy issues tailored to the existence of multiple studies, and collusion-tolerant federated GWAS.

The solutions offered in this thesis rely on TEEs to provide privacy-preserving outsourcing and processing of federated GWAS and extend existing genome-oriented statistical inference methods for the protection of GWAS statistics releases. In particular, the proposed solutions extends and adapts the Zhou et al.'s [Zho+11] complexity analysis of genomic recovery attacks and SecureGenome [San+09b; San+09a] to allow the creation of private releases of GWAS statistics while supporting the properties tailored to *practical GWAS*. Therefore, due to their importance, a more detailed discussion on TEEs, Zhou et al.'s and SecureGenome conditions for safe GWAS releases are presented when appropriate.

2.1 Genomics 101

Since its first discovery in the middle of the 1860s, deoxyribonucleic acid (DNA) has been substantially studied by the academic community. DNA is an elongated polymer composed of nucleotides that form the genetic code of all living beings. The DNA assembles the following nucleotides: adenine (A), thymine (T), cytosine (C), and guanine (G), in which its opposite bases strand their pairs specifically. An (A) pairs with a (T), and a (C) always pairs with a (G). The complete human DNA consists of more than 3 billion pairs of nucleotides over its 23 pairs of chromosomes. A haploid genotype is represented by the group of nucleotides on one chromosome (inherited from a single parent). In contrast, a diploid genotype consists of grouping nucleotide pairs from both chromosomes (inherited from both parents). In the chromosomes is present all human's genetic code.

Interestingly enough, it is known that any randomly chosen humans share approximately 99.9% of their nucleotides. In such non-shared regions, humans present genetic variations named Single Nucleotide Polymorphism (SNPs). Those variations are what make us unique as human beings. In particular, a SNP represents that different nucleotide information is found among individuals. Usually, these variations are identified when more than 0.5% of a population does not express the same nucleotide at a certain position of the genome, or when different nucleotides are found when a given genome sequence is compared to the human reference genome [Zoo+16]. Particularly, each SNP has two possible alleles (inherited from each one of our parents). An *allele* represents a nucleotide found in a variant position. Usually, two types of alleles can be identified in SNPs: (i) major allele, which is the most common nucleotide in the population (represented by 0's), and (ii) minor allele, which is the rarer nucleotide within the population, possibly the least common (represented by 1's). SNPs are frequently used for genomic studies that aim at finding correlations between genetic variations with particular traits or phenotypes. Figure 2.1 summarizes and provides a representation of the main aspects of human DNA.

In April 2003, the first human DNA was sequenced, costing approximately $3 billion US dollars while taking thirteen years to be sequenced [Tir]. From that moment on, DNA sequencing has been becoming cheaper and faster thanks to the advent of Next Generation Sequencing (NGS) machines that improved the computing and parallelism capacity of sequencing genomic data. For example, nowadays, one individual can have their whole DNA sequenced for less than $1,000 US dollars [Lam+18] and in two days at most [Lew]. Furthermore, Illumina, the largest manufacturer of DNA sequencers, is predicting to sequence genomes for less than $100 dollars each in the next years [Her]. Thanks to these developments, DNA has been becoming increasingly affordable for individuals of any social class. This outperformed Moore's Law growth of DNA costs has accelerated and increased the

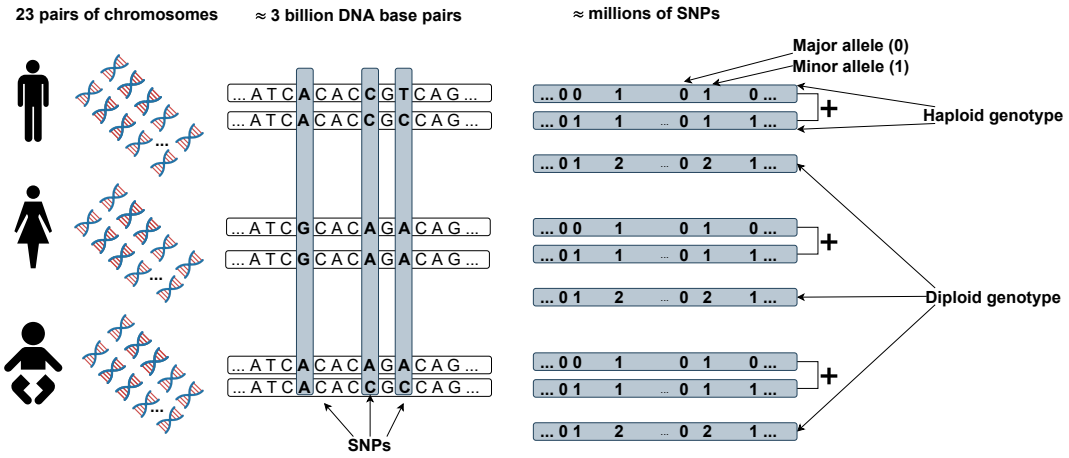


Figure 2.1: Genomics 101 - background and representation.

number of ongoing human genome studies. Wide-accessible genome sequencing is the base of these studies that can help individuals and researchers to earlier detect diseases' predispositions, improve personalized medicine, find better treatment methods, and even improve dating choices [Blo].

Nevertheless, despite all those benefits, a broader and easier access to such data has to be performed with proper care. Indeed, there are a variety of issues regarding genomic data. In the following, the main particularities, challenges, and privacy risks of managing genomic data are presented.

- **Price and size:** The size of the digital DNA data is enormous. The most common file formats for representing DNA data are FASTQ/FAST files[1], Binary Alignment Map (BAM), and Variant Call Format (VCL) files. Their size varies and correspond to ≈200 GB/genome, ≈100 GB/genome and ≈125 MB/genome, respectively [Rei]. Because of that, sequencing companies, hospitals, and research centers tend to outsource the storing and processing of such data to more powerful cloud servers as a means to reduce operational and storing costs [Fer+17; Fer+19];

- **Sensitiveness:** Personal genomic data reveals detailed and unique characteristics of each individual. Besides that, DNA data have been extensively used to find the identity of a person [Bal+11]. Indeed, it is already possible to predict individuals' faces, skin color, height, and weight from their genome [Lip+17]. Moreover, Cai et al. [Cai+15] showed in a recent work that individuals can be uniquely identified by the observation of a small subset of

[1]Text-based format file that stores nucleotide sequences. The difference between FASTQ and FASTA files is that the former has a quality score for each sequencing line.

25 randomly selected informative SNPs. Besides, DNA contains very sensitive health information about a person, such as physical and mental characteristics, disease status and predisposition, comorbidities, or environmental factors that might contribute to them [Aze18], which malicious entities might improperly use. Furthermore, any leakage of genomic information regarding a person, e.g., some region of his/her DNA or SNPs, does only directly reveal information about the concerned individual but also leaks information of his/her relatives such as parents, siblings, and children up to 5 degrees of separation for ethnicity [BG17]. Additionally, previous work has shown that genomic privacy deteriorates over time [Bac+18];

- **Ethical issues:** The above issues might lead to discrimination based on our genetic information, which has been a concern for over thirty years [Aze18]. Once DNA reveals sensitive characteristics of individuals, e.g., predisposition to drug addiction, disease possibilities, and even our IQ level (to name a few), such information might be used by hostile entities to undermine or take advantage of people. For example, malicious insurance companies might deny health insurance coverage for particular individuals or companies might avoid to hire certain applicants based on a person's genetic information, e.g., predisposition to a very rare disease (which means increased treatment costs) and low IQ level, respectively.

- **Control over data and trust:** The actual ownership of our DNA is still an issue. Nowadays, customers do not have complete control over their genomic data, i.e., where and by whom their DNA data is being observed after it has been sequenced. After sequenced, donors need to trust their data is secured by the sequencing institution. As more and more privacy issues and genomic-aimed privacy attacks are taking place, customers have been more and more concerned about the security of their DNA data. Currently, genetic testing companies keep the DNA data of their customers to themselves and do not transmit transparent guidelines and user rights. Data owners do not know if their samples are being reused by other researchers and what potential privacy risks they might face [Hee+11]. To illustrate that problem, a drug company very recently bought part of the 23andMe genetic testing company, meaning that this company now has access to 23andMe's customers' genome data [Jai]; Furthermore, when companies outsource their data to third-party service providers (e.g., for economic reasons), they have to entrust their data to untrusted parties, which is not ideal. Another critical fact is that DNA data cannot be simply revoked, i.e., an individual's genome cannot be merely canceled or blocked in case of a genomic leak [Aze18]. Also, DNA data usually does not change over time. Therefore, genome data also needs

to be protected over a long period of time. Finally, in order to follow and respect current privacy regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) [Hea] for the US and the General Data Protection Regulation (GDPR) [Par] for the EU; genome repositories should be able to assure that privacy regulations are being enforced in a transparent and trustworthy manner, which unfortunately is not still the case.

2.2 Genome-wide Association Study (GWAS)

A Genome-wide Association Study (GWAS) is an observational study that aims at identifying associations between specific genome variations (usually SNPs) with a particular phenotype, e.g., a disease. Therefore, by locating such correlations, GWAS can help the development of sooner diagnosis and treatment for diseases, for example. In particular, GWAS produces correlation statistics over a cohort of genomes to find high-correlated relations (genetic markers) between variant positions (SNPs) and a given trait. GWAS considers two populations, (i) the case group, which corresponds to the cohort of individuals that express the phenotype, e.g., presents a given disease; and (ii) the control group, which consists of healthy individuals.

Since any two genotypes of a person (i.e., a complete set of genes) are mostly identical, one can represent an individual's genotype by the difference in its information compared to a human reference genome [Zoo+16], usually SNPs. Hence, GWAS investigates a particular set of SNP to discover non-random connections. As introduced before, a SNP usually has two alleles that can be major or minor. Major alleles are commonly represented by 0's, while minor alleles by 1's. Therefore, assuming the genotype notation, the value of a SNP $\in \{0, 1, 2\}$, which informs the number of minor alleles found in a specific genetic position. In the case of "0", it means the presence of two major alleles, i.e., a major homozygous genotype. Intuitively, a value of "1" means the presence of a major and a minor allele, i.e., a heterozygous genotype, while a value of "2" represents a minor homozygous genotype with two minor alleles. Similarly, in the haploid notation (single-strand), the value of a SNP $\in \{0, 1\}$, and follows the same rationale presented before.

When conducting a GWAS, the set of SNPs belonging to specific individuals is encoded and used for statistical analysis. Table 2.1 provides an example of N haploid genomes $g_n \in \{g_1, \ldots, g_N\}$ that are described over L variants $\mathbf{SNP}_L \in \{\mathbf{SNP}_1, \cdots, \mathbf{SNP}_L\}$. For each SNP, an "1" in this record represents the fact that the associated genome contains a minor allele, "0" otherwise. Therefore, SNPs can be valued by the presence (or not) of the minor allele. For simplicity's sake, Table 2.1 illustrates the haploid notation (single-strand) to represent the allele sequences. Recall the genotype notation illustrated in Figure 2.1 if necessary.

Table 2.1: Genome encoding for GWAS.

	SNP_1	SNP_2	...	SNP_L	Phenotype_p
	A/T	C/G	...	A/T	Case/Control
Genome g_1	0	1		1	Case
Genome g_2	1	1		1	Control
⋮				⋮	⋮
Genome g_N	0	1		0	Case

Let us understand how genomes are encoded using Table 2.1. At a specific position of the human genome, a particular nucleotide is the most common. Let us assume it is the case of nucleotide A for $\textbf{SNP}_1$. However, in a minority of individuals, this position is occupied by the nucleotide T. Hence, it means that there are two possible alleles for $\textbf{SNP}_1$ (A or T). In consequence, in $\textbf{SNP}_1$ column at Table 2.1, every individual with allele A has the most common allele (i.e., a major allele, represented by 0s). On the other hand, individuals with allele T (the minor allele) are represented with 1s. The phenotype column identifies the concerned phenotype of the study, e.g., diabetes or lung cancer. This column also serves to label from which group a genome/individual belongs, i.e., to the case or control population.

Table 2.2: A singlewise contingency table for phenotype p.

		Phenotype_p		
		Population		
		Case	Control	Total
SNP_l	0 (major)	N_0^{case}	$N_0^{control}$	N_0
	1 (minor)	N_1^{case}	$N_1^{control}$	N_1
	Total	N^{case}	$N^{control}$	N_T

Generally, GWAS data is represented by contingency tables that summarize the information about the concerned phenotype of the study, the populations, the major and minor alleles of SNPs, and their corresponding counts over the groups. Table 2.2 shows an example of a singlewise contingency table for a given GWAS of phenotype p. Similarly, Table 2.3 presents a pairwise contingency table. There are two types of GWAS statistics: test and aggregate statistics. The following sections detail each one of these statistics.

Table 2.3: A GWAS pairwise contingency table for two variants. SNP_i and SNP_j, where $i, j \in \{1, \cdots, L\}$.

	Phenotype$_p$					
	SNP_j			SNP_j		
SNP_i	0	1	Total	0	1	Total
0	C_{00}^{case}	C_{01}^{case}	C_{0-}^{case}	$C_{00}^{control}$	$C_{01}^{control}$	$C_{0-}^{control}$
1	C_{10}^{case}	C_{11}^{case}	C_{1-}^{case}	$C_{10}^{control}$	$C_{11}^{control}$	$C_{1-}^{control}$
Total	C_{-0}^{case}	C_{-1}^{case}	$2N^{case}$	$C_{-0}^{control}$	$C_{-1}^{control}$	$2N^{control}$

2.3 GWAS aggregate statistics

The output of GWAS aggregate statistics consists of singlewise allele frequencies, pairwise allele frequencies, and minor allele frequencies (MAF) over the population. These statistics are jointly computed over the control and case cohorts of individuals. A singlewise contingency table (see Table 2.2) directly outputs the single allele frequencies associated to a given variant $\textbf{SNP}_l$, where N_i^{pop} is the count of allele $i \in \{0, 1\}$ in population $pop \in \{case, control\}$. N^{case} and $N^{control}$ are, respectively, the size of the case and the control population. N_0 and N_1 are the overall counts of major and minor alleles, respectively. The MAF is the frequency of the least common allele of a SNP in a population, e.g., $\frac{N_1^{case}}{N^{case}}$ for the case population, and $\frac{N_1^{control}}{N^{control}}$ for the control population. Similarly, the MAF for both populations is given by $\frac{N_1}{N_T}$, where $N_T = N^{case} + N^{control}$, i.e., the sum of both populations' size.

Table 2.3 illustrates a pairwise contingency table of two SNPs, $\textbf{SNP}_i$ and $\textbf{SNP}_j \in L$. C_{ij}^{pop} reports the number of occurrences of the four possible combinations of alleles $\{00, 01, 10, 11\}$ in a population $pop \in \{case, control\}$.

2.4 GWAS test statistics

The output of GWAS test statistics consists of *chi*-squares (χ^2), *r*-square (r^2), and their corresponding *p*-values of the most significant SNPs. The χ^2 hypothesis test determines whether or not to reject the null hypothesis, which states that the allele frequencies in the case and control populations follow a similar distribution. The χ^2 statistic of a single SNP is defined as:

$$\chi^2 = \sum_{i \in \{0,1\}} \frac{\left(N_i^{case} - N_i^{control}\right)^2}{N_i^{control}}$$

From the χ^2 statistic, one can then compute the *p*-value of each SNP, which is the probability of observing its contingency table should the null hypothesis

be correct with respect to some significance level α. In other words, p-values on χ^2 quantify the chances of falsely rejecting the null hypothesis while the null hypothesis is true. Hence, such a statistic allows us to measure how likely a genetic marker association with a putative phenotype is due to randomness. If a p-value is smaller than a given threshold (i.e., 10^{-8}) [Bar+12], then it indicates that the variant might be significant [Che+21].

In possession of a rank of highly associated SNPs, researchers usually want to observe how these genetic variants are correlated among them, i.e., identifying if their co-occurrences are truly random or not, by computing their Linkage Disequilibrium (LD). LD identifies associations between high-ranked SNPs within a given genetic locus. For example, alleles in the same chromosome and close to each other are commonly very dependent, i.e., they express a high linkage disequilibrium. In particular, LD is an important metric that indicates several evolutionary events, such as local adaptation, geographical structure, and chromosomal inversions of genomic data, and therefore LD is of utmost importance for understanding genomic studies [Kem+15].

LD is calculated from the pairwise allele frequencies between two SNPs. The value of this metric is determined by the results of the r^2 statistics and/or D', the former is defined as:

$$r^2 = \frac{(C_{00}^{l_1,l_2} \cdot C_{11}^{l_1,l_2} - C_{01}^{l_1,l_2} \cdot C_{10}^{l_1,l_2})^2}{C_{0-}^{l_1,l_2} \cdot C_{1-}^{l_1,l_2} \cdot C_{-0}^{l_1,l_2} \cdot C_{-1}^{l_1,l_2}}$$

while the latter is computed by:

$$D' = \frac{C_{00}^{l_1,l_2}}{2N^{pop}} - \left(\frac{C_{0-}^{l_1,l_2}}{C_{0-}^{l_1,l_2} + C_{1-}^{l_1,l_2}} * \frac{C_{-0}^{l_1,l_2}}{C_{-0}^{l_1,l_2} + C_{-1}^{l_1,l_2}} \right),$$

where l_i, $l_j \in \{1, ..., L\}$, i.e., any two SNPs in the SNP-set L. Similar to χ^2, P-values on r^2 or D' are computed to quantify the significance of the tests.

2.5 Towards large-scale genomics

The rapidly decreasing costs for sequencing human-genome data has accelerated human genome data generation and, consequently, its broader sharing. This phenomenon is not found only in the genomic research field but also in the private/industry sector. There are now various private genetic testing companies offering their DNA testing and analysis services directly to end customers. In fact, more and more individuals are deliberately sending their DNA samples for recreational purposes, such as ancestry and genealogy tests, and performance-enhancing hacks

based on DNA. Companies like 23andMe [2], MyHeritage [3] and CrossDNA [4] are openly offering such services with affordable prices.

Since the creation of the Human Genome Project (HGP) [Ins], an international program that aimed to uncover the complete set of genes and DNA bases of humans, genomic data has been shared at a higher pace [Hee+11]. The HGP project has also influenced funding and governmental institutions to sponsor and promote genomic research. Since then, several new programs have been created to enable a broader availability of genomic data and (open) access to research findings from studies. For instance, the 1,000 Genomes Project (1,000 GP) [Con+15a], which was one of the first open-access genome datasets, consisted of approximately 2,500 genomes from different populations around the world. Later, the 100,000 GP project [Eng16] was released in England with the intention to carry out larger studies relying on mostly sequencing patients' relatives. Likewise, the International HapMap Consortium project (HapMap) [Gib+03] concentrates on studying the characteristics of millions of sequence variants from heterogeneous populations with different ancestries (mainly Europeans).

Similarly, the UK Biobank initiative [Byc+18] offers access to phenotype and genotype data from the UK National Health Service (NHS) [5] to not only universities but also independent researchers, private and public companies. Additionally, other organizations from all around the world have been putting efforts into allowing ampler genomic studies. For instance, the Genetic Association Information Network (GAIN) [Man+07] and the Database of Genotypes and Phenotypes (dbGAP) [Wal+11] are well-known genomic data repositories. dbGAP has playing an important role for the proliferation of genomic studies. In particular, it stores and distributes genome data that are used by researches to conduct GWAS [Wal+11].

The creation of these projects undoubtedly helped the spread and the rate of discoveries from GWAS. In addition, these genome datasets are constantly used as use-case in privacy-enhancing technologies contests, such as iDash [6], an annual competition to raise awareness of genomics privacy while enabling efficient systems and availability of study results. In such a competition, researchers are challenged to offer new solutions for a variety of tasks, such as the privacy-preserving reading of DNA data and solutions for improving individuals' access control over their genomic data.

[2] https://www.23andme.com/en-int/
[3] https://www.myheritage.com/
[4] https://crossdna.com/en/
[5] https://www.nhs.uk/
[6] http://www.humangenomeprivacy.org/[place_year]/

2.6 Towards collaborative and practical GWAS environments

Along with cheaper costs for sequencing human-genome data and broader access to such data, the possibility to create cooperative (federated) systems where a large amount of genomic data from around the world can be collected and put together has become true. Indeed, such an environment enables more significant findings that can revolutionize genomics research. It is a clear advantage since conducting GWAS over larger datasets (by combining genome data from multiple data holders) increases the statistical power and the confidence in the findings [Fer+17]. In addition, relying on heterogeneous data from worldwide institutions helps to avoid erroneous conclusions from biased statistical findings (tailored with limited genome data availability). In particular, certain genetic variations linked to some particular traits (i.e., phenotypes that are more common in specific ancestries) might unexpectedly influence GWAS associations [Con+15b; Sad+18]. Therefore, performing GWAS relying on heterogeneous genomic datasets might produce more reliable findings by alleviating the over-representation of some populations in studies [Ost+21; SWT19]. For these reasons, the mindset of conducting genomic studies has rapidly changed. Indeed, instead of producing research studies alone, biocenters are now shifting towards a larger scale setting, conducting GWAS on a global scale [VB13; Bes+15]. Therefore, federated GWAS is itself a valuable and beneficial mechanism. In such a model, there is a cross-institutional collaboration among biocenters, institutes, or any genomic data holders to distributively aggregate and analyze GWAS.

Since access to genomic data repositories has become more accessible, several works have discussed the trade-offs of genomic data privacy and sharing in genomics research [VG16; Kay12]. For instance, keeping genome data confidentiality and integrity are essential properties from the moment such private goes beyond the premises of data holders. Some works have envisioned and offered architectures to enable such collaborative systems, where several data holders would store, share and distributively process genomic data [VB13; Rai+18; Men+19; Bla+18]. Nevertheless, as one would expect, given that these systems manage critical and sensitive data, there is an utmost need to protect genomic data at both individuals' and institutions' levels. Hence, there is a need for solutions that protect the private genome data of donors from genomic privacy attacks and, at the same time, enforce secure management and processing of such data.

In addition, there is pressure to enable individuals to control how their data is being used [Dan+20; Dec+18]. In particular, to comply with current data-privacy regulations' constraints such as HIPAA and GDPR, which demands that data subjects shall have the right to withdraw their consent to participate in a GWAS

at any time. In fact, when individuals' DNA is sequenced, they usually have to provide a "broad consent" that authorizes institutions to perform any type of processing over the data [Tka+18], which might decrease the number of individuals willing to voluntarily participate in such studies if no privacy-preserving and secure mechanisms are guaranteed to enforced by the GWAS federation.

Figure 2.2 depicts both stand-alone and federated GWAS settings. The right side of Figure 2.2 illustrates a standard federated GWAS setting. Each data holder (e.g., a biocenter institution) holds genomic data sequenced from several individuals and collaboratively outsources their data to conduct federated GWAS analyses.

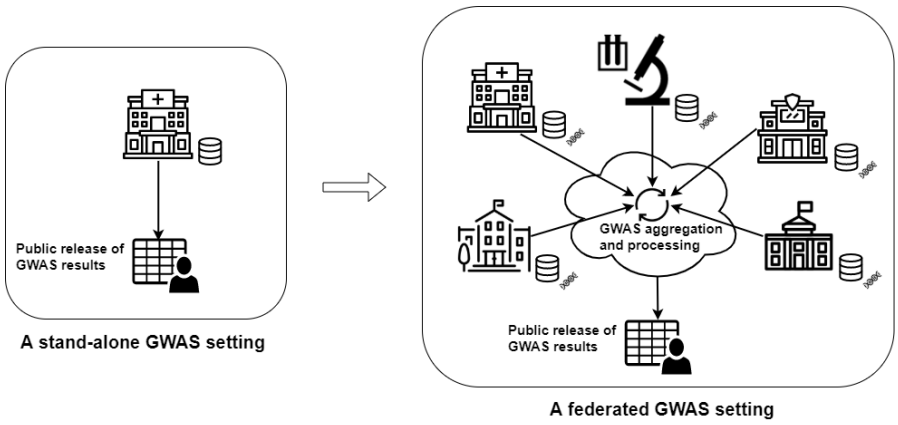


Figure 2.2: Moving from a stand-alone to a federated GWAS setting.

As introduced before, new factors and issues are raised when considering a federated GWAS scenario, which this thesis aims to address simultaneously. Furthermore, this work introduces the term *practical GWAS* that encompasses some novel features required for federated GWAS to comply with 21st-century privacy guidelines. The properties and features of *practical GWAS* are categorized next:

- **Secure outsourcing and privacy-preserving *processing*:** All genome data that is outsourced by institutions in a GWAS federation needs to be aggregated and processed in a safe and secure manner to avoid any type of privacy breaches. To ensure such functionalities, federations usually rely on privacy-preserving approaches, such as Secure Multiparty Computation (SMC), Homomorphic Encryption (HE), Differential Privacy (DP), and the use of Trusted Execution Environments (TEE). These approaches allow secure and private GWAS computation. However, each method has its advantages and drawbacks that need to be assessed according to the federations' expectations to choose the more suitable scheme. For instance, some approaches achieve poorer computational and storage performance when assuming an increased number of participants, while others suffer from decreased accuracy. Later,

this section describes in detail and compares the existing privacy-preserving approaches for federated GWAS. Therefore, choosing an efficient and doable approach is vital when designing federated GWAS.

- **Privacy-preserving *releasing*:** Although ensuring privacy-preserving outsourcing and processing of genome data is a crucial step towards *practical GWAS*, it is not enough. Few years ago, there was a belief that publicly publishing aggregated statistics was safe [Edi08; Cou08; Hee+11]. Nevertheless, it has been shown that the release of GWAS statistics can be subject to genomic privacy attacks. Indeed, several privacy attacks leveraging GWAS statistics demonstrated that GWAS results need special care before publication [Wan+09; Hom+08; Jac+09; Im+12; Cai+15]. Following the publication of these attacks, the NIH preventively removed all GWAS results from public access and instantiated an approval process one must follow to consult them [ZN08]. Unfortunately, such a restriction diminishes the spread of GWAS releases and slows down its benefits to the research community and society [BSB20]. Therefore, *practical GWAS* should enable safe and open-access releases of GWAS to the masses.

- **Collusion-tolerance:** To improve the level of the privacy guarantees when conducting federated analysis, the designing of federated GWAS should ideally consider the presence of Honest-but-Curious (HbC) institutions or even the presence of an adversary able to control one or more federation members and make them behave honest-but-curiously [Pas+21]. When facing such an adversary, the federated protocol and privacy-protecting algorithms need to enforce that both the distributed processing and the releasing of GWAS is processed in a collusion-tolerant manner. Indeed, one of the findings of this thesis is that a HbC adversary controlling several biocenters and monitoring the GWAS releases could combine data from specific institutions to isolate sufficiently small data of honest institutions, and therefore being able to leak their private data. As a result, HbC parties can circumvent the protection provided by existing privacy-preserving solutions that do not consider collusion among members. Therefore, *practical GWAS* should enforce that the data of each biocenter is secured (i.e., used to produce results but not revealed if facing collusion between unethical parties).

- **Dynamism (dynamic GWAS setting):** Coupled with cheaper costs for sequencing human-genome data, GWAS could benefit from a dynamic scheme where its results are updated as soon as new individuals are sequenced and added to a study. In addition, there is pressure to enable individuals to control how their data are used [Dan+20; Dec+18], and therefore allowing individuals to request removal from studies. In particular, to comply with

current data-privacy regulations' constraints such as the US HIPAA the EU GDPR, data subjects shall have the right to withdraw their consent to participate in a GWAS at any time [Dan+20], which demand dynamic privacy-aware approaches. Enforcing privacy in this dynamic context is challenging since a potential adversary having access to several GWAS result releases could leverage the evolution of the results to infer data. Therefore, *practical GWAS* should ensure that genome additions and removals from existing GWAS are performed in a secure and private fashion.

- **Interdependent GWASes:** In real-life settings, GWASes might consider overlapping sets of individuals, each GWAS focusing on a specific set of genomic variations, some of which might also be used in other studies for economic reasons. In fact, it is rather likely that federations will run different GWASes simultaneously (e.g., one on diabetes and a second studying lung cancer [Den+20]). Furthermore, as later presented in this thesis, new means to breach genomic privacy arise from the fact that more and more genotype-phenotype data and GWAS releases are available. Therefore, cross-referencing multiple studies might lead to privacy leaks [Gür+18]. Consequently, an adversary can base its attack on the results of a single multi-trait study or even from GWASes from multiple federations. Therefore, GWASes should be released only after carefully considering the interdependency among studies in the federation so that the efficacy of privacy-preserving release mechanisms continue satisfactorily.

To the best of the author's knowledge, this thesis is the first work that introduces and offers mechanisms to cope with the new issues raised to support the context of *practical GWAS* introduced above.

2.7 Privacy-preserving *processing* of GWAS

As presented before, due to its high sensitivity, genomic data must be managed and operated following the best practices of privacy and security. Otherwise, potential volunteers and/or institutions would not feel comfortable sharing their genomic data for collaborative genomic research. Thankfully, the creation and advance of cryptography-based primitives that enable privacy-preserving data processing have been employed as an alternative to mitigate some privacy risks, mainly when enforcing data protection under federated analyses. In particular, existing approaches rely on cryptographic primitives to protect the integrity and confidentiality of data and hence the privacy of both data holders and individuals when their data are shipped and processed. Nevertheless, these non-functional benefits come at a price. In fact, each one of the current approaches suffer from some type

of limitation. For instance, enhanced computational/network resources demand, lower data accuracy (from unexpected noise tailored to cryptographic operations or from noise-based release protection). These issues might impact the overall performance of federated GWAS systems, and therefore choosing the most suitable solution is not intuitive and highly depends on the GWAS federation goals and expectations.

In particular, existing privacy-preserving approaches are organized in several classes depending on the nature of their mechanism. The following describes the main privacy-preserving techniques adopted in the literature for the privacy-preserving *processing* of data.

- Secure Multiparty Computation (SMC) aims to enable secure outsourcing and collaboration among several parties. In particular, it allows each party to privately share their part of their data to compute a given function over aggregated inputs without the presence of a trusted party, whereas protecting private data. Entities privately share their inputs $(x_1, x_2, ..., x_n)$ and compute the result of a common function $f(x_1, x_2, ..., x_n)$ without revealing or disclosing parties' private share to others. It was first proposed by Yao et al. [Yao82], which presented an approach based on garbled circuits over boolean operations. Secret Sharing (SS), such as Shamir's secret sharing [Sha79] is another type of SMC approach. SS is a scheme used in cryptographic protocols that enables the distribution of private inputs by each party, which can only be reconstructed if a sufficient number of secrets are retrieved together. In addition, each isolate share does not reveal useful data for any party. One of the main drawbacks of SMC approaches is their increased computational overhead and design complexity, which indeed needs some adaptations to allow the execution of specific tasks. Moreover, SMC presents limited scalability as its performance decreases with the number of parties, which limits its use, flexibility, and practicality [Che+16b].

- Homomorphic Encryption (HE): The main goal of HE is to allow arithmetic operations over encrypted data [Gen09]. In summary, HE enforces that the output of a function over plain text data is the same as if it were to be performed over two encrypted files containing the same information as in the plain texts. The benefits of using HE is straightforward. Indeed, performing operations on encrypted data is more secure and keeps a higher level of security and data privacy. Furthermore, only the players in possession of the correct keys will be able to decrypt and read the final output. Therefore, HE is able to protect not only the inputs of the data holders but also limit access to the final output of the desired computation. Nevertheless, such advantages come with some performance costs. Regrettably, only a limited set

of arithmetic operations can be computed over homomorphically encrypted data, such as addition and multiplication, which limits its adoption when performing GWAS (because it demands the computation of more complex statistics). However, it has been shown that fully homomorphic encryption can perform more complex computations over encrypted data but it exhibits even higher storage and computational resources overhead, limiting its usability. Moreover, HE might be subject to cipher-blow-up issues [MC19].

- Differential Privacy (DP): DP [Dwo11] is a data perturbation-based mechanism that provably (mathematically) guarantees the privacy of each record when statistical data computed over a given data set is released. It provides a privacy gain method that ensures that the removal or addition of any single record from a data set does not compromise the privacy of any other record. It is achieved by computing a probabilistic metric of privacy and applying random noise to data so that the identification of any subject is not possible. There are several versions of DP depending on the type of perturbation added. The most common approach uses the Laplace distribution [Dwo+10; DP13]. More formally, let D and D' be two neighboring datasets that differ by a single element, and let O be the set of all possible outputs of a query. A release R is ϵ-differentially private if:

$$Pr[(R(D) \in O] \leq exp(\epsilon) \times Pr[R(D') \in S], \tag{2.1}$$

  where ϵ is the privacy parameter that determines the level of privacy protection that comes as a random noise added to the outputs in O. When leveraging the Laplace mechanism with l_1 sensitivity level of a function f defined as $\Delta_f = \max_{D,D'} ||f(D) - f(D')||$ (which portraits the largest change in f when a single record is replaced) [DR+14], the applied noise is derived from the Laplace distribution with mean 0 and scale $\frac{\Delta_f}{\epsilon}$.

  In particular, as the probabilities differ by a factor of ϵ, DP has a privacy and data utility trade-off. Intuitively, a smaller ϵ means stronger privacy with lower accuracy [DP13]. Such mechanism is defined by as *privacy budget*.

  Therefore, DP can be used to protect the individual inputs of federation members (i.e., their local genome data) before it is outsourced to the federation. Such a feature is achieved using DP at a local level (local DP). DP can also be used to protect the genome sequences participating in a study (ensuring global DP for the protection of public releases). Nevertheless, DP suffers from the loss of data utility given its noise-based nature, which leads to less accurate outputs.

- Trusted Execution Environments (TEEs): TEE assures confidentiality and

integrity of the data being processed in a processor's secure area. It leverages a set of operations into trusted zones of processors that enable protection to the code and data managed inside the trusted area, namely an enclave. One of the most common implementations of TEEs is Software Guard Extensions [CD16] by Intel. SGX defines the concept of the enclave as an isolated unit of data and code execution that cannot be accessed even by privileged code (e.g., from the operating system or hypervisor). Enclaves can be attested to prove that the code running in the enclave is the one intended and that it is running on a genuine Intel SGX platform. Once attested, enclaves can be provisioned with secret data by using authenticated secure channels. Moreover, enclaves can persist confidential data outside the trusted zone by using a sealing mechanism. By relying on TEE, it is expected to reduce the computational complexity and restrictions of the other cryptography-based approaches. For example, TEE does not suffer from running only limited types of operations and demands less complex designs, which increases its communication and computation efficiency. Nonetheless, current TEE implementations still suffer from a limited amount of memory in their secure regions (128 MB - of which only 96 MB is usable without paging [CD16]). In addition, it has been shown that it is vulnerable to side-channel attacks [Bra+17].

Table 2.4 summarizes and compares the privacy-preserving *processing* approaches for securely processing data in collaborative environments. Following the same idea, Table 2.5 presents a performance overhead discussion of each one of approaches. The conclusions are made from an performance analysis of existing works.

In summary, when compared to the other cryptographic methods, TEEs are significantly faster and admit a larger set of operations (not just arithmetic operations) [Che+16b]. In addition, although SMC and HE techniques allow privacy-preserving computation, they lack scalability and need domain-specific adaptations [Che+16b; Zha+15]. In contrast, TEE inherits fewer issues as its overall framework is able to facilitate the secure sharing of data. Moreover, its lightweight cryptographic methods offer a more cost-effective model. Besides, when using Intel SGX, for example, both application code and computation are protected from any interference from outside of the dedicated secure area. Hence, guaranteeing data confidentiality and integrity at the same time. Yet important, TEEs do not suffer from accuracy loss, such as local DP-based approaches.

This thesis leverages TEE as the main component to achieve privacy-preserving *processing* of federated GWAS and also for the verification of private releases. TEE was chosen because it was identified as the most fittable approach in terms of performance, accuracy, and efficiency trade-off when compared to existing privacy-

Table 2.4: Overall comparison of privacy-preserving *processing* approaches.

Approach	Hardware requirements	Type of operations	Limitations/Risks
DP (local DP)	Any CPU	Any	Decreased output accuracy that impacts data utility. Vulnerable to collusion attacks [Eig+14] and highly depends on the statistical independence of records in a data set [LCM16].
HE	Any CPU	Addition and multiplication	Limited number of operations and cipher-blow-up issues. Reaction attacks on fully HE protocols [ZPS11].
SMC	Any CPU	Boolean	Task-based designing is required and not easy to scale. Not secure against malicious adversaries [Yao82].
TEE	Isolated cryptographic-based processor	Any	Limited amount of memory [CD16] and vulnerable to side-channel based attacks [Bra+17].

preserving solutions (as perceived by the discussion in this section). Therefore, the next section presents TEE in more detail.

2.7.1 Trusted Execution Environments (TEE)

This section presents TEE's main concepts, and more specifically on Software Guard Extensions (SGX), the TEE-based solution provided by Intel Corporation [CD16]. Our solutions rely on Intel SGX [McK+16] as a vehicle for our implementation without relying on any specific feature of SGX. Our choice for SGX is motivated by previous works leveraging this technology and its increased availability in cloud services [Pas+21; BAZ20; Koc+19]. However, our solutions apply equally well to other TEE implementations.

The development of embedded hardware and secure cryptographic co-processors has evolved rapidly in the last few years. Thanks to these advancements, various TEE-based technologies are now available to the general community. As a re-

Table 2.5: Overall comparison of the performance of privacy-preserving *processing* approaches.

Approach	**Communication costs**	**Computational costs**	**Storage costs**	**Output accuracy**
DP	Low	Medium	Medium	Low
HE	Medium	High	High	Medium
SMC	High	Medium	Medium	Medium
TEE	Low	Low	Low	High

sult, the most significant processor manufacturing companies have developed their own TEE-based solutions. For example, in the form of Trusted Platform Module (TPM) [Kin06], virtualization (AMD Secure Virtual Machine (SVM) [Van06], and hardware-enforced isolation in CPUs, such as ARM Trust Zone [PS19] and Intel Software Guard Extensions (Intel SGX) [CD16]. Given its popularity, availability to the general community, and adoption, this thesis uses SGX as the means to leverage TEE's benefits. In the following, the main concepts of Intel SGX are discussed.

Intel SGX is a collection of x86-64 instruction extensions that provide application code and data with hardware-based memory encryption and isolation. The protected memory region (also known as an enclave) is located in the address space of a program and provides confidentiality and integrity protection. Software and code residing inside enclaves are protected. They cannot be tampered with thanks to memory encryption mechanisms and isolated execution generated within the enclave's boundaries. In particular, an enclave's memory is mapped to the Enclave Page Cache (EPC), which is a unique physical memory region. The Memory Encryption Engine (MEE) is responsible for encrypting data in EPC, making it inaccessible to other system applications (even the host OS, system BIOS, and processes of other enclaves).

One crucial characteristic of SGX is that all messages exchanged between the enclave and the CPU cache are encrypted. Thus, SGX's trusted computing base (TCB) can only involve the trusted code inside the enclave and the processor itself. To enforce trust and security, SGX uses standard cryptographic primitives along with three main functionalities that play an important role when (i) inputting/outsourcing data to enclaves, (ii) ensuring that only certified/trusted code will process the data, and (iii) protecting data outside the secured region.

To achieve the above goals, SGX assumes the use of known encryption methods in its design. Namely, it uses Advanced Encryption Standard (AES) to perform authenticated encryption to provide data authenticity and confidentiality, Ellip-

tic Curve Diffie-Hellman (ECDH) to enable secure sharing of shared symmetric keys for AES on insecure channels, and Elliptic Curve Digital Signature (ECDS) algorithm to sign and verify data authenticity/integrity. When the data are decrypted inside an SGX enclave, only authorized and authenticated code can access the data. This is enforced by hardware-supported access control mechanisms that certify that any component of the hosting system (e.g., malicious software or operating system) can modify or access data inside the enclave.

The SGX suite consist of the following primitives:

- **Remote attestation:** This is the method that clients can use to certify that an application is being executed inside an authenticated enclave and running correct and trusted code [CD16; McK+16]. This verification is performed by allowing a remote machine (the client) to determine the level of integrity of the other platform (the SGX enclave). Usually, it is accomplished by leveraging cryptography signature schemes to allow clients to verify the hash of enclaves' content/code/information. If this process does not succeed (i.e., if the enclave's hash does not match the expected hash), a client would refrain from sending data or relying on such an enclave.

  Usually, there are three parties involved during Intel SGX's attestation process, (i) the Independent Software Vendor (ISV), e.g., the one providing the source code and who also needs to be registered at Intel as the recognized code developer; (ii) the Intel Attestation Service (IAS), which hosted by Intel and is responsible for verifying authenticity, confidentiality, and integrity of the enclave; and (iii) the SGX service platforms, i.e., the service provider that is hosting the SGX-enabled machine, usually a cloud service such as Azure Confidential Computing [7].

  The attestation process starts with the ISV issuing an attestation request challenge, which might be produced by an enclave user who wishes to complete the enclave's attestation. The attested enclave then creates a verification report, which includes the enclave measurement and is verified through local attestation by a specific enclave signed by Intel called quoting enclave (QE). Once the QE signs the report using the attestation key, the generated quote is then sent to the IAS. Finally, if the quote can be successfully verified by the IAS, it signs the verification result using an Intel secret key. Such a file can be used by the enclave's clients and/or ISV to check the authenticity of the enclave.

- **Secure data outsourcing:** When the remote attestation process succeeds, it means that the client has attested the integrity of the enclave. Therefore,

[7] https://azure.microsoft.com/en-us/solutions/confidential-compute/

clients can trust and upload their privacy-sensitive data to the enclave to be processed in a secure fashion. Clients' data is sent through a secure channel established with the enclave that also allows future communications. Additionally, the outsourced data is usually sent in an encrypted manner that only the enclave and the client can decrypt. In addition, encrypted data can be safely held outside enclaves' premises. For instance, stored in an untrusted third-party service provider. The enclave can retrieve such data at any time.

- **Data sealing:** As introduced before, enclaves have limited space. Therefore, it is needed a method to enable the retrieval of data outside the enclave in a secure and long-term manner. This is achieved by the data sealing process that encrypts and stores data in a particular manner such that the enclave is the only component that is able to "seal"/encrypt and "unseal"/decrypt it [CD16]. More specifically, the data residing in the enclave is encrypted using an encryption key generated by SGX's CPU hardware. In particular, each enclave has its unique key that is used to encrypt and securely store data outside the enclave's boundaries. When retrieving "sealed" data from outside, that same key is used to decrypt it already inside the enclave. In addition, this process is also used because every time an enclave exits to the host OS, all the data inside that enclave is destroyed. As a result, if an enclave needs to access the data again at a later point, it also needs to use the sealing mechanisms. Last but not least, data sealing is commonly also used as a means to increase the scalability of TEE-based solutions by allowing the retrieval of data at later stages or future steps of an algorithm running inside an enclave [CD16; BAZ20; Pas+21].

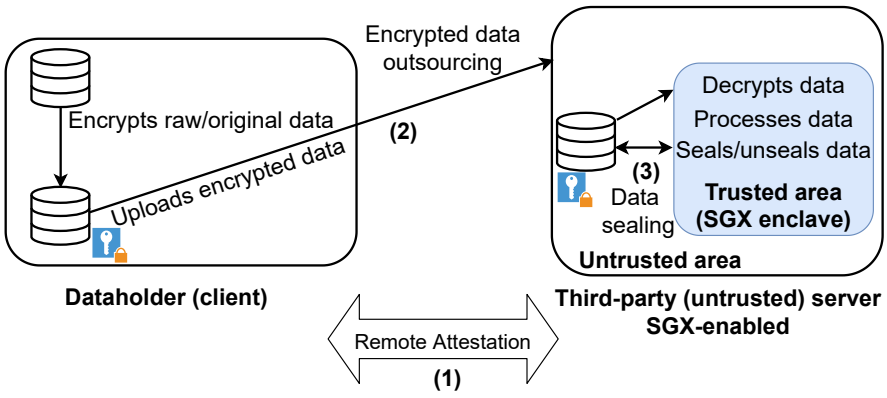


Figure 2.3: A typical setting of SGX-based solutions.

Figure 2.3 illustrates the default setting of SGX-based approaches. In a regular SGX pipeline, client machines (the entities sharing private data) should first

conclude the remote attestation process (step (1) in the figure) with the enclave being hosted in a third-party (untrusted) server. Once this step is performed, the encrypted data can be securely shared and processed (step (2)). Indeed, even even if the third-party server behaves maliciously, it will not be able to decrypt or forge the data (if it is tampered with, the enclave will notice that the data has been modified while decrypting it). In addition, the data is only accessible to the trusted area inside the enclave. Once private data is moved inside the enclave, the enclave can decrypt and securely operate on the data. If the data is needed at a later point of the application, it can be sealed and stored again in the untrusted area of the server and then recovered later by the enclave (step (3)). Finally, the enclave can send back the outputs of its application to the client (using cryptography methods again to make sure that only authorized clients can access it). However, as this thesis assumes public releases of GWAS statistics results, our solutions do not apply such an operation, i.e., the GWAS results produced by the enclave are publicly shared (after proper crafting to impede genomic privacy attacks from released statistics).

Limitations of SGX-based privacy-preserving systems. TEEs and, in particular, Intel SGX have been extensively used to build secure and privacy-preserving systems. Despite presenting many advantages, SGX-based solutions also suffer from some limitations. Firstly, when relying on Intel SGX, not only the clients but also the server needs to trust the hardware manufacturer, which in this case is Intel, and also the hardware itself. Depending on the envisioned system model, such additional trust requirements might not be reasonable.

Secondly, SGX uses a particular Memory Encryption Engine (MEE) to encrypt and decrypt data inside the enclave while data is processed. The issue is that the available size of the Enclave Page Cache (EPC) memory of an enclave is only 128 MB. Furthermore, only 96 MB out of its 128 MB size can be used by applications running inside the enclave [CT18]. However, it is true that SGX enclaves are under constant development, for example, SGX 2 [McK+16] has been recently released and offers dynamic memory management and allocation within enclaves. Notwithstanding, it is crucial noticing that even though the enclave memory can be expanded to 4 GB by using software pagination mechanisms [Che+17a; Che+16b], it shows an increased performance overhead when dealing with larger data and high-load algorithms, which is the case of genomic data. Therefore, it is clear that the memory limitations of SGX may impose an obstacle when utilizing it. It is expected that the memory limitation problem of SGX will end in the near future. In particular, some recent processors are already giving support to the creation enclaves that can manipulate up to 512 GB [8].

[8] https://lenovopress.com/lp1262-intel-xeon-sp-processor-reference

Moreover, some previous works have shown the vulnerability of SGX enclaves to memory access pattern-based attacks, such as side-channels attacks [Man+18]. The majority of side-channel attacks aimed at SGX enclaves are exploiting the memory cache access of "non-oblivious" implementations of algorithms inside enclaves. Generic memory-oblivious solutions have been offered to overcome this issue, such as path RAM (PRAM) [Ste+18] and Oblivious RAM (ORAM) [Gol87], and Oblivious B+ tree shuffling [Vim+15]. More recently, some approaches to circumvent such attacks have been proposed in the literature, such as adapting the genomic workflow algorithms to work in a data-oblivious fashion (ensuring random memory access patterns) [Man+18; ZBA15; AKM20]. Nevertheless, most of these approaches are offered for general purposes and therefore end up impacting the overall performance of the system. In addition, other ad-hoc solutions, such as employing encoding techniques to fit genomic data in a certain way so that paging attacks cannot succeed, have also been offered. For example, by fitting data within 4 KB page-wise blocks [Che+17a] or processing a limited number of SNPs at a time [Che+16b].

Finally, enclaves can also be subject to Denial-of-Service (DoS) attacks [TPV17; Che+17b]. Although those attacks do not compromise privacy, they might disrupt the pipeline and the expected behavior of the application. As this vulnerability is out of the scope of the objectives of this work, it has not been addressed.

2.8 Genomic privacy attacks on GWAS releases

Privacy-preserving *processing* approaches enable safe and secure outsourcing and computation of data for federated analysis. Nevertheless, only enforcing privacy-preserving *processing* is not enough to provide a fully privacy-preserving federated system. In particular, the final output of a computation operated over aggregated data from multiple parties also needs to be protected against existing attacks once results, in our case, GWAS statistics, are published. In addition, sequencing individuals represents a financial effort, and private institutions would refrain from participating in a study if there is any risk of seeing their data being inferred by competitors.

Previous works have been shown that the simple release of GWAS (even if they are operated leveraging privacy-preserving *processing* schemes) results might be subject to privacy attacks launched by an external adversary [Wan+09; Hom+08; Jac+09; Im+12; Cai+15]. The goal of an adversary when launching a genomic privacy attack is to leak sensitive genetic information of the victim(s).

Adversaries might compromise the genomic privacy of participants in two ways. The first (traditional) way, an external adversary launches genomic privacy attacks by observing the GWAS statistics from releases. The goal is to leak confidential/se-

cret data of participating individuals. For instance, reconstructing individuals' genome sequences or inferring the participation of a particular victim in a study.

Another way to leak the private data is to collude with other (colluding) members of the federation to increase knowledge regarding the aggregated data (final result), which can reduce the needed effort and make an attack possible. In particular, this scheme allows colluders to isolate honest members' data and as a result being able to circumvent privacy-preserving ***releasing*** schemes [Pas+21]. For instance, in a recovery attack, colluding parties can share their inputs with other colluders to isolate the data of other members. Thus, reducing the complexity of the attack (since isolated data searching space is smaller). Thus, potentially succeeding on leaking data of other parties in the federation. Unfortunately, such threats have not been given the appropriate care in federated GWAS scenarios.

Last but not least, existing solutions that enables safe releases of GWAS statistics assume a static setting, i.e., they do consider that GWAS results can be updated over time and that adversaries might leverage how statistics have evolved to mount attacks. Therefore, current mechanisms cannot cope with the new privacy issues that arise when studies are updated in a dynamic fashion. Notably, next chapters present how adversaries might explore how statistics have evolved between any two releases in order to breach existing safe release conditions.

Attacks on GWAS results are classified according to the type of information the adversary aims to leak and also on how it is carried. The following sections detail the two categories of genomic privacy attacks, namely recovery and membership attacks.

2.8.1 Recovery attacks

A recovery attack aims at reconstructing the allele sequence of individuals who participated in a study (i.e., the content of the encoded genome table used for GWAS computation, recall Table 2.1). Recovery attacks are also referred as attribute inference attacks. Adversaries mount this attack leveraging GWAS metadata (e.g., number of participants and SNPs) and GWAS statistics released in studies [Zho+11; Fre+14; Ber+18; Dez+17].

In particular, a recovery attack leverages the observation of GWAS statistics (which may include one or several of the statistics introduced in Section 2.2.) released over a certain number of SNP positions (let us assume this number as L), and knowing the number of individuals that have participated in a given study (i.e., N genome pseudonyms). Possessing such data, an adversary is able to generate possible combination of genome sequences to build matrices that satisfy the information provided by the GWAS release. This scheme is presented in Figure 2.4.

Fredrikson et al. [Fre+14] present an approach to infer genotype sequences of individuals by leveraging demographic information and pharmacological data from

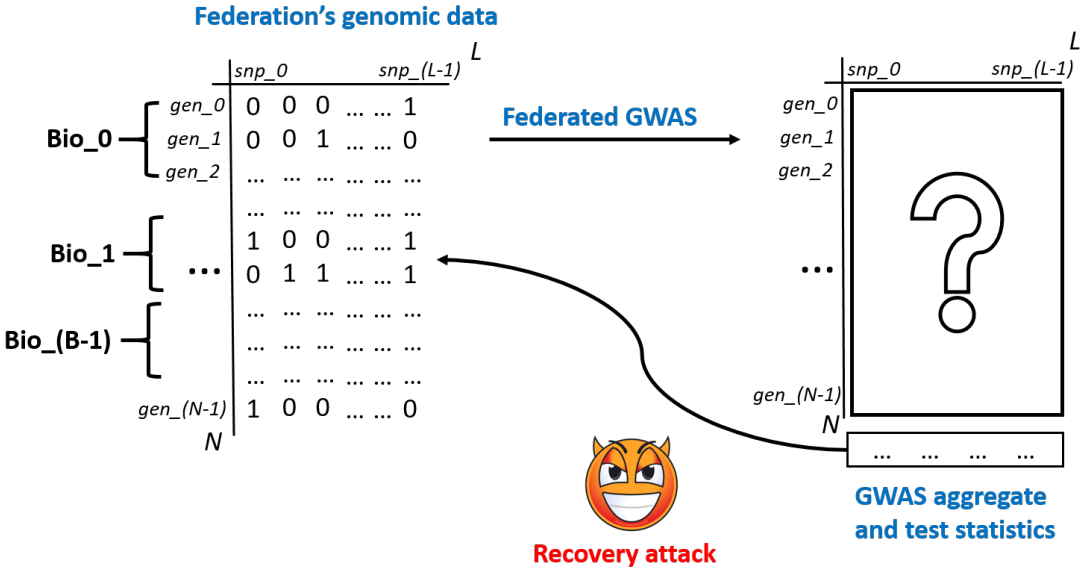


Figure 2.4: Illustration of a recovery attack on the observation of GWAS statistics.

the victim. Wang et al. [Wan+09] describe a statistical attack capable of inferring a considerable number of SNPs from individuals by observing GWAS statistics, specifically r^2 values between SNPs. Assuming the adversary has access to partial genotype data of victims, Samani et al. [Sam+15] illustrate an attack that uses public GWAS statistics among correlated SNPs, such as pair-wise allele frequencies and linkage-disequilibrium to infer unknown or hiddens SNPs of the victim. Humbert et al. [Hum+17] evaluate interdependent genomic privacy risks among individuals. In this work they show that an attacker might correctly infer genotype information of the relatives of an individual by leveraging statistical relationships among genomic variants and leveraging genotype and phenotype information of a person. In addition, Ayday et al. [AH17] illustrate a similar attack where hidden genomic data from the victims can be inferred by using partially obtained data from their relatives, such as mother and father. Similarly, He at al. [He+18] present an attack based on belief propagation in factor graphs that combines phenotype-genotype data from public GWAS in order to infer not only genotype data but also phenotype traits of the victims.

The works presented above have considered that additional side information ("external knowledge") might be observed by the adversary when launching recovery attacks, e.g., the parental relationship among genomes, demographic data, and partial access to individuals' genotype sequences). This thesis assumes the threat model of recovery attacks used in [Zho+11; Wan+09], where a probabilistic polynomial-time (p.p.t.) adversary has access to GWAS statistics data and

metadata (e.g., anonymized genome ids and SNP ids).

This thesis builds on Zhou et al. [Zho+11] conditions, where the theoretical complexity of recovery attacks are presented. The rationale behind this attack is that from the observation of GWAS statistics, an adversary is able to generate at least $\binom{2^L}{N}$ candidate matrices that matches the statistics results of the observed GWAS [Zho+11]. Out of all candidate (valid) matrices, there is a certain number of matrices that fully overlap (i.e., contains the same SNP sequences regardless their order). If the attacker can find am unique matrix out of all valid ones and that also matches the statistics results of the GWAS, she/he has successfully recovered SNP sequences of the individuals in a study. However, such a task is NP-hard [Zho+11].

Due to its reversed-engineering nature, a recovery attack demands more computational resources and running time to be launched because a huge number of combinations have to be generated and compared. On the other hand, the adversary does not need access to the real genome sequence of the victims (as most versions of recovery attacks that relies on additional background information assume).

Recovery attacks might escalate and become more dangerous. Indeed, inferred genotypes might allow the unwanted identification of subjects who participated in a specific study. For instance, after successfully reconstructing a genotype sequence of an individual in a recovery attack, the adversary can launch a membership attack to detect the participation of the concerned individual in other studies. Such an attack is explained in the following section.

2.8.2 Membership attacks

In a membership attack, an adversary aims at determining whether a genotype gen_{victim} participated in a GWAS. The attack works as follows, given the genotype sequence of a victim, the metadata, and GWAS statistics over the L SNPs, a membership attack aims at determining whether the victim belongs in the case population [Hom+08] by computing statistical tests to measure how likely a participant belongs to a study. This attack allows the adversary to link the victim with the phenotype studied, which is a serious privacy breach. A membership attack scheme is detailed in Figure 2.5.

Different variations of the attack have been implemented based on various approaches: from the Likelihood-Ratio LR-test that uses genotype frequencies [Jac+09; San+09a; VH09], leverages correlations among SNPs of the human DNA [Wan+09], such as applying Markov Chain Models [Zho+11], using Bayesian approaches [Cla10] and Belief propagation methods [He+18]. Visscher et al. [VH09] combined linear regression and LR-tests statistics to infer the presence of an individual from its genotype and MAF of a GWAS study. They show a clear correlation between the number of SNPs and individuals in the cohort for the success of the attack. Craig

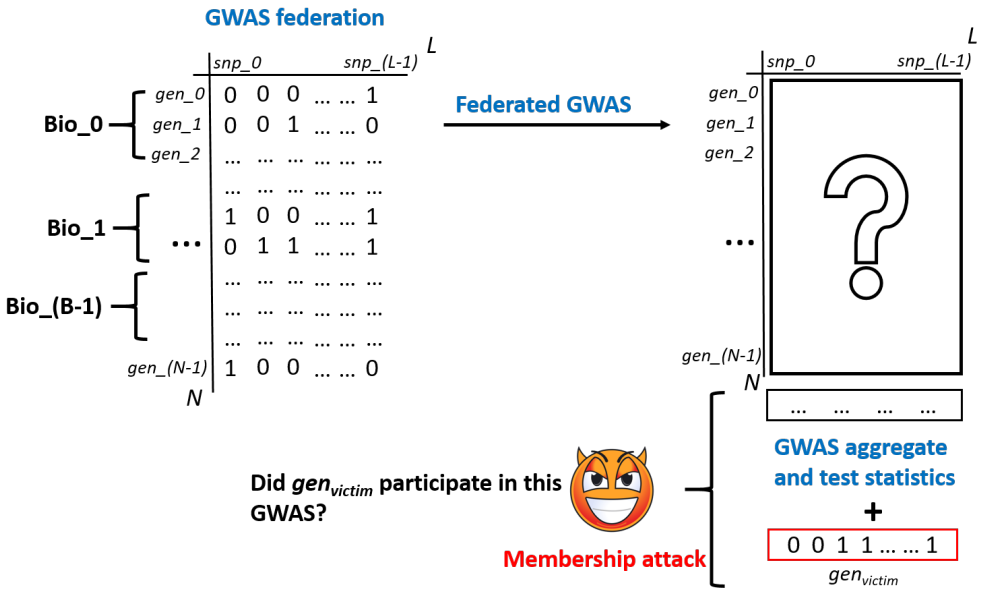


Figure 2.5: Illustration of a membership attack on the observation of GWAS statistics.

et al. advocate for an approach based on positive predictive values [Cra+11]. Im et al. [Im+12] show the feasibility of membership attacks using regression coefficients from quantitative phenotypes instead of individuals allele frequencies. More recently, [Cai+15] detailed a practical membership attack that can detect individuals from GWAS results using only 25 random SNPs. Simmons et al. argue that basing releases on the two previous approaches provide significantly weaker privacy guarantees for some individuals because the privacy measures used to decide for a release are averaged over all individuals [SB15b]. Humbert et al. [Hum+15] show the feasibility of re-identifying individuals in public genomic databases (such as OpenSNP[9]) by knowing their phenotypic traits, and then launch membership attacks using victims' genotype data to infer new traits (phenotypes).

2.9 Privacy-preserving *releasing* of GWAS

As introduced earlier, enforcing privacy-preserving data aggregation and processing for federated GWAS is not enough since the existence of genomic privacy attacks on GWAS releases might compromise the privacy of participants. Inspired by this issue, works in the literature have provided approaches to protect individuals from genomic privacy breaches while allowing the publication of GWAS

[9]2 https://opensnp.org

metadata and results.

Data anonymization approaches enforce privacy properties such as t-closeness, k-anonymity, and l-diversity. Their objective is to keep personally identifiable data protected, i.e., indistinguishable from other records in a data set [Swe02]. However, many works in the literature have been showing that such methods are not able to protect personal genomic data. In particular, k-anonymous genomic data sets have been deanonymized [SAW13; Vai+13; SBS19], even with high dimensional data [ZN+15].

Another approach used to protect releases is DP [Dwo+10]. Recall DP definition and properties explained in Section 2.7. DP can be used to ensure differential privacy by relying on the addition of noise to the final output. Nonetheless, as one could expect, such properties are kept by decreasing the accuracy of the released data. The privacy level certainty sustained by DP depends on the statistical independence of the records in a data set. Previous works have shown that highly correlated records in data sets can diminish the DP's guarantees [KM11; LCM16]. In addition, if not well designed, DP approaches can also suffer from collusion attacks, which might disclose individuals' data [Eig+14]. More recently Almadhoun et al. [AAU20a] have shown that differentially private results are vulnerable to genomic inference attacks as DP cannot not cope with dependencies of the records within a genome data set, which diminish the privacy assurances of using DP-based mechanisms.

In addition, DP under continual observation [Dwo+10; CSS11] and for growing databases [Cum+18] allow DP guarantees under dynamic scenarios (i.e., results are updated and released as more data is gathered in a federation). However, these benefits come with an increased accuracy loss due to higher levels of noise they are added over the releases, which decreases, even more, the accuracy of results. Besides that, to the best of the author's knowledge, no usable dynamic DP-based approach has been offered to allow safe updates of GWAS.

To avoid the use of noise-based techniques, several works adopt a different strategy to protect GWAS releases against *recovery attacks*. More specifically, it relies on measuring the theoretical complexity bounds necessary to safeguard releases from probabilistic polynomial-time (p.p.t.) adversaries, i.e., asserting that a p.p.t. adversary with exponential computing power cannot successfully recover the complete solution space and determine the right genotype sequences within the human-genome data set (i.e., individuals' real genome sequence) that participated in a given release. Zhou et al. [Zho+11] evaluated the theoretical complexity of recovery attacks on GWAS results. In summary, in that work, the authors offered a scheme to measure and decide when GWAS statistics can be safely released by enforcing that the solution space an adversary has to infer is larger enough when compared to the "given" side-information (i.e., the results of a GWAS released

in open-access). Their method quantifies the solution space in terms of N (the number of subjects participating in a GWAS study) and L SNP positions having GWAS statistics released. Depending on the type of statistics being released (i.e., aggregate or test statistics), the frequency space (seen by the adversary) gets smaller or larger when compared to the solution space. Therefore, if releases are built over too few records, the sequences can be inferred and leaked by the p.p.t. adversary. Thus, their methods decide when releases are permitted when the GWAS settings (number of participants and SNPs) satisfy the conditions for a safe release.

Similarly, to protect GWAS releases against *membership inference attacks*, several works proposed statistical inference methods to avoid noise-based solutions. These solutions are based on statistical inference methods that aim at measuring the risks of identifying the presence of individuals in a population from the observed GWAS statistics. These mechanisms are based on statistical tests, such as the Likelihood-ratio tests (LR-test), to measure how likely a particular individual belongs to a study. For instance, Homer et al. [Hom+08] proposed an attack that measures the identification power of a particular individual to be present in the case population. Their technique can be reversed-engineered and used to protect releases by blocking the participation of genomes that might be subject to privacy attacks. Additionally, Wang et al. [Wan+09] proposed a new hypothesis test (T_r) used to ensure that the identification power of all genomes is kept sufficiently below a given identification power threshold. Similarly, Zhou et al. [Zho+11] proposed the Λ metric used for the same goal. A more broader approach has been offered by Sankararaman et al. [San+09a], namely SecureGenome (SG). SG consists of several genome-oriented statistical verifications over the cohort of genomes being used in a GWAS and a reference population. SG consists in identifying SNP positions that would allow membership inference attacks. As a consequence, prohibiting the release of statistics over those SNPs.

This thesis builds on statistical-based protection methods for privacy-preserving releases of GWAS. In particular, the solutions presented extend Zhou et al.'s conditions [Zho+11] to allow protection against recovery attacks and on SecureGenome [San+09b] for allowing membership inference protection of *practical GWAS*. Due to their importance, both approaches are detailed later in Section 4.1.

2.10 Enabling practical federated GWAS

The previous sections have introduced the main features that this thesis envisions for allowing *practical GWAS*. They also discussed existing mechanisms that can support some of the functionalities, e.g., privacy-preserving *processing* and *releasing* of federated GWAS. Unfortunately, these properties have not been offered in

a uniformized manner by existing works. In addition, other issues are still to be solved, e.g., enforcing collusion-tolerance on the genomic data privacy level.

Therefore, this thesis identifies open challenges and enables several properties that are still uncovered by existing works, where most of them arise from the new assumptions, functionalities, and threat models the present thesis unfolds to enable *practical GWAS*.

An overview of the approaches and techniques this thesis leverages to address the unsolved problems and the creation of a *practical* federated GWAS scenario is presented below.

(i) Relies on TEE to ensure safe, secure, and privacy-preserving outsourcing and processing of genome data used throughout federated GWAS, while not impacting the accuracy and performance (in terms of computational resources) of GWAS releases. In addition, the proposed frameworks leverage TEE to perform privacy-protection mechanism verifications over the data to assure that only safe/private releases are publicly published.

(ii) Extends genome-oriented statistical methods to allow private releases of federated GWAS, while maximizing accuracy and utility of releases. In addition, it assumes a privacy-aware environment in terms of respecting current data-privacy regulations, such as HIPAA and GDPR. In other words, enabling donors the chance to withdraw consent over their genomic data at any time. Therefore, enabling safe updates of GWAS results since our solutions permit removal and addition of genomes over time (i.e., dynamic GWAS), while impeding attacks that leverage some aspects of dynamic releases to facilitate attack. For instance, impeding adversaries to build their attacks based on how statistics have evolved.

(iii) Replicates the same privacy guarantees for open-access dynamic releases of GWAS while assuming that up to all-but-one members in the federation might collude in order to collect additional knowledge to mount attacks against other members' genome data.

(iv) Considers and identifies new private release conditions when the existence of overlapping data (genomes and SNPs) among multiple GWASes. In other words, enforcing dynamic releases of interdependent GWASes.

(v) Enables the analysis of privacy-preserving *releasing* mechanisms in a distributed fashion while removing the need for genome data outsourcing and the presence of such data in a centralized location.

These choices were based on selecting the most fittable system architecture that can accommodate the privacy-preserving mechanisms employed by proposed solutions.

Chapter 3

Related Work

This chapter first presents the overall state of the art on federated GWAS and discusses the existing limitations and unsolved challenges this thesis aims to solve in this work. It presents the current approaches for privacy-preserving releases of GWAS, where the drawbacks and trade-offs are analyzed and compared to the solutions offered in the present thesis. Finally, it concludes by discussing the privacy risks when dealing with interdependent releases of GWASes.

3.1 Federated GWAS

Allowing collaboration among several genomic data holders certainly increases the accuracy and the confidence of the statistical findings. Given its benefits, the idea of adopting federated GWAS has become a real trend nowadays, with a variety of solutions being proposed to achieve this goal [Con+15b; Zha+18; Sad+18; Che+16b; Che+17a; Rai+18]. There are basic goals when conducting federated GWAS: (i) accuracy of the results, (ii) security (integrity and confidentiality) of the data while being outsourced and processed, (iii) privacy of both data holders and data donors, and (iv) overall performance/efficiency of the system, e.g., in terms of scalability [Fro+21].

It is not trivial to design an approach able to encompass all these features. Indeed, there is trade-off depending on the chosen cryptographic scheme, design and functionalities supported by the federation that needs to be taken into account. Recall Table 2.4 and Table 2.5) to see a comparison of existing privacy-preserving schemes features.

In addition, this thesis assumes new adversarial and threat models that have not been tackled in the literature yet. Therefore, some design goals, such as dynamic GWAS releases and GWAS private release conditions under the presence of interdependent studies could not be directly compared to existing works. Never-

theless, the proposed solutions are also compared to other existing approaches. For instance, they are compared with a adapted DP-based dynamic release mechanism (in Section 5.5).

The following sections provide a detailed discussion and compare the contributions of this thesis with related works.

3.2 Solutions for privacy-preserving *processing* of GWAS

As introduce in Section 2.7, there are a certain number of privacy-preserving approaches that can be used to conduct federated GWAS. This section presents existing federated GWAS works, while comparing them to the solutions presented in this thesis.

SMC-based approaches. Cho et al. [CWB18] offer a SMC mechanism where both individuals and computing parties (CP) privately share their data using the Beaver multiplication triples secret sharing mechanism. The system consists of three CPs that jointly combine their shares in order to computer GWAS statistics, e.g., p-tests, by employing cryptographic pseudo-random generators (PRGs) and random projection techniques to accelerate the GWAS computation. As a result, they claim that their framework scales better and has better efficiency than existing SMC solutions due to its linear complexity in terms of the number of individuals and SNPs considered in the study. While Kamm et al. [Kam+13] introduce a SMC framework where institutes share their genome dataset to third data storage for computing χ^2 tests, Zhang et al. [ZBA15] use secret sharing, which assumes (n, t)-threshold for corrupted parties. However, both only support a limited number of data and computations. Bogdanov et al. [Bog+14] first proposed a secret-sharing based SMC generic framework for conducting privacy-preserving federated analysis. Later, they offer a similar SMC scheme [Bog+18] to perform Principal Component Analysis (PCA) over distributed genomic data. PCA is a method used to detect and avoid group stratification-like errors while performing GWAS. Constable et al. [Con+15b] propose a *Secure Two-Party Computation* (STPC) approach to perform privacy-preserving χ^2 and MAF processing. Their approach uses the Portable Circuit Format (PCF), which is a garbled circuit-based SMC framework. Yet another SMC-based approach based on garbled circuit is offered by Jagadeesh et al. [Jag+17]. However, it has some limitations, such as only allowing boolean operations and cannot be deployed in larger-scale GWAS settings. Similarly, Tkachenko et al. [Tka+18] offers another STPC approach that relies on the ABY framework to compute χ^2 in a privacy-preserving manner. Schneider

et al. [ST19] offers another SMC approach that leverages the ABY SMC framework for privacy-preserving *processing* of Similar Sequence Queries (SSQs) over aggregate genome datasets outsourced by several data holders. Currently, the main focus of the community is to improve and offer the performance of SMC approaches for conducting privacy-preserving distributed GWAS.

HE-based approaches. Mott et al. [Mot+20] study several HE-based encryption schemes for human genotype and phenotype data to allow private sharing while maintaining their statistical and structural properties. They compare the advantages and limitations of encryption using orthogonal and linear transformations. Lu et al. [LYS15] describe a method where a researcher creates a couple of keys and communicates the public key to biocenters that sequence genomes. The biocenters then encrypt their genomes using a HE-key scheme and store the resulting encrypted genomes on a public cloud. The cloud then uses homomorphic computations on the encrypted genomes to obtain encrypted GWAS statistics that only the researcher can decrypt. Their solution assumes that no one can gain access to both the encrypted genomes and the private key (e.g., if the cloud and the researcher collude). In such a situation, all genomes that the biocenters shared would be leaked. Hasan et al. [Has+18] propose another hybrid scheme that combines Yao's garbled circuits and tree-based Paillier homomorphic encryption to enable privacy-preserving aggregation and output of genomic count queries. Similarly, Kim et al. [KL15] propose a fully HE scheme to run χ^2 tests using 80-bit key security. More recently, a distributed GWAS system, which uses somewhat HE and SMC methods, and answers only *yes/no* responses for putative markers SNPs (rather than releasing χ^2 values), was introduced by Bonte et al. [Bon+18]. Nonetheless, it is known that protecting private data only by denying access to it, is not enough. Indeed, one can leak genomic private data by exploiting *yes/no* answers from such a system, as shown in [SB15a; AAC21; Ayo+20]. For instance, an attack introduced by Shringarpure et al. [SB15a] demonstrated that leveraging *yes* or *no* responses from the Beacon's Network service [1] is sufficient to infer the membership/participation of known genomes in the dataset. Other similar works, such as [VAC19; Al +17; Rai+17a] have shown the practicality of privacy attacks over the Beacon platform. Lauter et al. [LLN14] propose a level homomorphic scheme where all genomic data of individuals are homomorphically encrypted and GWAS statistics are computed over them at once. Nevertheless, as stated by the authors, HE still needs improvements to allow efficient operations over encrypted data under multiple keys (one key per individual, for example). Zhang et al. [Zha+15]

[1] The Beacon Network is a global genome database engine that answers only (yes/no) GWAS-based queries and hence believed to not disclose private genomic information of donors. Available at: https://beacon-network.org

present two methods for computing χ^2 statistics of GWAS leveraging HE, namely the error-less division protocol and secure approximation division that can be chosen according to the design goals of the system in terms of accuracy and complexity. Ugwuoke et al. [UEL17] created a hybrid framework that combines Paillier HE for computing addition operations and SMC methods to perform multiplication over the genomic data used to compute GWAS statistics, such as LD. It is assumed that entities cannot collude while the data is aggregated. Wang et al. [Wan+16] offer HEALER, an improved HE-based protocol to compute exact logistic regression models for GWAS. They used a compression scheme to reduce HE-encrypted data and a parallel computing mechanism to operate encrypted data. However, they have assumed small datasets in their experimental evaluation. More recently, Blatt et al. [Bla+20] show a HE framework under larger GWAS settings (25,000 genomes). It is assumed the presence of a centralized entity collects homomorphically encrypted genomes from a number of individuals that specially encodes the data to allow parallel execution of HE operations inside a HE-enabled cloud machine. However, their approach is vulnerable to a collusion attack between the GWAS coordinator and the cloud machine.

DP-based approaches. A framework that combines data anonymization techniques and DP was offered by Wang et al. [WMC14]. In their solution, blocks of data are privately shared by data holders relying on DP to perturb data. Local DP is a variation of DP where data owners add noise to their local data before sharing it for aggregation [Cor+18]. Inspired by that, Lu et al. [LS17] described a Distributed Differential Privacy (DDP), in which parties perturb their local data shares before sharing them so that both data aggregator and possibly colluding parties cannot launch successful inference attacks over the aggregate released data. However, their system evaluation considered movie ratings and electricity consumption data, which do not need high precision data results as needed by genomics studies. Equivalently, Liu et al. [Liu+21] offered another local DP scheme where random perturbation is applied at the genome level, i.e., each genome sequence receives a DP noise, and therefore each genome has a local privacy budget. Once all perturbed genome data is aggregated in a federated fashion, responses to genomics queries are protected by a global privacy budget. Similarly, another DP-based approach has been proposed by Simmons et al. [SBS19]. It combines the addition of minimal amounts of noise perturbation using Bayesian and Markov Chain Monte Carlo techniques. The authors claim that their approach is able to release more data with minimal privacy protection loss. Local DP-based approaches enforce secure and privacy-preserving sharing of genomic data. Nevertheless, it comes with a reduced data utility due to the introduction of noise in the data [Fre+14], which decreases the accuracy of final GWAS outputs. MedCo [Rai+18] is a distributed

protocol built on Unlynx [Fro+17], that allows exploratory medical analysis. It combines HE, DP, and other improvements to compute statistics over medical data in a private manner. The system consists of several data providers that secretly share their records to aggregate data and answer authorized queries. The authors claim that their scheme can also be used for GWAS. Recently, Aziz et al. [Azi+21] present new DP-based algorithms that dynamically manage privacy budgets of the DP mechanism to find optimal values for ϵ. Thus, increasing the accuracy of GWAS releases. In their work it is assumed both centralized and distributed models for conducting GWAS. Similarly, Zhang et al. [Zha+22] propose a fragmentation method, which keeps ϵ-indistinguishability based on local DP, to split genome data into different partitions so that the knowledge an adversary can acquire by compromising several nodes conducting aggregation tasks is not enough to mount successful genomic de-identification attacks.

In comparison to the above solutions, the techniques used in this thesis do not rely on adding noise to data, therefore keeping the accuracy of released results. Additionally, it is worth recalling that the encryption-based approaches, such as HE and SMC, come with high costs and complex designs. Consequently, they face scalability issues.

TEE-based approaches. In 2012, Canim et al. [CKM12] offered the first approach that leverages secure cryptographic hardware to allow secure sharing and storage of genomic data inside a third-party machine. More specifically, they relied on the IBM 4764 cryptographic co-processor [IBM21] installed on the untrusted server. Therefore, allowing a tamper-resistant process of genomic data, and using symmetric encryption to receive and answer GWAS queries. Next, PREMIX [Che+16a] was one of the first approaches that relied on Intel SGX enclaves to evaluate individual genomic admixture by allowing the collaboration of multiple entities sharing genomic data. The encrypted data from each site is sent to a centralized enclave that answers GWAS queries from authorized clients. PRINCESS [Che+16b] also performs GWAS tests using SGX enclaves for rare-disease collaboration studies, where genomic data is securely shared and computed inside a centralized enclave. Before being transmitted, all genomic data go through some pre-processing steps, such as data segmentation and compression to improve the efficiency of the system. In addition, PRESAGE [Che+17a] applies encoding and indexing methods on genomic data to answer private queries with high performance. After encryption, the data is outsourced to an untrusted party, e.g., an SGX-enabled cloud provider, which answers genomic queries with encrypted results. SAFETY, proposed by Sadat et al. [Sad+18], combines HE for the aggregating data holder's genomic data inputs and SGX enclaves for more complex statistical processing. Similarly, Chenghong et al. [Che+17b] offered SCOTCH,

which uses a hybrid platform that leverages HE to gather genomic data from several data holders and compute aggregate statistics in a faster manner, and uses SGX to compute the data securely. Carpov and Tortech [CT18] won the Track 2 of the iDash Privacy and Security Workshop 2017 competition that challenged the research community to offer the most efficient approach to compute *chi*-square statistics using SGX. They could succeed in this task by implementing a horizontal partitioning technique to encode the genomic data in a more efficient way and using parallel processing to speed up computations. SkSES is a framework offered by Kockan et al. [Koc+19] that applies filtering and compression mechanisms to VCF files being shared to a centralized enclave, which securely computes *chi*-square statistics. They also proposed the use of sketching data structures to increase performance and computational running time. More recently, Bomai et al. [BAZ20] presented another hybrid approach that combines multi-key HE and SGX to enable secure sharing of genomic data from multiple data holders to a SGX-enabled cloud provider that computes *chi*-square GWAS statistics to answer queries from authorized users.

It is essential to notice that all solutions mentioned above are not concerned about privacy-preserving *processing* of GWAS, i.e., they only focus on the sharing and processing part. Therefore, the released results might still be subject to genomic privacy attacks. In contrast, this work supports privacy in both aspects of fully privacy-preserving federated GWAS (processing and releasing).

3.3 Solutions for privacy-preserving *releasing* of GWAS

Section 2.8.1 and Section 2.8.2 introduced recovery and membership attacks, respectively. This section presents the related work on the protection of GWAS releases against these attacks. Existing approaches are categorized into three types, depending on the method they build on:

- Measuring the theoretical complexity of recovery attacks in order to define safe thresholds based on genomic-oriented statistical analysis to certify that a study has used enough genomes so that probabilistic polynomial-time (p.p.t.) adversaries cannot correctly infer genotype information of the participants in the study [Zho+11].

- Using (reverse-engineering) statistical inference test methods [Hom+08; San+09a; Wan+09; Cai+15], such as LR-tests, to evaluate and measure the probability of identifying vulnerable individuals in the study, and posteriorly removing potential targets out of the study [Hom+08] or prohibit the releases of statistics over SNPs that would enable membership inference [San+09a; San+09b].

- Utilizing DP mechanisms to perturb the results (applying noise) of a study to enforce differentially-private releases [SB16; SBS19; AAU20a], i.e., ensuring that no individual can be identified as a participant of a study.

In the following, it is presented existing solutions that rely on DP mechanisms to safely release GWAS statistics. Jiang et al. [Jia+14] propose to use a new privacy-budget approach that balances data perturbation with privacy risks using statistics of a LR-test. On the other hand, DP was used to enable differentially private logistic regression by perturbing the objective function [Yu+14] instead of the final output of a GWAS. Uhlerop et al. [USF13] offered a scheme to release GWAS statistics over M best ranked (most significant) SNPs using DP with Laplace mechanism. Tramèr et al. [Tra+15] evaluated several potential DP manipulations for genomic membership privacy in order better balance the trade-offs between data utility and privacy. Simmons et al. [SB16] enables differentially private GWAS by leveraging the neighbor distance algorithm proposed by Johson et al. [JS13] in order to apply noise in a more efficient manner. Their approach is able to protect individuals in both case and control populations. In another work, Simmons et al. [SSB16] propose two DP frameworks for privacy-preserving GWAS: PrivSTRAT applies data perturbation considering the group stratification in a study, and PrivLMM is based on Linear Mixed Models (LLMs).

Although efforts and works have been proposed to enforce DP in a dynamic environment (i.e., assuming continuous releases), such as DP under continual observation [Dwo+10; CSS11] and for growing databases [Cum+18]. To the best of my knowledge, no work has applied such DP techniques under a dynamic GWAS scenario.

Furthermore, it is important to recall that such DP-based approaches directly impact the accuracy of GWAS releases, and therefore an expected data utility loss comes inherited with these approaches. In addition, two recent works by Almadhoun et al. [AAU20a; AAU20b] have shown that the existence of dependent records (e.g., relatives' genomes) in a genomic database can diminish the privacy guarantees of DP mechanisms. In contrast, this thesis combines statistical tests with exhaustive verification methods to enforce the genomic privacy of individuals over continuous releases, without perturbing data.

Finally, the most similar work to this thesis is presented by Ayoz et al. [AAC21]. In their work, they show that both recovery and membership attacks can be launched by sequentially querying the genomic data-sharing Beacon's platform. They assume a similar threat model as this thesis, where new participants are added over time and statistics results (queries, in their case) are updated. The challenge behind their threat model is the same as assumed in this thesis. In particular, an attacker can learn genomic information of new participants as they are added by observing how the answers of the Beacon evolved within time, i.e.,

between time t and $t+1$. Nevertheless, the authors only discuss ideas and some potential countermeasures to mitigate the risks of recovery and membership attacks under this setting, not offering a concrete solution.

Nonetheless, all these approaches cannot cope with the dynamic settings of *practical GWAS*, i.e., new genomes being added and removed on the fly. Therefore, they can no longer keep genomic privacy when performing continuous releases of GWAS results. In addition, these techniques cannot cope with collusion among entities sharing the genomic data. In particular, some data holders might collude and exchange information about the shared data in order to circumvent the conditions for safe releases of the protection mechanisms. The issues that are raised by these new assumptions and the threat model (collusion among participating data holders) are one of the main contributions of this thesis, which are presented and addressed in Chapter 4 and Chapter 5.

3.4 Issues of interdependent GWAS releases

The decreasing genome sequencing costs have been motivating a scenario towards sharing the results of independent GWASes on different phenotypes to construct multi-omics datasets [Im+12]. As the availability of genomic data and multiple GWAS releases are becoming more accessible, there is now an increased risk for new genomic privacy attacks, as adversaries can now cross-reference several studies in order to gain additional knowledge and circumvent existing safe release conditions [Gür+18].

Indeed, in real-life settings, GWASes might consider overlapping sets of individuals, each having a focus on a specific set of genomic variations, some of which might also be used in other studies for economic reasons. It is rather likely that federations will run different GWASes simultaneously (e.g., one on diabetes and a second studying lung cancer [Den+20]). As a consequence, an adversary is able to base its attack on the results of a single multi-trait study or even from GWASes released from multiple federations.

Although some works have started looking at the problem of dependency among genomic data subjects (i.e., the existence of dependent records within the same study or genomic database, such as an individuals' relatives [AAU20a; AAU20b; Hum+17; AH17; HTH19; Hum+22; Der+22]), and studied how this scenario might compromise genomic privacy, this thesis is the first to evaluate privacy the risks when releasing statistics of multiple interdependent GWASes.

In particular, Chapter 5 shows that protecting single-GWAS releases is not enough. It shows that by the observation of several "safe" single-GWAS releases, an adversary can still leak genomic data information from individuals by carrying out new variations of recovery and/or membership attacks leveraging overlapping

data (e.g., genomes and SNP positions) used in different studies. As a response, this thesis evaluates the risks and defines the new conditions to allow safe releases of multiple interdependent GWASes.

3.5 Overview and current stage of federated GWAS

Table 3.1 presents an overview of the solutions found in the literature that enables federated GWAS. These works have been discussed in previous sections. This table considers works that conducted actual GWAS, not Principal Component Analysis (PCA), such as Bogdanov et al. [Bog+18] and Ostrak et al. [Ost+21]; and Similar Sequence Query (SSQ) such as Schneider et al. works [ST18; ST19] under the federated setting. In addition, despite their valuable contribution, some works such as PRINCESS [Che+16b], Froelicher's et al. [Fro+21] assumed the outsourcing and sharing of PLINK format data (rather than VCF files) to conduct GWAS. PLINK [Pur+07] is open-source C/C++ application comprised of a variety of tools to facilitate and conduct GWAS.

Even though assuming different cryptographic schemes, not all works are able to reconcile privacy-preserving ***processing*** with privacy-preserving ***releasing*** of GWAS. Indeed, only two other works [Rai+18; Azi+21] and the works offered in this thesis are able to securely process and privately release GWAS. In addition, except for our solutions (**DyPS** and **I-GWAS**), the existing works cannot conduct dynamic and public releases of GWAS where results are updated when new genome requests are generated. Besides, **I-GWAS** framework is the only solution that can also allow privacy-preserving releases of interdependent GWASes. Lastly, **GenDPR** allows distributed assessment of private GWAS releases by designing a multi TEE-enclave environment where federation members jointly verify which data can be safely used for the creation of safe releases.

Table 3.1: Overview of existing federated GWAS solutions. **STPC:** Secure Two-Party computation; **CPs:** Computing parties; **SS:** Secret-Sharing mechanism; **GC:** Garbled circuit; **MCMC:** Markov Chain Monte Carlo; ***:** Releases *yes/no* answers, which can be vulnerable to similar Beacon's privacy attacks [AAC21; Rai+17a; SB15a; Al +17; VAC19].

Work	**Cryptographic scheme**	**System model**	**Secure outsourcing/aggregation/-computation**	**Privacy-preserving *processing***	**Privacy-preserving *releasing***	**Open-access/Public releases**	**Dynamic releases**	**Privacy-preserving interdependent releases**
Kamm et al. [Kam+13]	SMC	Distributed CPs	✓	✓	X	X	X	X
Zhang et al. [ZBA15]	SS + SMC	Fully distributed	✓	✓	X	X	X	X
Cho et al. [CWB18]	SMC	Distributed CPs	✓	✓	X	X	X	X
Bogdanov et al. [Bog+14]	SMC (Sharemind framework [BLW08])	Distributed CPs	✓	✓	X	X	X	X
Constable et al. [Con+15b]	GC-based SMC	STPC	✓	✓	X	X	X	X
Jagadeesh et al. [Jag+17]	GC-based SMC	STPC	✓	✓	X	X	X	X
Tkachenko et al. [Tka+18]	SMC (ABY framework [DSZ15])	STPC	✓	✓	X	X	X	X
Hasan et al. [Has+18]	HE + GC-based SMC	Centralized	✓	✓	X	X	X	X
Lauter et al. [LLN14]	HE	Centralized	✓	✓	X	X	X	X
Lu et al. [LYS15]	HE	Centralized	✓	✓	X	X	X	X
Kim et al. [KL15]	Fully HE	Centralized	✓	✓	X	X	X	X

Bonte et al. [Bon+18]	HE and SMC	Centralized (HE) and Distributed (3 CPs)	✓	✓	✓ *	X	X	X
Zhang et al. (FORESEE) [Zha+15]	HE	Centralized	✓	✓	X	X	X	X
Wang et al. (HEALER) [Wan+16]	HE	Centralized	✓	✓	X	X	X	X
Blatt et al. [Bla+20]	HE	Centralized	✓	✓	X	X	X	X
Raisaro et al. [Rai+18]	HE + DP + Unlynx [Fro+17]	Fully distributed	✓	✓	✓	X	X	X
Aziz et al. [Azi+21]	Improved ϵ-DP	Centralized	✓	✓	✓	X	X	X
Froelicher et al. [Fro+21]	Multiparty HE	Fully distributed	✓	✓	X	X	X	X
Canim et al. [CKM12]	TEE (IBM 7464 cryptographic co-processor [IBM21])	Centralized	✓	✓	X	X	X	X
Chen et al. (PREMIX) [Che+16a]	TEE (Intel SGX)	Centralized	✓	✓	X	X	X	X
Chen et al. (PRINCESS) [Che+16b]	TEE (Intel SGX)	Centralized	✓	✓	X	X	X	X
Chen et al. (PRESAGE) [Che+17a]	TEE (Intel SGX)	Centralized	✓	✓	X	X	X	X

Sadat et al. (SAFETY) [Sad+18]	TEE (Intel SGX)	Centralized	✓	✓	X	X	X	X
Chenghong et al. (SCOTCH) [Che+17b]	TEE (Intel SGX)	Centralized	✓	✓	X	X	X	X
Carpov and Tortech [CT18]	TEE (Intel SGX)	Centralized	✓	✓	X	X	X	X
Kockan et al. [Koc+19]	TEE (Intel SGX)	Centralized	✓	✓	X	X	X	X
Bomai et al. [BAZ20]	TEE (Intel SGX)	Centralized	✓	✓	X	X	X	X
(This thesis) Pascoal et al.(DyPS) [Pas+21]	TEE (Intel SGX)	Centralized	✓	✓	✓	✓	✓	X
(This thesis) Pascoal et al. (I-GWAS) [Pas+23])	TEE (Intel SGX)	Centralized	✓	✓	✓	✓	✓	✓
(This thesis) Pascoal et al. (GenDPR) [PDV22])	TEE (Intel SGX)	Distributed (multi-enclave setting)	✓	✓	✓	✓	✓	✓

Chapter 4

Dynamic, Privacy-Preserving and Secure Federated GWAS (DyPS)

The previous chapters presented remaining challenges to allow fully privacy-preserving federated GWAS in addition to introducing foreseen features to enable *practical GWAS*. Namely, (i) fully privacy-preserving GWAS design in terms of sharing, processing, and open-access releasing; (ii) privacy-aware in the sense of complying with data-privacy regulations (such as GDPR) and so allowing participants to withdraw consent at any time while producing safe releases of dynamic GWAS, where results are updated over time once new genomes are added or removed; (iii) collusion-tolerant GWAS, i.e., the federation is able to face all-but-one participants colluding to attack others' data and still be able to compute and protect GWAS releases without privacy breaches; (iv) enforcing all above constraints but assuming the existence of multiple overlapping studies (e.g, reusing same genomes over several studies), which demands new release conditions to not compromise the safety of previous or next releases); and (v) enabling privacy-protecting mechanisms to be performed in a distributed fashion.

In particular, this chapter addresses challenges (i) to (iii), whereas the next Chapter 5 details the extensions needed to obtain property (iv). Next, Chapter 6 offers a framework to distributively assess private GWAS releases without genome data outsourcing before Chapter 7 introduces a holistic scheme that accommodate all functionalities simultaneously.

This chapter presents **DyPS**, a novel and scalable framework that reconciles secure and privacy-preserving *processing* and *releasing* of federated GWAS, while allowing updates of results and collusion-tolerance. Particularly, **DyPS** leverages Intel SGX to enable secure sharing and processing of genomic data while computing GWAS statistics and evaluating releases' safety conditions over genome data being outsourced by several biocenters of a GWAS federation. Additionally, **DyPS** improves the current state-of-the-art mechanisms for safe releases of GWAS,

which allow only safe releases of static GWAS. In fact, DyPS enables safe releases of dynamic GWAS (i.e., allowing GWAS results to be updated over time when new genome operation requests come, for example when addition or removal of genomes are requested by data holders). To the best of author's knowledge, DyPS is the first solution able to reconcile in a homogenized form the issues of privacy-preserving computation and releasing of not only static but also dynamic GWAS. By assuming this dynamic GWAS model, DyPS is able to accept participation consent withdrawal from individuals participating in a study (in order to comply with data-privacy regulations, for example). DyPS implements efficient algorithms that determine how to safely release and update GWAS statistics without noise addition, which guarantees no data utility loss. Moreover, DyPS tolerates up to all-but-one colluding biocenters without privacy leaks.

4.1 Conditions for safe GWAS releases

Genomic privacy attacks leveraging GWAS results. Let us first recall how privacy attacks on GWAS releases work. An adversary may try to leverage a GWAS's metadata (i.e., lists of SNPs and pseudonymized genomes) and test and/or aggregate statistics to breach the genomes' owners privacy. Figure 4.1 illustrates the typical information that an adversary can observe: (i) the list of the L SNPs; (ii) the N genome pseudonyms used in the GWAS, and (ii) the GWAS results which may include one or several of the statistics introduced in Section 2.2. To be noted, that if the adversary is a biocenter contributing to the GWAS computation, this adversary knows a subset of the SNPs and genome pseudonyms as well as a subset of the content of the table. This adversary knowledge increases in case of collusion between several biocenters.

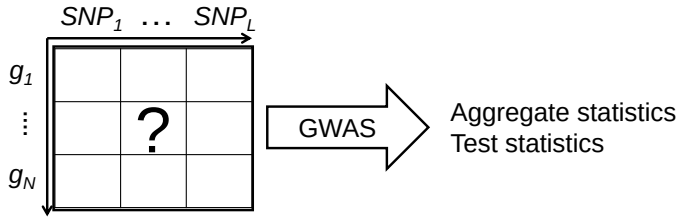


Figure 4.1: Observable data for privacy attacks on GWAS.

Section 3.3 introduced the main existing approaches for protecting GWAS releases. DyPS builds on Zhou et al. [Zho+11] and SecureGenome [San+09a] solutions. The former is used to quantify the needed number of genomes to protect recovery attacks, while the latter is used to guard releases against membership attacks. It is worth recalling that those approaches consider static GWAS, which

in turn cannot be directly applied when conducting dynamic releases as DyPS does. In particular, DyPS enforces that these conditions are met while GWAS results are updated, which is more challenging. Furthermore, DyPS ensures these conditions are enforced when facing collusion among participants.

The existing conditions for GWAS safe releases differ according to the statistics produced by the GWAS and the targeted attack. Table 4.1 summarizes these conditions according to the type of release and attack.

Table 4.1: Release conditions for GWAS aggregate or test statistics computed over L SNPs and N individuals.

Observable statistics	**Attack**	**Attack unfeasibility conditions**
Aggregate	Membership	*Single allele freq.:* LR metric is sufficiently low, $N > 100$ *and* MAF > 0.05 (1) *Pairwise allele freq.:* Λ or T_r metric is sufficiently low. (2)
	Recovery	$2(N-1)/\texttt{log}(N+1) > L$ (3)
Test	Membership	$2N/(\texttt{log}(N+1)-1) > L$ (4)
	Recovery	$2(N-1)/(\texttt{log}(N+1)-1) > L$ (5)

4.1.1 Protecting recovery attacks

Zhou et al. [Zho+11] showed that recovery attacks on allele frequencies and on test statistics are NP-complete. They argue that the release of a GWAS is safe if the solution space an adversary has to explore is significantly larger than the GWAS result space. Let us take a closer look at this safety condition with a single GWAS study comprised of L SNPs and N genomes. The size of the ***solution space*** $|S|$ is the number of possible matrices that verify a given statistical result. $|S|$ is at least $\binom{2^L}{N}$, that is, the complexity of selecting N SNP sequences from 2^L sequences into which the L SNPs expand. The size of the allele ***frequency space*** $|D|$ is equal to $(N+1)^{L+\binom{L}{2}}$, which corresponds to all possible values for L single SNPs and $\binom{L}{2}$ SNP pairs over N sequences. The condition for a safe GWAS release of this single study is that the size of the solution space S is large compared to the size of the frequency space D:

$$|S| > |D| := \frac{(2N-1)}{log(N+1)} > L \tag{4.1}$$

Similarly, when considering test statistics releases (e.g., p-values and r-squares (r^2)), the ratio between the solution space and the test static space, i.e., $|R^2|$, is also to be kept within safe boundaries. One crucial information when launching the attack over GWAS test statistics data is to correctly identify the values of r (or their signs) given r^2 results. Given that the space size for r values (assuming the adversary was able to recover it) is approximately $(N+1)^{L+\binom{L}{2}}$, and by assigning r values to r^2, $|R^2| = \frac{(N+1)^{\binom{L}{2}+L}}{2^{\binom{L}{2}}}$. Hence, one can obtain the following relation:

$$|S| > |R^2| := \frac{2N}{log(N+1) - 1} > L \tag{4.2}$$

Recalling Table 4.1, it is noticed that Equation (3) covers (i.e., demands more genomes for protection) Equations (4) and (5) for the protection of aggregate statistics releases against recovery attacks and test statistics against membership and recovery attacks. Hence, the solutions of this thesis use Equation (3) as the upper-bound to select safe batch of genome requests. Thus, assuming a more conservative approach. In addition, it is important to notice that the offered solutions combine the use of the equations with LR-tests so that releases are protected against both attacks simulteneously.

4.1.2 Protecting membership attacks

As discussed in Section 2.9, several works that proposed statistical-based solutions to mitigate membership attacks from the observation of GWAS releases. Despite presenting an genomic privacy attack, Homer et al. method [Hom+08] can be used to identify genomes vulnerable to membership inference, and therefore impeding the participating of individuals at risk. A more recent approach, SecureGenome [San+09a; San+09b] performs several genome-wide statistical verification to select safe data that can be used to create safe GWAS releases. Besides, It proposes to calculate the sensitivity and specificity of a LR-test to decide which allele frequencies for a given dataset can be safely released. The hypothesis test T_r metric [Wan+09] or the Λ metric [Zho+11] can also be used to ensure that the identification power of released genomes is sufficiently low.

The solutions of the present thesis build on SecureGenome [San+09a; San+09b] due to its genome-oriented approach and because it has been adopted and commonly used to quantify membership risks for protecting static GWAS releases by many works [AH17; Zho+11; Pas+21; Hal+21]. Besides that, our solutions show the feasibility of extending SG to cope with the new issues brought by the existence of dynamic GWAS releases. Since SecureGenome is an important part of the present thesis, and provides the ground properties of our solutions, SecureGenome (SG) is detailed in the following sections.

4.2 Detailing SecureGenome

SG's rationale. The goal of SG is to select a safe subset of SNPs from the original SNP-set of a GWAS from which the observation of released statistics would not allow membership inference of any participant (genome) relying on several genome-oriented statistical verifications along with a likelihood-ratio test (LR-test). To achieve that goal, SG applies a combination of privacy assessments by computing some statistics over the pool of genomes participating in a study and a reference set. Therefore, SG assumes the availability of a reference genome dataset and of the pool of individuals participating in the study. These genome datasets are used to draw the hypothesis of the test.

Assumptions of the SG's model. SG assumes that the SNPs considered in the LR-test are independent due to the fact that selected SNPs can be far apart each other. In addition, the standard SG analysis does not assume genotyping errors, i.e., the allele information over the SNPs are precise, and this is a common assumption in the literature. Although, the authors show by experimentation that genotyping errors only decrease the identification power of the attack.

Furthermore, SG's LR-test assumes that SNP allele frequencies in the population are bounded away from zero and one. Hence, there is $a > 0$ such that $a \leq p_l \leq 1 - a$, where p_l corresponds to the allele frequency of SNP l in the cohort. This is an expected assumption due to the fact that GWAS only considers SNPs whose minor allele frequencies (MAF) are well represented in the selected population. In a nutshell, SecureGenome consists of the following steps:

1. Removing SNPs with rare allele frequencies (MAF $< MAF_{cutoff}$): In this step, SG pools and compute the allele frequencies of each position (considering both case and control genome sets) and checks if the MAF of SNPs are below or equal to MAF_{cutoff}, usually below 0.05. SNP positions with low MAF forms characteristic outliers that can be used by adversaries to deduce membership. They are therefore not considering for the subsequent steps.

2. Removing SNPs in high LD (p-value on $r^2 < LD_{cutoff}$): High linkage disequilibrium (e.g., p-value on $r^2 < 10^{-5}$) indicates highly connected SNPs. Such information can be leveraged to attack individuals by using the association levels among SNPs, as shown in [San+09a; Zho+11]. For this step, SG employs a greedy algorithm that checks and removes SNPs that are in LD. If any two SNP positions are found to be in LD, the most ranked SNP is kept for further verification. SNP positions in high LD are then prohibited from participation. Notice that Steps 1) and 2) are particularly important to match the assumption of the presence of independent SNPs for the LR-test analysis in the next step.

3. Identifying SNPs that would allow membership inference (LR-test verification over remaining SNPs) at specified detection power threshold and false-positive rates. Therefore, SG uses the remaining SNPs from the previous steps to conduct the LR-test described below.

   SG's null hypothesis draws the probability of an individual belonging to the case genome set consisted of N genomes under Hardy-Weinberg Equilibrium (HWE)[1]. In contrast, SG's alternative hypothesis draws the probability of the individual belonging to the pool consisted of $N-1$ genomes under the null hypothesis and HWE.

$$LR = \sum_{l=1}^{L} [x_{n,l} \log \frac{\hat{p}_l}{p_l} + (1 - x_{n,l}) \log \frac{1 - \hat{p}_l}{1 - p_l}], \tag{4.3}$$

   where L is the number of pre-selected independent SNPs in a study (recall steps (i) and (ii) discussed above), $x_{n,l}$ is the allele information at SNP position l of individual $n \in [0, N-1]$, p_l is the allele frequency of SNP position l in the population, and $\hat{p}_l$ is the frequency of SNP position l in reference set.

   The power $1-\beta$ of the LR-test can be found as a function of the pool size N, the number of SNPs L with a tolerable false-positive rate α. Conversely, using the Neyman-Pearson lemma that states that no test can have larger power than the LR-test, the power $1-\beta$ achievable for the LR-test given (L, N, α) determines the maximal L so that no (α, β)-test can be obtained for a pool of size N.

   In particular, SG uses the LR-test to empirically verify the identification power of the N individuals in a study by sampling their allele sequences over several subsets of SNPs $\in L$ in an iterative fashion while removing SNPs that would keep the identification power of any participant above a specified detection power threshold. Thus, impeding the release of GWAS statistics of SNPs that would pose membership inference risks of any individual.

   According to Sankararaman et al. [San+09b], the SG's LR-test approximates the Gaussian distribution that is parameterized by the relationship between sample size N, number of independent SNPs L, statistical power $1-\beta$, and type I error probability (or significance level) α when both L and N are moderately large, according to the central limit theorem. The extended version of SG [San+09b] further demonstrates that this approximation also holds when N is not assumed to be large using the Lindeberg-Feller central

[1] HWE states that genomic variations are stable among generations given the absence of disturbing/external factors.

limit theorem. Besides, the authors empirically show that this approximation also hold for small N and L values.

As a summary, at the end of the LR-test, SG has identified a SNP-set L' belonging to the original SNP-set L (i.e., $'L \in L$), from which the observation of GWAS statistics over these SNPs would not allow an adversary to identify the presence of individuals participating in the study respecting the configured (i) upper bound on the power (the probability that an individual is correctly identified to be in the population); and (ii) upper bound on the false positive rate (the probability that an individual that is not in the population is erroneously identified to be in the population) of the LR-test. Therefore, the SNPs belonging to L' can then be safely used in a GWAS as their statistics would not allow inferring the presence of underlying individuals.

Additionally, SG can be extended to cope with the existence of relatives in a cohort and how their presence might compromise the risk of others. For instance, the authors showed that the identification of first-order and second-order relatives of a target individual decreases according to the number of exposed SNPs [AH17; San+09a].

The output of SG is used to detect SNPs that can have their statistics released while protecting membership attacks on the genomes participating in a study. Nevertheless, SG assumes only single and static GWAS releases, which is not enough for allowing the properties of *practical GWAS* assumed in this thesis.

Indeed, as shown throughout this work, new conditions and so new approaches need to be enforced to accommodate safe releases guarantees under these settings. The extensions and steps enforced by **DyPS** to guarantee SG protection in a dynamic setting (i.e., over continuous releases) are presented in Section 4.5.4. Similarly, Section 5.4 presents the steps of **I-GWAS** to allow dynamic releases on the presence of overlapping studies.

This thesis adopts the privacy parameters suggested in SecureGenome [San+09b] after extensive evaluation of their approach. Namely, $MAF_{cutoff} = 0.05$, $LD_{cutoff} = 10^{-5}$, false-positive rate $= 0.1$ and 0.9 detection power rate threshold $= 0.9$.

4.3 DyPS' system and threat models

This section details **DyPS**' system and threat models.

System model. Figure 4.2 illustrates **DyPS**' system and threat models. **DyPS** considers a federated system of B biocenters $\{bioc_1, \cdots, bioc_B\}$, which obtain and locally store the allele sequences of individuals. Each biocenter may sequence the genomes of case and/or control individuals, potentially at different speed rates

due to various models of sequencing machines. DyPS assumes that biocenters can contribute genomes to a GWAS, and that individuals retract their participation consent, at most once. The biocenters are connected through an asynchronous communication network. Their common goal is to perform a GWAS over a set of L SNPs. DyPS denotes by $LtoN(L)$ the minimum number of genomes that need to be used to release statistics computed over L SNPs so that Equations (3), (4) and (5) in Table 4.1 are enforced. Reciprocally, DyPS denotes $NtoL(N)$ the maximum number L of SNPs that can be safely released according to the number of participating genomes N in a study.

DyPS assumes the availability of a server equipped with a TEE dedicated to the federation. Consequently, it might identify the pseudonyms of the genomes and SNP ids that are used for the computation of GWAS statistics. DyPS assumes accordingly that the parties have access to this information since the pseudonyms of the used genomes are made public. This TEE is responsible for executing the actual GWAS computation, and ensuring that the GWAS statistics can be safely released before openly published.

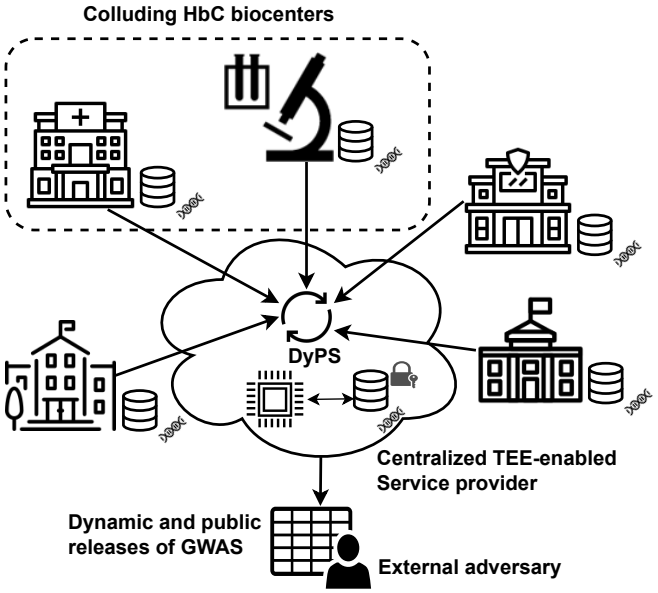


Figure 4.2: DyPS' system and threat models.

Threat model. DyPS assumes a probabilistic polynomial-time adversary that has access to a reference population with an allele distribution identical to that of the case population. Note that this reference population is not the same as the control population of the study. To launch membership attacks, the adversary also has access to a victim's DNA profile (i.e., its genetic code).

DyPS considers an honest-but-curious adversary controlling biocenters and

monitoring the GWAS releases: biocenters follow the protocol and do not forge genomes. In addition, DyPS considers collusion between biocenters that aim at inferring information from the non-colluding biocenters. DyPS assumes that up to $f = (B - 1)$ biocenters can collude to launch either membership or recovery attacks on the released GWAS results. The investigation of additional adversarial behaviors is left for future work of this thesis. Indeed, enforcing genomic data genuineness is an open challenge as digital genomes can be forged [Hua+15; Rai+17b] or synthesized [Ney+17], which would make data poisoning attacks undetectable.

As presented in Section 2.7.1, relying on TEE such Intel SGX has some limitations. For example, even though all data is stored encrypted on the TEE server and only manipulated by the enclave, DyPS does not cope with possible Intel SGX side-channel attacks [Bra+17]. In addition, DyPS assumes that the SGX enclave is always available. For instance, enclaves can be subject to Denial-of-Service (DoS) attacks [TPV17; Che+17b]. Although those attacks do not compromise privacy, they might disrupt the pipeline and the expected behavior of the application. However, addressing this vulnerability falls out of the scope of this thesis.

4.4 Overview of DyPS

DyPS adopts a federated architecture that allows each biocenter to safely share genomes through a TEE-enabled server computing GWAS, while keeping the control of their own genomes (i.e. without revealing their data to other biocenters and by ensuring no privacy leakage from GWAS results). DyPS' architecture is illustrated in Figure 4.3. To ensure a secured computation potentially performed by untrusted machines, DyPS relies on a TEE which leverages custom microprocessor zones, to enforce isolation, confidentiality and integrity of both the data and operations. Periodically, the enclave collects the requests from the various biocenters and decides which requests are to be executed to safely and dynamically update the GWAS results. In the following, a discussion on how the TEE is utilized (Subsection 4.4.1), the workflow of DyPS (Subsection 4.4.2), how batches of requests are selected to produce safe test statistics (Subsection 4.5), how DyPS can dynamically increase the number of SNPs over which statistics are computed (Subsection 4.5.2), and the production of aggregate statistics from the selected requests (Subsection 4.5.4), are presented.

4.4.1 TEE-based architecture

DyPS uses Intel SGX enclaves that can be attested to prove that the code running inside it is the one intended, and that it is running on a genuine Intel SGX platform. Once attested, enclaves can be provisioned with secret data by using authenticated

secure channels. Moreover, enclaves can persist secret data outside the trusted zone by using a sealing mechanism.

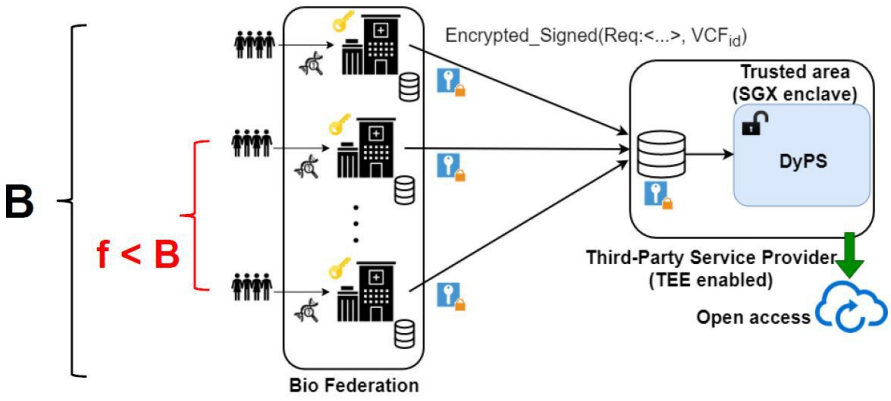


Figure 4.3: DyPS' federated architecture.

Once DyPS' enclave has been initialized, each biocenter executes a remote attestation procedure to authenticate it and establish a secret symmetric key. The biocenters sign their data with their private key, encrypt it with the shared symmetric key and send it over the network to the enclave's host. Upon reception of the encrypted data by the untrusted host, the enclave loads it into its protected memory space and decrypts it.

As time goes by, the biocenters are expected to sequence genomes, and might receive participation consent withdrawals from donors. For each genome addition or removal, the biocenters send a request to the enclave. This request $\langle bioc_{id}, g_{id}, seq_{id}, pop, op, \texttt{VCF}_{id}\rangle$ contains the biocenter ID ($bioc_{id}$), the donor's pseudonym (g_{id}), the operation sequence number of the biocenter (seq_{id}), if the donor belongs to the control or case population (pop), whether the genome should be added or removed ($op \in \{Add, Rmv\}$), and the corresponding genotype data following the Variant Cell Format file format ($\texttt{VCF}_{id}$) in case of genome addition.

4.4.2 Workflow diagram

GWAS can produce both test statistics and aggregate statistics. DyPS follows the workflow depicted in Figure 4.4 to ensure safe releases in both cases. This workflow is executed in the enclave and contains multiple modules: (1) the pending requests queues, (2) the request selection to produce safe test statistics, (3) the GWAS processing, and (4) the test to produce safe aggregate statistics.

(1) FIFO pending requests pool. DyPS maintains FIFO queues of genome additions or removals for each biocenter. DyPS tries to execute the received requests

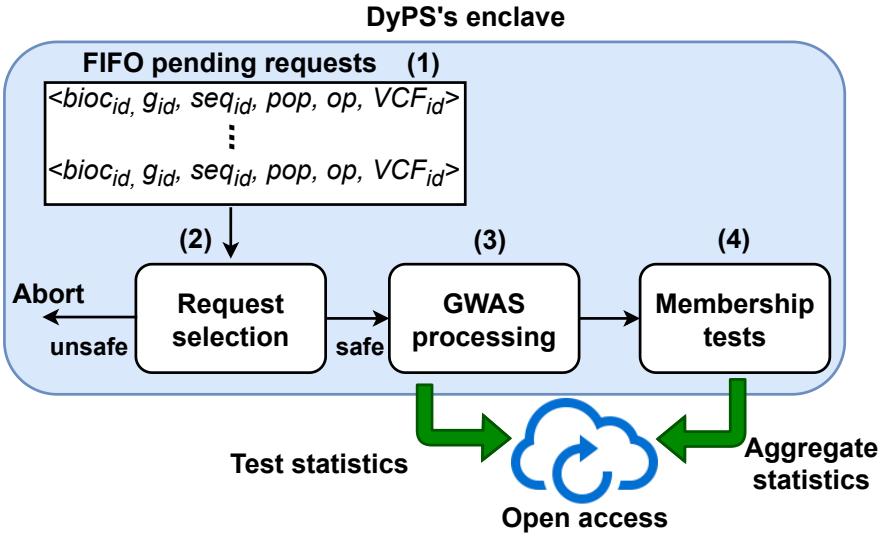


Figure 4.4: DyPS' workflow diagram.

according to their initial ordering by the biocenter through the use of their sequence number. However, DyPS might not always treat requests across the FIFO queues (e.g., because a genome A cannot be immediately removed, while a genome B can be added). A particular case occurs when a request to remove a genome that has not yet been added to the GWAS is received, in which case, both requests can immediately be executed by removing them from the FIFO queues.

(2) Requests selection. During this phase, DyPS aims at identifying a subset of genome operations that can be safely executed to update the GWAS results. To avoid exhaustive search of safe operations, DyPS assembles batches of requests where each set of non-colluding biocenters $(B - f)$ contributes more genome additions than removals and has sufficiently enough requests. The genome additions and removals to be executed are selected according to their FIFO order. The algorithm that DyPS is detailed in Subsection 4.5 and prove that it prevents all privacy leaks on test statistics. If no set of requests can be processed to release statistics, the process aborts. Since DyPS is periodically executed, requests are eventually processed.

(3) GWAS processing. After collecting a batch of requests that verifies the safe conditions for a GWAS release, DyPS computes the GWAS statistics over the overall remaining genomes in the SGX enclave. If the GWAS only aims at releasing test statistic, at this point DyPS can safely release or update the publicly accessible results (i.e., skipping step (4)).

(4) Membership tests. If DyPS aims at computing aggregate statistics, this additional step verifies that membership attacks cannot be executed on aggregate

statistics. To do so, DyPS extends SecureGenome [San+09a] while performing additional operations that are presented in Subsection 4.5.4 to decide which SNP positions might have their statistics released.

4.5 Request selection to address test statistics releases

This section details how biocenters can add or remove genomes from the results of a GWAS assuming (for now) that the number of studied SNPs remains constant. Both operations, if not handled carefully, can create privacy issues when the released statistics are updated. For example, updating statistics by adding, or removing, a single genome might directly leak this genome to an adversary that would observe publicly released statistics.

An exhaustive search (i.e., a brute force approach) checking if any candidate set of selected requests combined with the sets of requests used in previous releases verify Equations (3), (4) and (5) in Table 4.1 is not practical and would require exponential time. To avoid this issue, DyPS waits to have received sufficiently enough requests from the biocenters to verify equations listed in Table 4.1. More specifically, DyPS uses equation (3) as it implies equations (4) and (5).

DyPS assembles batches of genome operations according to the FIFO ordering of requests, and so that a batch contains more additions than removals for every subset of $(B - f)$ biocenters, and an overall number of genome operations either equal to 0 or larger than $LtoN(L)$ - equation (3) of Table 4.1 - for every subset of $(B - f)$ non-colluding biocenters. In summary, DyPS enforces that a safe batch always has more genomes $LtoN(L)$ but also ensuring that the number of genome addition operations are larger than the number of removals, and that both addition and removal operations combined are larger than $LtoN(L)$.

The following Section 4.5.1 presents the pseudocode of DyPS' algorithm that uses the requests selection mechanism compositions to select a safe batch of genome operations when the federation is facing up to $B - f$ colluding biocenters. From a high-level perspective, this algorithm works as follows. First, all pending addition requests of biocenters are selected, and for each of them, at most an equal number of pending removal requests. Then, the biocenters with the smallest number of selected requests are eliminated, until the $B - f$ biocenters with the least numbers of operations collectively possess enough requests (i.e., more than $LtoN(L)$) or until each biocenter has enough requests by itself. All requests from the biocenters that have not been discarded participate in the next release. This algorithm has a low complexity, since the previous statistic releases are not considered, while a brute force algorithm would have a exponential complexity with the number of

previous releases (as shown in the results of the experiments of Section 4.6.2). The GWAS test statistics can then be dynamically updated using the selected genome requests.

Next, Section 4.5.2 presents how DyPS can scale the number SNPs being considered by GWAS over time. This is an important property that would allow sooner releases of GWAS. Particularly, considering a larger number of SNPs in the beginning of a study demands a huge number of genomes to satisfy release conditions, e.g., millions of genomes, which might not be feasible for some federations. Hence, DyPS shows mechanisms to allow initial releases considering less SNPs and so less genomes, while also increasing the number of SNPs over time (as more genomes are added to studies).

Finally, Section 4.5.3 formally presents the conditions that DyPS uses to select a safe batch of genome requests, and it proves by induction that the adversary is never able to isolate test statistics where less than $LtoN(L)$ genomes participate in releases.

4.5.1 Pseudocode of DyPS requests selection mechanism

DyPS uses the algorithm reported in Algorithm 1 to select the biocenters that participate in a batch of requests to be executed. In the case of test statistics, all addition requests from a selected biocenter are selected, and a lower or equal number of removals. This algorithm assumes that up to f biocenters are colluding, with $f \leq (B-1)$.

DyPS first retrieves and binds the requests to their corresponding biocenters in FIFO order (line 7 to 9), and adds them to `bioList`. After gathering the requests from the biocenters, DyPS sorts the list of biocenters according to their number of addition requests (line 10), before selecting a batch of requests (lines 11 to 34).

The rationale behind the selection algorithm is to select a group of biocenters such that their combined requests cannot be attacked by the f biocenters that participate the most, and who might be colluding. From line 12 to 20, DyPS checks if the number of additions of the i selected biocenters are large enough considering the requests of f malicious biocenters and that this number is equal or greater than $LtoN(L)$.

If such a set of biocenters is not found, DyPS checks if some biocenters have enough requests to update statistics individually, considering Theorem 2 in the previous section and $LtoN(L)$ (lines 21 to 28). Note that during this step, the algorithm limits the number of removals per biocenter to the number of additions it is also executing. From lines 30 to 33, DyPS checks if biocenters were selected and adds them to the list of selected biocenters (`selectedBios`). Finally, the algorithm retrieves the selected biocenters and returns the sets of addition and

Algorithm 1 DyPS pseudocode for requests selection and test statistic releases.

```
1: procedure DyPS request selection algorithm(B, f, L)
2:     Input: B set of biocenters, f number of colluding players, L number of SNPs.
3:     Output: sets of selected genome addition and removal requests.
4:     Uses: NtoL(L), the minimum number of genomes required to update L SNPs.
5:     bioList = ∅;
6:     selectedBios = ∅
7:     for b in B do// retrieve pending requests from each biocenter in FIFO order
8:         bioList[b.id].add(b.toAddGenomes,b.toRmvGenomes);
9:     end for
10:    bioList.sort();// sort using the number of addition requests
11:    istart = −1;
12:    for (int i = 0; i < B; i++) do
13:        if (bioList[i].addCount == 0) then
14:            continue;
15:        end if
16:        if (bioList.size() - i > f and sumBioReq(bioList, i, bioList.size() − f − 1) ≥ LtoN(L) then
17:            istart = i;
18:            break;
19:        end if
20:    end for
21:    if (istart == −1) then // assemble all biocenters that individually have enough operations
22:        for (int i = 0; i < B; i++) do
23:            if (bioList[i].addCount ≥ bioList[i].rmvCount and bioList[i].addCount +
   bioList[i].rmvCount ≥ LtoN(L)) then
24:                istart = i;
25:                break;
26:            end if
27:        end for
28:    end if
29:    if (istart != −1) then // assemble the selected biocenters
30:        for (int i = istart; i < B; i++) do
31:            selectedBios.add(bioList[i]);
32:        end for
33:    end if
34:    // assemble the batch of requests from selected biocenters
35:    for (int i = 0; i < selectedBios.size(); i++) do
36:        Adds_Batch := selectedBios[i].addRequests
37:        Rms_Batch := selectedBios[i].rmvRequests
38:    end for
39: end procedure
40: // release test statistic over selected set of requests
41: computeTestStatistics(Adds_Batch, Rms_Batch)
```

removal requests that can be executed to update the GWAS test statistics (line 35 to 39).

In the case of aggregate statistics, the actual composition of the requests batch is determined by the dedicated SNPs selection algorithm presented in the next Section 4.5.4.

4.5.2 Scaling the GWAS over number of SNPs

So far, it has been assumed that statistics are computed over a static set of SNPs. However, as more genomes become available, DyPS can dynamically increase the number of SNPs over which statistics are computed, as illustrated in Figure 4.5.

The initial statistics release (i.e., release 1 in Figure 4.5) happens when the enclave can assemble a batch of N_1 genome addition requests such that $L_1 = LtoN(N_1) \geq 1$. DyPS then automatically decides the subsequent releases, based on the conditions of Table 4.1, as follows. Let us assume that the i-th release of statistics decided by DyPS covers L_i SNPs, and let $N_i = NtoL(L_i)$. The number of genomes N_{i+1} and the number of SNPs over which to release statistics L_{i+1} are determined as follows. First, N_{i+1} must verify $N_{i+1} - N_i > NtoL(L_i)$, which states that the statistics over the first L_i SNPs need to be sufficiently updated to preserve the privacy of the newly considered individuals over these SNPs. The value of L_{i+1} is then computed using $L_{i+1} = LtoN(N_{i+1} - N_i)$. This process is called a diagonal expansion (release 1 to 2 in Figure 4.5). The actual composition of the requests may contain additional genome removals. In that case, DyPS uses the methods that were defined previously for test and aggregate statistics over the SNPs considered both by release i and $i + 1$ to prevent privacy leaks.

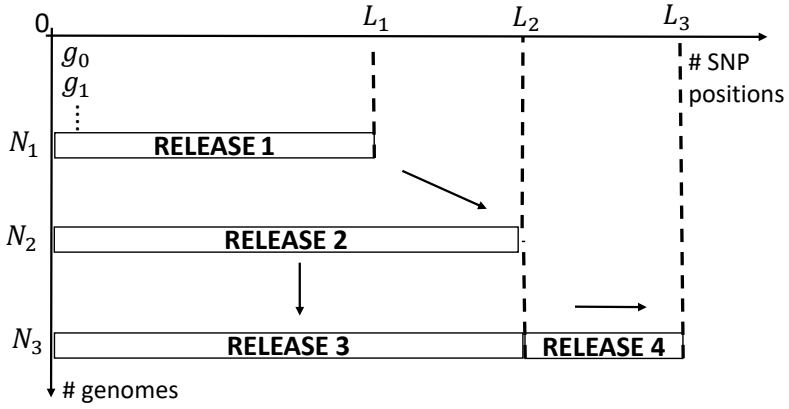


Figure 4.5: SNPs set dynamic scaling.

DyPS can also handle two additional special cases. The first one happens when $L_{i+1} = L_i$, which can happen when release L_i updated the full set of studied SNPs

(vertical expansion, release 2 to 3 in Figure 4.5). The second case happens when the value of L is increased by the system administrator. In that situation, the number of SNPs over which statistics are released might be increased immediately if the number of genomes added allows it (horizontal expansion, release 3 to 4).

4.5.3 Composition property and proofs for genome requests selection to protect test statistics releases

This section demonstrates that the method DyPS leverages to select the set of genomes to be added, or removed, from a GWAS release prevents both membership attacks on test statistics, and recovery attacks on both aggregate and test statistics. The reader can recall Table 4.1 for a summary of the equations that DyPS relies on to evaluate the conditions of feasibility of those attacks. More precisely, those equations state that there is a function such that a GWAS release that studies L SNPs can be considered secure if it used $LtoN(L)$ genomes. Similarly, one could invert these equations to discover how many SNPs L can be safely released according to the number of genomes N, namely $NtoL(N)$.

The following theorems and proofs describe how DyPS selects safe batches of genome requests, and the types of GWAS updates that can be performed according to the nature of the release. Let A_i be the set of genome additions and R_i the set of genome removals used for a specific release, or update, i of the GWAS results.

Theorem 1 (Vertical expansions with $f = 0$)**.** *If each release i is such that $|A_i| + |R_i| \geq LtoN(L)$ and that $|A_i| \geq |R_i|$, then each combination of releases involves more than $LtoN(L)$ genomes, which prevents an adversary to successfully launch a privacy attack.*

Proof. This theorem is proved by induction. The first release does not contain any genome removal and, therefore, adds more than $LtoN(L)$ genomes. The property to prove is then verified for the first release. Let us assume that this property is verified for any combination of releases whose ids are lower than or equal to i. Let j be the ID of the $(i+1)$-th release, which contains the additions and removals of genomes A_j and R_j. Let us consider a combination of releases whose ids are lower than or equal to i. This combination contains a set of genome additions A and a set of genome removals R. Now, if we were to combine this combination with release j, we would still obtain a secure release.

The number of genomes that an adversary can isolate by combining the releases is equal to $|R| + |A \setminus R_j| + |R_j \setminus A| + |A_j|$. By adding and removing $|A \cap R_j|$ to this expression one can obtain:
$(|R| + |A \setminus R_j| + |A \cap R_j|) + (-|A \cap R_j| + |R_j \setminus A| + |A_j|)$ (1).

The values of the two parts of the previous expression can be bounded thanks to the following two inequalities:

- By assumption, we have $|A|+|R| \geq LtoN(L)$ and, therefore, $|R|+|A \setminus R_j|+|A \cap R_j| \geq LtoN(L)$ (2)
- By construction, $|A_j| \geq |R_j|$, and therefore $|A_j| \geq |A \cap R_j| + |R_j \setminus A|$ (3)

We can now bound each term in (1). First, we already established (in (2)) that $(|R| + |A \setminus R_j| + |A \cap R_j|) = |A| + |R| \geq LtoN(L)$, which bounds the first term in (1). Second, by using (3), one can see that $-|A \cap R_j| + |R_j \setminus A| + |A_j| \geq -|A \cap R_j| + |R_j \setminus A| + (|A \cap R_j| + |R_j \setminus A|) \geq 2 * |R_j \setminus A| \geq 0$, which bounds the second term in (1). By adding these two inequalities, we finally obtain that $|R| + |A \setminus R_j| + |R_j \setminus A| + |A_j| \geq LtoN(L)$, which concludes. □

Theorem 2 (Horizontal expansions with $f = 0$)**.** *Let (A_i, R_i) be the set of genome additions and removals respectively executed during the GWAS results update i. The horizontal expansion algorithm allows an expansion of L SNPs and does not allow an adversary to successfully launch a privacy attack.*

Proof. By construction, the released set of L_{i+1} SNPs have been chosen so that $|A_i| + |R_i| \geq NtoL(L_{i+1})$, which prevents any genomic data over the SNPs that are newly considered in release $i + 1$ to be inferred. □

Theorem 3 (Diagonal expansions with $f = 0$)**.** *Let (A_i, R_i) and (A_{i+1}, R_{i+1}) be the sets of genome additions and removals respectively executed during the GWAS results updates i and $i + 1$, between which a diagonal expansion occurred. The diagonal expansion algorithm does not allow an adversary to successfully launch a privacy attack.*

Proof. A diagonal expansion can be seen as a combination of vertical and horizontal updates, which are respectively proven safe in Theorems 1 and 2. □

In the following, let $A_{i,S}$ be the set of genome additions and let $R_{i,S}$ the set of genome removals used for a specific release, or update, i of the GWAS results by a set S of players (i.e., biocenters). Now, it is shown how the previous results are applied to the situation where up to f of the B biocenters might be colluding.

Theorem 4 (Releases with $f \neq 0$)**.** *For any set S of $(B - f)$ non-colluding biocenters, if each release i either verifies:*

- $|A_{i,S}| + |R_{i,S}| \geq LtoN(L)$ *and* $|A_{i,S}| \geq |R_{i,S}|$, *or*
- $|A_{i,S}| = 0$ *and* $|R_{i,S}| = 0$,

then each combination of releases involves either no genomes from the $(B - f)$ biocenters at all, or more than $LtoN(L)$ genomes, which prevents an adversary to successfully launch a privacy attack.

Proof. A release that does not contain any additions or removals cannot leak any private information. We can therefore only reason about combinations of releases that each satisfy the first condition that was listed. This Theorem is therefore a direct consequence of Theorem 3, if one considers the genomes that have been released by a given subset of $(B - f)$ biocenters. □

4.5.4 Membership tests to address aggregate statistics releases

DyPS only includes in the GWAS results of the SNPs which depicted safe aggregate statistics (i.e., that would now allow membership inference). Similarly to safe test statistics, preventing recovery attacks from aggregate statistics relies on enforcing a minimum batch size of genome (i.e., Equation (3) in Table 4.1). However, to provide aggregate statistics while also preventing membership attacks, the current batch of requests and its combinations with previous releases must verify conditions (1) or (2) of Table 4.1. To ensure these conditions, DyPS relies on metrics that bound the identification power achievable over a set of requests (genomes) and GWAS result to identify over which SNPs to update the GWAS results. DyPS extends SecureGenome for this verification.

More specifically, given a set of genomes, a set of SNPs, and a set of control genomes (the adversary knowledge), DyPS determines which SNP positions can have their allele frequencies safely released by firstly removing very rare allele frequencies (MAF ≤ 0.05), and SNPs in high linkage disequilibrium (*p-value* below 10^{-5}) among the participating SNPs. After this step, DyPS computes and evaluates the detection power achieved by the singlewise LR or pairwise Λ in order to decide over which SNPs aggregate statistics can be safely released.

Identifying a set of SNPs to update in a given batch of genome operations however does not guarantee that the privacy of each genome will never be breached. Indeed, any release of aggregate statistics can be combined with past releases, and the genomes that a subset of up to f colluding biocenters contributed can be removed from the resulting aggregate statistics.

To verify whether a SNP's single or pairwise allele frequencies can be updated within a batch of genome operations, DyPS executes an exhaustive verification. More precisely, for a given SNP, this exhaustive verification (i) gathers all releases where statistics over the selected SNP have been released, and (ii) verifies that any combination of these releases have a low enough LR (i.e., identification power rate) score for the given SNP for any combination of up to f adversary biocenters. The complexity of this verification scheme is $O(L' \cdot 2^R \cdot \binom{B}{f})$, where L' is the number of selected SNPs in the current candidate batch, and R is the current number of releases. In practice, one could simply tune DyPS to limit the maximum number

of releases it wishes to avoid spending too much time performing verification.

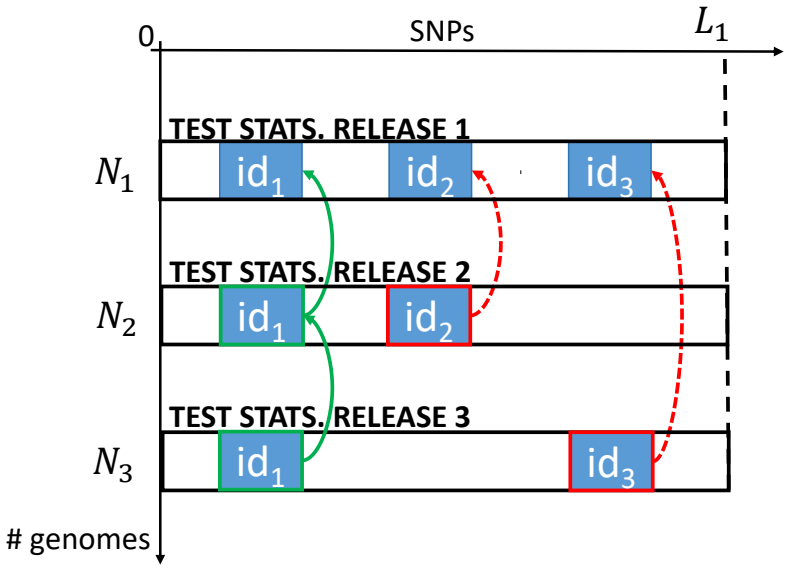


Figure 4.6: Successive releases of test and aggregate statistics as new genome addition or removal requests are executed.

Figure 4.6 illustrates some scenarios DyPS might face with aggregate statistics. In this example, DyPS determined a batch of N_1 genomes over which statistics might be released according to the method we defined for test statistics, over L_1 SNP positions. Using the SNP selection algorithm, DyPS determines that aggregate statistics over a subset (id_1, id_2, id_3) of those $L' = 3$ SNPs can be released (release 1). In release 2, following the same algorithm, DyPS determines that aggregate statistics can be released over $L' = 2$ SNPs (id_1 and id_2). DyPS then verifies whether each combination of previous releases with the current one still allows aggregate statistics to be released, i.e., the combination of releases 1 and 2. This verification passes for SNP id_1 (represented with plain arrows), which means that the statistic can be updated, while it does not for SNP id_2 (represented with dashed arrows), which cannot be updated during this release. Over release 3, DyPS determines that L' SNPs (id_1 and id_3) can be released over the selected genomes. However, the verification process identifies that the aggregate statistics cannot be released for SNP id_3 (because of the combination of releases 1 and 3, dashed arrow did not pass), while all verification pass for SNP id_1 that can be updated (all combinations of releases are not shown for simplicity).

The next Section 4.5.5 discusses the pseudocode of the algorithm that DyPS uses to perform the exhaustive verification analysis over existing release and selected SNPs. Posteriorly, Section 4.5.6 explains how DyPS extends the SNP se-

lection mechanism to pairwise statistics releases.

4.5.5 Pseudocode of DyPS exhaustive verification mechanism for aggregate statistics

At this point, **DyPS** had already selected a safe batch of requests to use to update test statistics, as explained in Section 4.5.

DyPS then checks whether aggregate statistics can be updated using the selected batch of requests. This process is shown in the algorithm reported in Algorithm 2.

Algorithm 2 DyPS pseudocode for aggregate statistic releases.

```
1:  procedure DyPS for Agg. statistic releases(S, Rels) // selected set of genomes, list of releases so far
2:      Input: set of genomes from selected biocenters.
3:      Output: set of SNP positions for safe aggregate statistics releases.
4:      Uses: SNPSelection(S, B, f): returns safe SNP positions for a set of genomes S and combinations of (B f) genome sets; shareGenomes(rel): checks if a release shares genomes with release rel; AllCombinations(Rel_shared_SNPS): create all combinations of releases that shares SNPs with Rels_shared_SNPS.
5:      safe_SNPs := SNPSelection(S, B, f)
6:      for s in safe_SNPs do // for each selected safe SNP
7:          for rel in Rels do// for each release so far
8:              if (s == rel.s) then // SNP position s has been released in a previous release rel
9:                  if (S.shareGenomes(rel)) then // candidate release S also shares genomes with previous releases rel
10:                     Rel_shared_SNPs.add(rel)
11:                 end if
12:             end if
13:         end for
14:     end for
15:     for combRel in AllCombinations(Rel_shared_SNPs) do
16:         testSet := combRel + S // merge participating genomes in both releases
17:         checkSafeSNPs := SNPSelection(testSet)
18:         if (s in checkSafeSNPs) then
19:             continue // this SNP can be released
20:         else
21:             safe_SNPs.del(s) // cannot find a safe release this SNP this round
22:         end if
23:     end for
24:     return safe_SNPs // set of safe SNPs for candidate release S
25: end procedure
26: computeAggStatistics(safe_SNPs) // compute and release aggregate statistic over the set of selected SNPs
```

DyPS first collects the safe SNP positions that can be released given the selected genomes and all combinations of genomes considering up to f adversary biocenters (line 5). For each selected SNP, **DyPS** then collects the previous releases where it has been previously updated, and checks whether they involved genomes that participate in the current batch of genomes (lines 6 to 14). **DyPS** then generates and loops over all combinations of releases that share the same SNPs (lines 15 to 23). For each possible combination, **DyPS** executes the SNP

selection software over the resulting set of genomes. If the SNP position s in $safe\_SNPs$ is also found to be safe in $testSet$, it means that it can be updated, otherwise, it is ignored. In the end, DyPS has a list of safe SNP positions to be updated (line 24).

4.5.6 Verification for pairwise statistics releases

In order to extend the mechanism that finds the list of SNPs over which statistics can be released using a batch of genome operations from singlewise frequencies to pairwise frequencies, DyPS considers the best of the SecureGenome [San+09a] approach and pairwise-based LR-tests T_r [Wan+09] and/or Λ [Zho+11]. On singlewise frequencies, DyPS uses SecureGenome's strategy to remove SNPs in linkage disequilibrium, and very rare SNPs (i.e., SNP positions with MAF ≤ 0.05). After removing those SNP positions, DyPS runs the singlewise-based LR-test to identify the safe SNP positions.

In addition, DyPS can also verify the probability of re-identifying participants leveraging pairwise-based LR-tests (using the T_r and/or the Λ metrics) in order to decide which pairs of SNP positions can also have their pairwise frequency safely released.

The exhaustive verification for pairwise frequencies follow a similar scheme, and can be executed in parallel of the verification for singlewise frequencies. It is important noticing that T_r and Λ provide a membership metric for safe releases of pairwise statistics. However, they do not apply any SNP pruning mechanism (i.e., removal of dependent and very rare SNPs). Therefore, DyPS extends the state-of-the-art by not only offering a mechanism to safely release both types of allele frequencies but also accomplishing it in a dynamic fashion.

4.6 Experimental evaluation

4.6.1 Experiments setup

It was used both Windows 10 Enterprise and an Ubuntu 18.04 LTS in a 64-bit machine, equipped with 16 GB of RAM and an Intel i7-8650U @ 2.11.GHz, which supports Intel SGX. DyPS's performance is evaluated under several scenarios using simulated and real genomes. DyPS' code was run both in Java using Java JDK 12.0.1 and Eclipse IDE (4.11.0), and inside an Intel SGX enclave using Graphene SGX [TPV17] to implement DyPS in C++, so that it can run inside the SGX enclave. DyPS uses AES 256 to encrypt messages, and ECDSA for signatures. When it executes the remote attestation procedure, a biocenter agrees on a key with the enclave, which it uses to encrypt and sign the data it sends to the enclave,

while the enclave can verify it upon reception.

During the experiments, each round represents the moment when biocenters generate genome operation requests that are sent to the enclave, and the enclave tries to generate a GWAS statistics release. In real settings, rounds would typically have a one day duration. The biocenters use a Poisson distribution to generate genome addition or removal requests. The parameters of these Poisson distributions were set so that biocenters generate more genome additions than removals since it is expected to reflect reality. For the experiments, based on simulated or on real genomes, it was assumed a default $\lambda = 8$ for additions, and $\lambda = 6$ for removals as default. For larger GWAS settings, the value for λ has been proportionally increased. These values were adopted based on the increasing rate of genome sequencing, and the growing concern about genome privacy risks among society nowadays. For example, Dankar et al. [Dan+20] have recently evaluated and claimed the need for the creation of dynamic information consent models capable of autonomously enabling individuals to opt out of participating in genomic studies at any time.

The performance of DyPS' request selection heuristic was compared to a brute force (*BF*) mechanism, and a naïve algorithm. The *BF* approach aims at adding or removing genomes by assembling batches of genome operations, and checking whether they are safe by combining them with all previous combinations of data releases. DyPS scales better than the *BF* algorithm with the number of data releases by avoiding this brute force verification for test statistics. The naïve approach waits for biocenters to collectively have $LtoN(L)$ genomes (additions or removals) when performing a release. This method can be seen as the current state-of-the-art, and does not allow genomes removals. It is shown that this method executed with genome removals leads to privacy risks.

Regarding aggregate statistics, since DyPS is the first dynamic GWAS protocol, it could only be compared to a static SOTA GWAS algorithm, which would wait for all requests to have been collected before releasing statistics. To do so, DyPS is compared with a static approach that would rely on the LR metric [San+09a] to release singlewise allele frequencies at the end of the experiment. For this experiment, the default LR parameter values defined in [San+09a] were used: a MAF cut-off of 0.05, a LD between SNPs cut-off of 10^{-5}, a false positive rate of 0.1, and a true positive rate of 0.9.

DyPS was stressed by simulating the generation of up to 6 million genome requests studied over up to 300K SNPs to evaluate its performance with test statistics. In addition, DyPS was also evaluated using two real genome datasets to evaluate its performance on both test and aggregate statistics: the idash2017 dataset [Pri], which consists of real 2,000 genomes, and the phs001039.v1.p1 dataset from dbGAP [Wal+11] of an Age-Related Macular Degeneration study, which con-

sists of 14,860 case genomes and 13,035 control genomes.

4.6.2 Bandwidth, CPU and memory consumption

DyPS uses 64-bits integers to encode the ID fields in a request, except for the *pop* and *op* fields that only require one bit. Overall the size of a request is 258 bits, which represents approximately 48 Bytes after encryption. A genome is encoded using 2 bits per SNP, which would represent only 75 KB for a GWAS studying 300,000 SNPs. Given those numbers, bandwidth is not a bottleneck for DyPS since communications only occur when biocenters asynchronously send requests to the enclave.

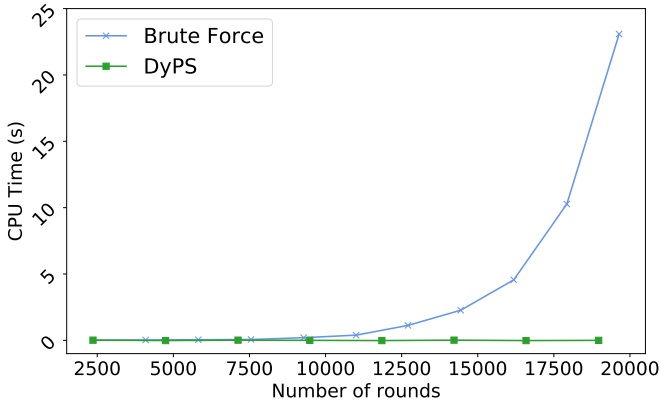


Figure 4.7: Running time of the brute force and DyPS request selection approach for test statistics over 5,000 SNPs ($LtoN(L) = 38,040$) ($B = 4, f = 0$).

To measure DyPS' CPU and memory consumption, it was considered a GWAS scenario that involves 4 non-colluding biocenters ($B = 4$, $f = 0$). Figure 4.7 shows the CPU running time of the brute force (***BF***) and of DyPS' request selection algorithms during a synthetic GWAS that involves 5,000 SNPs and 20,000 rounds. DyPS' SNP selection algorithm has a constant complexity and is very fast (less than 350 ms), while the ***BF*** selection algorithm, with exponential complexity, requires more than 20 seconds after 20,000 rounds.

Figure 4.8 shows DyPS' performance when deployed in an enclave. Every point represents a release that took place during the experiment. As can be noted, DyPS's selection algorithm for test statistics has a constant running time (and below 1 ms). In addition, the release construction running time varies according to the number of requests. The longest release construction running time was below 500 ms in a release with 38,059 genome operations, and the average during the experiment was 269 ms. With similar settings, DyPS was executed inside an SGX enclave for conducting a GWAS studying 300,000 SNP positions and with

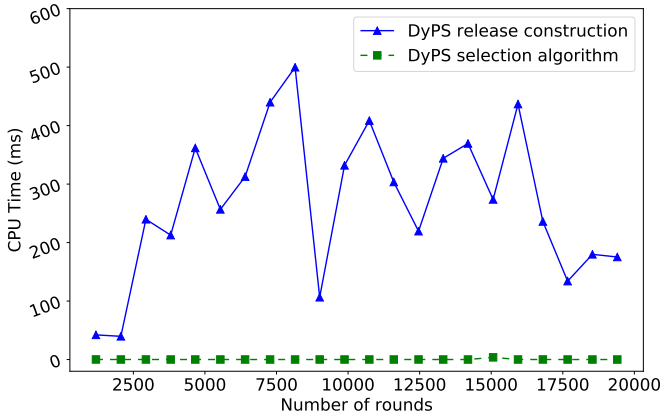


Figure 4.8: Running time of DyPS request selection approach for a test statistics over 5,000 SNPs ($LtoN(L) = 38,040$) inside the SGX enclave ($B = 4, f = 0$).

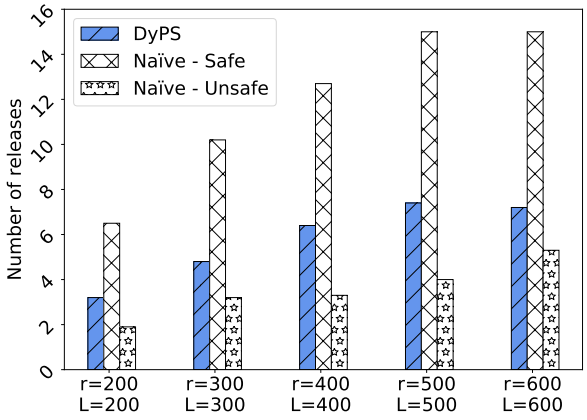
four biocenters. Although **DyPS** was able to add 6,078,551 genomes and to remove 899,278 genomes, it was observed similar behavior for the CPU running time.

DyPS' memory consumption in the SGX enclave was also monitored assuming different system settings (i.e., varying number of B and f). A significant change in memory consumption was not observed, which stays around 2 MB per round when the numbers of participating and colluding biocenters evolve. This is expected, since genomes are stored encrypted outside of the enclave and at a given time, only a limited number of genomes are loaded into the enclave's memory to be processed.

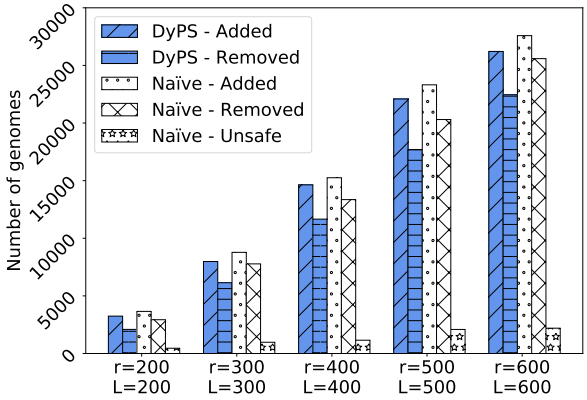
4.6.3 Naïve dynamic release vs. DyPS

Now, **DyPS**' releases performance in terms of number of releases and number of requests is compared to a naïve approach. More specifically, the number of releases that can be performed by the naïve approach, which updates GWAS results as soon as more than $LtoN(L)$ genomes are collected without making sure that the combination of these releases are also safe, to the number of releases that **DyPS** performs. Figure 4.9 reports the number of releases for both approaches, where the label of a bar plot reports the number of rounds for which the experiment ran (i.e., the value of r), and the number of SNPs per GWAS (i.e., L).

Figure 4.9a shows the corresponding number of releases done by each approach, and for the naïve approach shows the number of releases for which a genome was at risk. Up to 4.98% of the releases contained at least one genome that was at risk. Figure 4.9b shows the number of genomes that were added, or removed by each method during the experiments. It also shows how many genomes were at risk with



(a) Average number of releases.



(b) Average number of added, removed and unsafe genomes.

Figure 4.9: Comparison between the naïve release approach and DyPS under different scenarios (r rounds, and L SNPs) for ($B = 4, f = 0$).

the naïve release approach. Overall, the naïve approach was only able to consider at most 11% more genomes than **DyPS**, and exposed up to 8% of the genomes to privacy leaks (i.e., recovery attacks). Even though **DyPS** updates the GWAS results less frequently, which is to be expected, it is overall of low consequence on the number of considered genomes. In addition, **DyPS** enforces that no genome is at risk.

4.6.4 Impact of dynamic SNP-set scaling

To measure how **DyPS**' scaling mechanism approach reduces the treatment delay of requests, the effect of the dynamic SNP-set scaling mechanism was measured.

In particular, it was simulated a situation where the number of rounds is limited, and L (the number of SNPs) is large, such that DyPS without dynamic scaling could create only a limited number of releases whereas DyPS was able to create more and earlier releases.

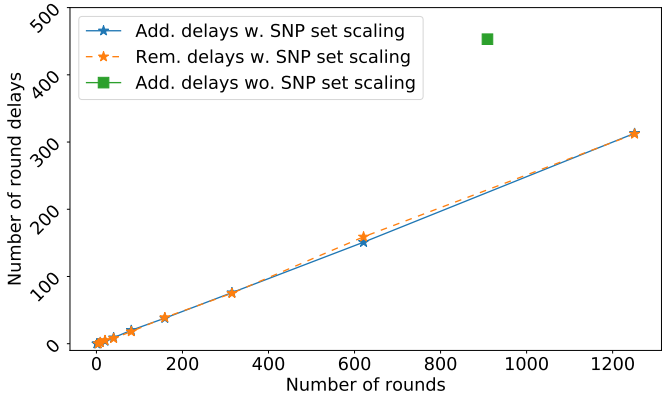
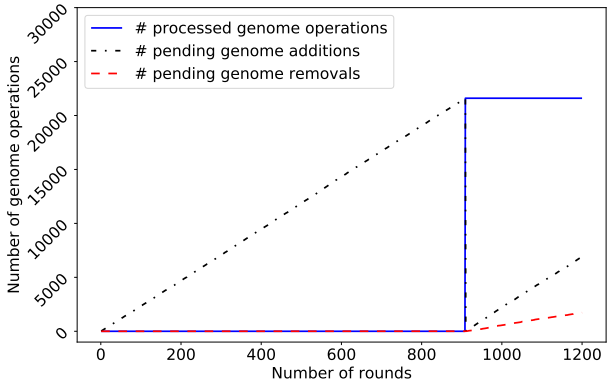


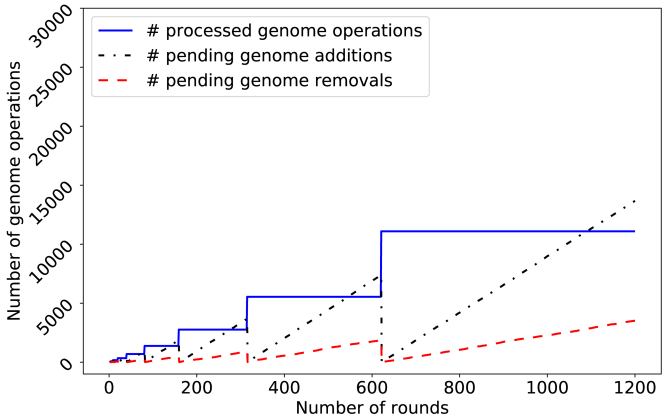
Figure 4.10: DyPS with or without dynamic SNP-set scaling - round delays for a GWAS consisting of 3,000 SNP positions ($LtoN(L) = 21,600$) ($B = 4, f = 0$).

Figure 4.10 provides the round delays for both approaches per type of genome operation (addition or removal). DyPS without the dynamic release mechanism was able to perform a ***single*** release (additions of 21,602 genomes), represented by the square at the 908th round (with an average round delay above 430 rounds), during the whole experiment, but could not remove any released genomes. On the other hand, DyPS with dynamic scaling was able to execute 11 secure releases, varying among diagonal, vertical and horizontal releases. One can also observe that DyPS treated genome additions and removals with very similar delays (the star markers), and that the dynamic mechanism has an order of magnitude lower delay. One can also notice that the time to release of requests increases as the number of SNP positions L increases over time, even when using dynamic SNP-set scaling since more genomes operations are required per batch of requests to ensure privacy.

It was then measured the number of pending operations, and the overall number of applied genome operations. Figure 4.11a and Figure 4.11b respectively report those numbers for DyPS with or without dynamic scaling. DyPS without dynamic scaling was not able to apply any removal requests during the experiment. Moreover, at the end of the experiment, 14,119 genome addition and 3,572 genome removal requests were still pending. In total, 21,602 genomes have been added and none were removed. In contrast, DyPS could publish more safe releases starting from the first round. The experiment was ended at the 1250th round to compare the performance of both approaches. At that point DyPS was able to add 22,179



(a) DyPS without dynamic scaling of the SNPs set.



(b) DyPS with dynamic scaling of the SNPs set.

Figure 4.11: DyPS without and with dynamic scaling of the SNPs set for a GWAS consisting of 3,000 SNP positions ($LtoN(L) = 21,600$) ($B = 4, f = 0$).

and to remove 7,580 genomes. At the end of the experiment, 19 genome additions and 2 genome removals remained, which represented a decrease of more than 99% for both cases when compared to **DyPS** without dynamic scaling.

The CPU running time of the two versions was also compared. It was noticed that **DyPS** with dynamic scaling has a slight increase of CPU time when compared to **DyPS** without scaling, due to the fact that it needs additional analysis to dynamically evaluate and safely decide the increasing of the number of SNPs over which statistics are released. However it still remains practical and a magnitude lower than the ***BF*** approach for the selection of requests (which does not consider a dynamic scaling scheme).

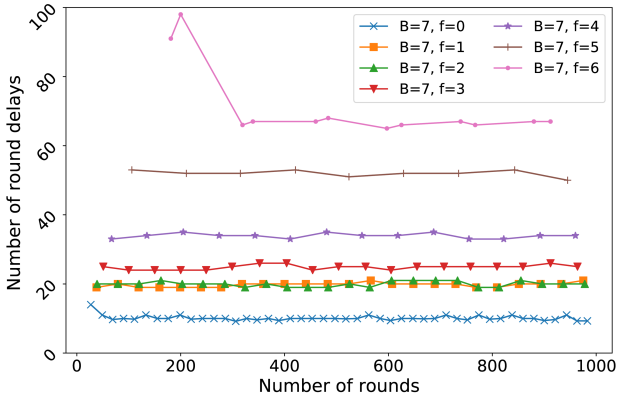
4.6.5 Impact of colluding biocenters

In this section, DyPS' performance with colluding biocenters is evaluated. For this experiment, it was considered an increased number of biocenters (seven biocenters) and several values for the number of colluding biocenters (f). This experiment was performed in the enclave. Figure 4.12 illustrates the pending rounds for addition and removal requests per update of the GWAS results for varying number f of colluding biocenters. As one could expect, when more biocenters collude the requests are applied with more delay, since it takes more time to assemble larger sets of requests, which are required for safety when facing more adversaries. For example, with the first threat model ($f = 0$), the average number of pending rounds was 10.13 and 10.11 for addition and removal requests, respectively. On the other hand, with $f = 5$ and $f = 6$, the average processing delays were equal to 52 and 71.25 rounds for additions, and 53 and 60.2 for removals, respectively. An interesting event happens in Figure 4.12b for the $B = 7, f = 6$ line where there is a very small delay at the 200th round. It is explained by the fact that a release was created with the genomes of a single biocenter, in the 181st round (first release in Figure 4.12a), and then after just 19 rounds, this same biocenter was able to execute removal requests at the 200th round, which explains the short delay.

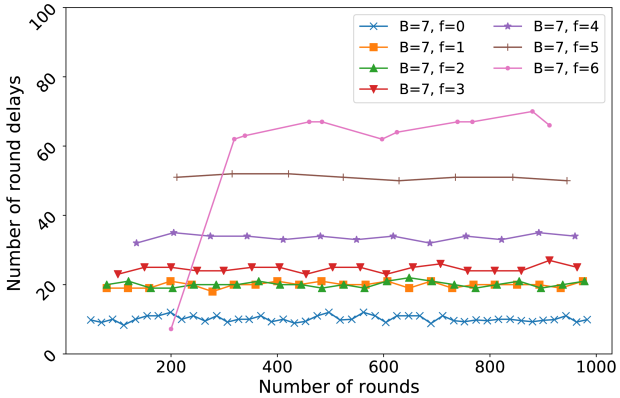
Table 4.2 details the number of genome additions and removals, the number of releases, and the average addition and removal request delays under several scenarios. As expected, fewer addition and removal operations are executed when facing stronger adversaries. However, the numbers decrease by at most 8.34% for the additions, and 26.57% for the removals when comparing without collusion ($f = 0$) to six colluding biocenters over seven ($f = 6$). The latter number is explained by the fact that DyPS does not currently wait to perform requests, and because removals require that sufficiently enough additions are simultaneously executed.

Table 4.2: Average number of processed addition and removal requests, number of GWAS releases, and average round delays for addition and removal requests depending on the number of colluding biocenters.

Threat Model (B, f)	***#Additions, #Removals, #Releases, #Add. round delays, #Rmv. round delays***
$B = 7, f = 0$	55,435 / 19,917 / 46 / 10.13 / 10.1
$B = 7, f = 1$	55,093 / 19,421 / 24 / 19.75 / 10.9
$B = 7, f = 2$	55,013 / 19,692 / 24 / 19.96 / 20.1
$B = 7, f = 3$	54,353 / 18,984 / 19 / 24.84 / 24.6
$B = 7, f = 4$	54,033 / 18,825 / 14 / 33.93 / 33.6
$B = 7, f = 5$	52,655 / 17,994 / 9 / 52.0 / 51.0
$B = 7, f = 6$	50,813 / 14,625 / 12 / 71.25 / 60.2



(a) Time to release in rounds for genome addition requests.



(b) Time to release in rounds for genome removal requests.

Figure 4.12: $(B - f)$ DyPS: Time to release in rounds for genome requests for a GWAS consisting of 300 SNP positions ($LtoN(L) = 1,598$) during 1,000 rounds, and different number of possibly colluding biocenters.

4.6.6 SNP selection for aggregate statistics

DyPS' releases of aggregate statistics depend on the SNP distribution among the set of added genomes. DyPS is first evaluated using the idash2017 dataset [Pri], which consists of 2,000 real genomes (1,000 case and 1,000 control). All the genomes in the control set was used as the adversary knowledge when launching the membership attack. For aggregate statistics, DyPS' release mechanisms require more extensive computations when the number of previous releases and when the number of genomes used per release increase (cf. Section 4.5.4). Therefore, it was primarily considered scenarios where $f = 0$, because it maximizes the number of releases and their size. This section considers a GWAS that studies a

small set of L SNPs (i.e., the top 10 most significant SNPs) so that releases are more frequent. Given these parameter values, it can be studied the worst case of the exhaustive verification procedure (i.e., more combinations to be checked).

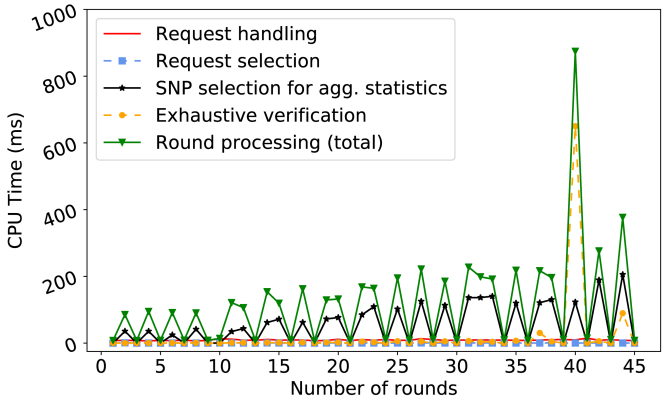


Figure 4.13: Running time for the different steps of DyPS execution for a GWAS studying 10 SNP positions ($B = 5, f = 0$). Reference group size: 1,000. Total number of real genomes used: 2,000.

Figure 4.13 details **DyPS**' CPU consumption during each round per category: request handling, request selection, SNP selection for aggregate statistic release, exhaustive verification, and total round processing. Each CPU running time peak happened when a round resulted in a release of GWAS statistics. **DyPS**' selection algorithm, which runs for every round, used less than 200 ms for almost each round, and less than 1 second overall, which is very reasonable given that sequencing a genome usually requires around 1 day. For the largest measured value, obtained in the 42nd round, **DyPS** verified more than 2,359,296 combinations of releases in the enclave. The largest part of the computation was used by the exhaustive verification process for aggregate statistics release.

4.6.7 DyPS vs. static release of aggregate statistics

Next evaluation compares **DyPS** with a state-of-the-art static release algorithm over a larger set (the top 1,000 SNPs) of the idash2017 dataset in Figure 4.14. More precisely, it was measured over how many SNPs both approaches are able to release aggregate statistics only, which enables frequent updates. The LR-metric [San+09a] was used to decide whether singlewise frequencies can be released. The static release method identified that singlewise frequencies could be safely released over 45 SNPs out of the 1,000 studied, using the full set of genomes remaining at the end of the experiment (i.e., in only one release - the dashed bar).

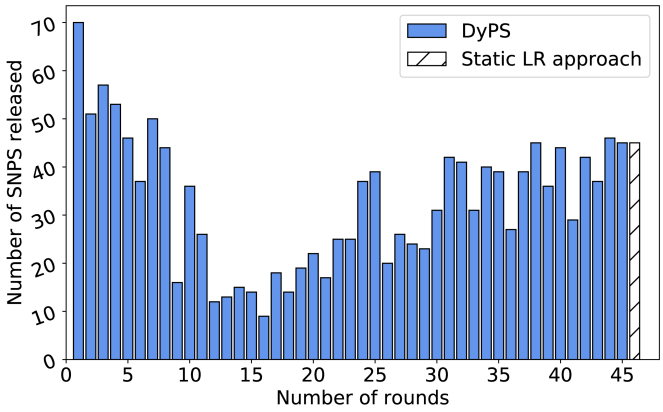


Figure 4.14: Comparison between a static approach and DyPS for releases of aggregate statistics for a GWAS studying 1,000 SNP positions ($B = 5$, $f = 0$). Reference group size: 1,000. Total number of real genomes used: 2,000.

On the other hand, the number of selected SNPs for **DyPS** varied according to the rounds, as expected, that is, in each different round, a different distribution of genomes participated in a release and, therefore, the set of SNPs over which statistics can be released evolves. The maximum number of SNPs **DyPS** released during the experiment was 70, while small releases, which do not prevent previous releases to be accessed, were more frequent. Note that in this experiment, **DyPS** released aggregate statistics after every round, while test statistics would not have been updated as frequently. Therefore, one can conclude that **DyPS** not only provides more frequent releases but also over a largest set of SNP positions. In a similar scenario with 350 SNPs, **DyPS** was able to release 2.6 times more statistics (i.e., multiple safe releases with at most 44 SNPs instead of 17 only released once).

Figure 4.15 illustrates the results of the CPU running time. As can be noted, in the 44th round a more extensive verification took place, which has checked 4,972,331 combinations of releases in total. The running time was 1,467 milliseconds. Furthermore, comparing to Figure 4.13 experiment with 10 SNPs, there is now an expected slight increase in the SNP selection software algorithm running time (star marker line) because now a larger number of SNPs have been checked. Nevertheless, even increasing the SNP-set size by 100x times (from 10 to 1,000 SNPs scenario, the longest time have just approximately doubled in average (200 to 500 ms). Similarly, the longest time for the exhaustive verification, took approximately 2,000 ms in the 42nd round.

As expected, the presence of colluding biocenters on aggregate statistics impacts the performance of **DyPS**. In particular, when facing colluding biocenters, **DyPS** has to evaluate more combinations of genomes considering potential combinations colluding biocenters might form. This results in an increased running time

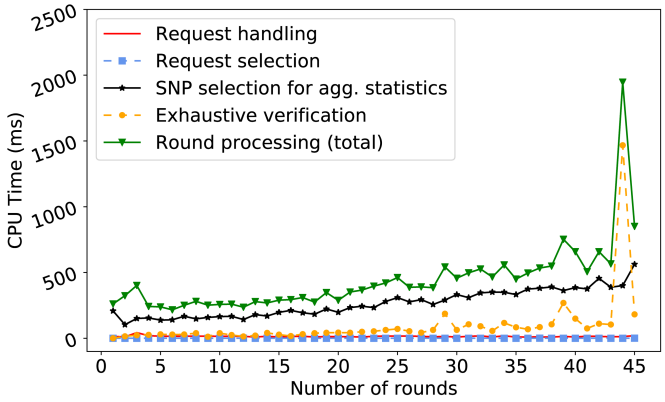


Figure 4.15: Running time for the different steps of DyPS for a GWAS studying 1,000 SNP positions ($B = 5$, $f = 0$). Reference group size: 1,000. Total number of real genomes used: 2,000.

due to the additional verifications. Considering the same scenario, and now using f equals to $(B - 1)$, we can notice that **DyPS**'s running time increases slightly. Figure 4.16 shows the computation time required by **DyPS** when the number of colluding biocenters is either 0 or $B - 1$, where $B = 5$. In particular, in this new experiment the peak and average running time for $f = 0$ were 1,612 and 896 milliseconds, respectively. Whereas for the $f = 4$ case, it took 2,348 and 1,212 milliseconds.

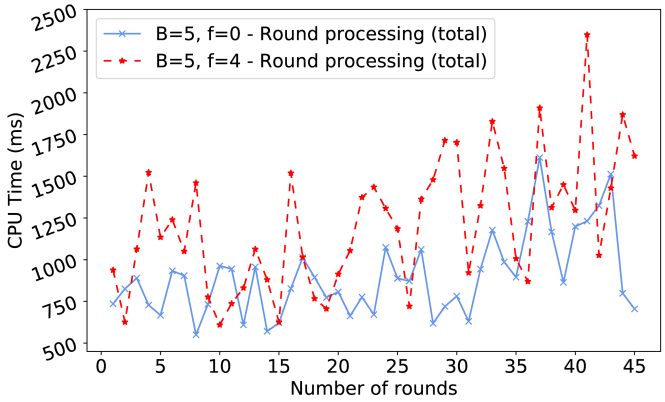


Figure 4.16: Running time for $f = 0$ and $f = 4$ with aggregate statistics computed over 1,000 SNP positions. Reference group size: 1,000. Total number of real genomes used: 2,000.

4.6.8 DyPS over a large-scale GWAS

DyPS' performance was also evaluated over another another real genome dataset, dbGAP *phs001039.v1.p1* [Wal+11], which consists of more than 35,000 genomes from which 27,895 could be used under the General Research Use (GRU) consent, of which 14,860 genomes are cases and 13,035 are controls, respectively. The SNP positions that appear in both cohorts are studied. The chromosome 1 was considered for this experiment as it is the chromosome with the largest number of remaining SNPs. It was considered 5,000 SNP positions to evaluate DyPS's algorithm over this larger dataset. Besides, both the addition and removal lambda parameters were multiplied by 16 ($\lambda = 128$ for additions and $\lambda = 96$ for removals), so that more genome operations are generated per round. It was considered the whole control dataset (13,035 genomes) as the adversary reference group.

Figure 4.17 shows the performance of DyPS over this larger dataset. Overall, 12,418 genomes were added, and 5,120 genomes were removed. Compared to the experiment in Figure 4.15, this experiment had an expected longer running time (average of 207 seconds, and peak of 2,500 seconds, when a cohort made of more than 27,800 genomes was studied) for the SNP selection algorithm because of the larger genomes cohort (i.e., more participating genomes and a larger reference group). On the other hand, Figure 4.17 shows a shorter running time for the exhaustive verification step due to a smaller number of SNP positions over which statistics could be released. This was because most of the SNP positions were being filtered out by the linkage disequilibrium and MAF step of DyPS over previous rounds.

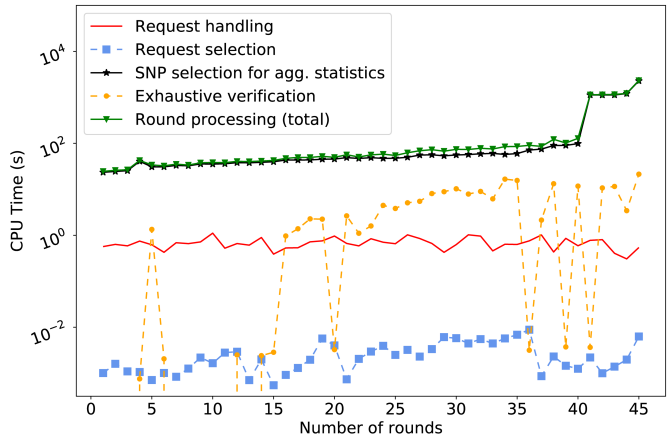


Figure 4.17: Running time for the different steps of DyPS execution for a GWAS studying 5,000 SNP positions ($B = 5, f = 0$). Reference group size: 13,035. Total number of real genomes used: 27,895.

In addition, in scenarios where **DyPS** deals with a larger number of SNPs, it could rely on existing SNPs batching mechanisms inside enclaves, such as [Che+16b], in which SNPs are firstly separated in batches of equal size, and processed separately. Later, SNPs in different batches are processed in a crossed-over manner in order to keep a global set of safe SNPs updated as batches are processed.

Table 4.3: DyPS' average memory consumption inside the enclave depending on several controls group sizes and 5,000 SNPs.

(#B / #f) **DyPS inside enclaves**	**Average Memory Consumption (KB)**
($B = 5/f = 0$), controls: 1,000	2,160
($B = 5/f = 0$), controls: 5,000	2,224
($B = 5/f = 0$), controls: 7,500	2,228
($B = 5/f = 0$), controls: 10,000	2,228
($B = 5/f = 0$), controls: 13,035	2,228

DyPS' memory consumption is reported in Table 4.3 when different sizes of control groups (i.e., adversary reference groups) are used. The memory consumption of the machine was identical to the one reported in previous experiments, i.e., around 2 MB. **DyPS** can therefore support a large number of genomes and SNP positions without significant performance penalty. Overall, the memory consumption stays well below the theoretical 128 MB (of which only 96 MB is usable without paging [CD16]) memory limitation of SGX enclaves.

Chapter 5

Privacy-preserving Interdependent GWASes (I-GWAS)

Chapter 4 offered methods to achieve dynamic, privacy-preserving and secure federated processing properties to enable several properties of *practical GWAS*. Nevertheless, the presented contributions are still insufficient to cover all the properties of *practical GWAS* that are envisaged in this thesis. This chapter addresses the remaining challenges. Namely, identifying new safe conditions for the creation of dynamic safe releases under the presence of interdependent GWASes, which encompasses the property (iv) of *practical GWAS*, i.e., offering solutions to enable safe releases of potentially overlapping GWASes.

The decreasing genomic sequencing costs motivated a trend towards sharing the results of independent GWASes on different phenotypes to construct multi-omics datasets [Im+12]. As a consequence, some studies might consider the same SNP positions and/or use the same genomes that have participated or will be used in other studies. For example, a particular individual whose genome is present in the control population of several studies or a certain individual that coincidentally participated in two different GWASes belonging to the case population, e.g., a person that has diabetes and high blood pressure and participated in two studies separately. This problem can also be extended to a scenario where an external adversary can observe overlapping GWASes releases from several federations, and then being able to circumvent the conditions that were presented to safely release single-GWAS results (see Chapter 4).

This chapter argues theoretically and confirm experimentally that both naïve and individually-safe releases enable privacy attacks in such multi-GWAS settings. For instance, it is shown that an adversary exploiting as little as two GWASes, can learn about the participation of involved individuals by performing a membership attack if no further protection is enforced. In addition, the adversary can reconstruct genetic variations of up to 28.6% of the participants by launching a

recovery attack, even if individually each GWAS release is safe. Furthermore, it shows that overlapping studies need to follow new conditions for safe GWASes releases, otherwise, private genomic data can be leaked even from the observation of "safe" single-GWAS releases. In particular, it contributes precise conditions under which interdependent GWAS preserve the privacy of the individuals who share their genetic data.

To address these issues, this chapter introduces **I-GWAS** as a privacy-aware solution for releasing to the public the results of interdependent and dynamically updated genome-wide association studies. **I-GWAS** successfully prevents privacy risks from such interdependent GWASes by discarding from the studies only those genetic variations that are vulnerable to membership inference attacks, and by selecting safe batches of genomes for the requested GWAS that can be safely used while mitigating recovery attacks. Moreover, **I-GWAS** similarly prevents membership attacks thanks to the SecureGenome LR-test, which provides an upper bound for the likelihood to learn about the membership of an individual in the case group. Nonetheless, such a LR-test is conducted in a crossed-over manner so that several combination of exposed SNPs and genomes among several studies are checked. **I-GWAS** uses the same TEE-enabled architecture of **DyPS**, and therefore can be seen as an extension of **DyPS**' protocol so that the GWAS federation can also conduct and releases multiple GWASes at same time.

5.1 I-GWAS' system and threat models

I-GWAS system and threat models slightly differ from **DyPS**. In particular, **I-GWAS** assumes that federations conduct several GWASes at the same time. These multiple studies might share some genomes, or consider the same SNPs. **I-GWAS**'s models are described in the following:

System model. **I-GWAS** assumes a similar scenario as **DyPS**. However, it assumes dynamic releases of multiple and potentially overlapping GWASes. Therefore, there is a federation comprised of B biocenters, $\{bioc_1, \cdots, bioc_B\}$, each sequencing genomes and requesting the addition to or removal of individuals from the federation. Now, in contrast to **DyPS**, the federation jointly operates on a set of P phenotypes $\{p_1, \cdots, p_P\}$, which they study using several GWASes. Each p_i represents a study with a corresponding set of SNPs and genomes, possibly added or removed dynamically to continuously update results. All genome operation requests sent by the biocenters are treated in FIFO order. All requests are evaluated in rounds, and **I-GWAS** certifies that a safe batch of requests (that meets the interdependent GWASes criteria) can be selected. If a safe batch of genomes cannot be found in a round, **I-GWAS** aborts the round. Eventually, a safe release will take place as new genome requests come over time.

Similar to DyPS, it is assumed that all GWASes release allele frequencies or test statistics, and such statistics are only released after the evaluation of privacy risks in interdependent and heterogeneous GWASes are evaluated.

I-GWAS gives the conditions under which such operations preserve the privacy of the individuals who share their genetic data for the purpose of public releases of GWASes. I-GWAS also receives all genome data and requests in an encrypted form and performs the safe interdependent release conditions analysis inside the TEE enclave. In summary, I-GWAS determines the minimal number of genomes each GWAS should use for any type of statistics to prevent recovery attacks, and performs additional checks on the actual statistics to release to prevent membership attacks. Moreover, I-GWAS assume that no information leaves the TEE before I-GWAS explicitly releases it and that cryptographic primitives are secure. In other words, resulting statistics are only released after the evaluation of privacy risks in interdependent and heterogeneous GWASes are evaluated. I-GWAS precises the conditions under which interdependent GWAS preserve the privacy of the individuals who share their genetic data.

Threat model. I-GWAS assumes that all biocenters in the federation are trusted to follow the protocol and to provide high-precision data [Pas+21; Zha+18; Sad+18; Che+17a; Rai+18]. In case of Honest-but-Curious (HbC) biocenters, I-GWAS is used in combination with DyPS (presented in Chapter 4) to enforce the safe release of results when facing colluding members. As in DyPS, I-GWAS assumes the threat of an external probabilistic polynomial-time adversary capable of observing released GWASes results, which it uses to mount recovery and membership attacks [Hom+08; San+09a; Pas+21; Zho+11]. Nevertheless, these releases might overlap, thus demanding new safe release conditions explained during this chapter.

Workflow overview. I-GWAS proceeds using the same pipeline as DyPS. To avoid repetition, Section 4.4 should be recalled. For the sake of completeness, the workflow is briefly summarized (and illustrated in Figure 5.1) in the following: (1) Before a biocenter starts interacting with the TEE, it remotely attests the authenticity of the hardware, software, and configuration of the TEE and finally establishes a secure connection with it. (2) Each biocenter locally encrypts and transfers data to the TEE, i.e., new genomes and their corresponding requests to add genomes to a study or to remove their participation from existing studies. (3) Upon reception of requests, the TEE decrypts the genomes, includes them in the data structures it maintains, and selects a batch of genomes (ideally including the newly added genomes) that can be safely used for a candidate release while impeding recovery attacks. Next, I-GWAS performs its extended LR-test analysis to evaluate the feasibility of membership attacks on the selected genome set merged with potentially overlapping releases. Only SNPs that would not allow such an

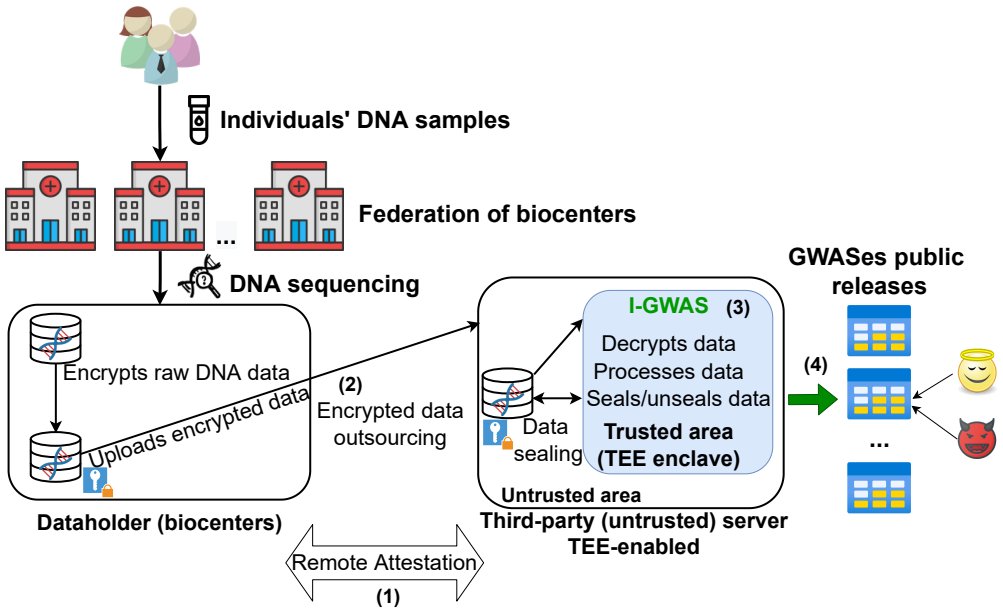


Figure 5.1: I-GWAS system and threat model.

attack will be considered in the study. (4) Finally, the TEE uses the data (genomes and SNPs) selected in step (3) to compute and release the actual GWAS statistics. The TEE periodically executes this workflow.

5.2 Safety conditions for interdependent GWASes

The overlapping region of SNPs and genomes from multiple GWASes can be leveraged by adversaries to reduce the solution space for inferring the matrices that verify an observed statistical result or update. This section reviews recovery attacks and extend the safety conditions proposed by Zhou et al. [Zho+11] and DyPS (Chapter 4) for interdependent GWASes.

Figure 5.2 illustrates two studies $\mathbf{GWAS}_1$ and $\mathbf{GWAS}_2$ that release statistics over L_1 SNPs and N_1 genomes, and over L_2 SNPs and N_2 genomes, respectively. The studies overlap in N_{ovl} genomes and L_{ovl} SNPs. Individually, both studies fulfill DyPS safety condition, that is, $|S_1|>|D_1|$ and $|S_2|>|D_2|$. However, leveraging knowledge about the overlapping regions of $\mathbf{GWAS}_1$ and $\mathbf{GWAS}_2$, adversaries might be able to reduce the search space for each possible situation in which these studies may overlap (i.e., evaluating addition, subtraction and union mappings over releases' solution spaces). Eventually, if the solution space of a combination of releases is not large enough (i.e., $|D| \approx |S|$ [Zho+11]) such a combination might be subject to a recovery attack.

Next, it is given a formal analysis of the complexity of the search space when adding, subtracting and taking the union of statistical results for single and pairwise allele frequencies, and also for test statistics when releases can be combined and leveraged by adversaries to circumvent privacy conditions in place

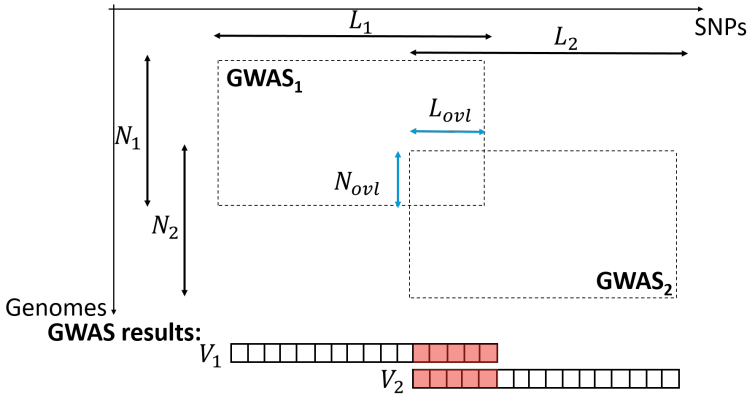


Figure 5.2: Illustration of two overlapping GWASes.

5.2.1 Recovery attack mappings

This section describes the statistics an adversary is able to compute from the results of two GWASes (for example, by observing two releases as in Figure 5.2). In particular, $\mathbf{GWAS}_1$ and $\mathbf{GWAS}_2$ release two maps m_1 and m_2 that associate SNPs to its allele frequency, counts or test statistics (V_1, V_2) in N_1 and N_2.

From the presented formulas, one can verify the formulas that are provided in the next sections for several search spaces an adversary has to infer, i.e., D_{add}, D_{sub}, D_{union}, S_{add}, S_{sub}, and S_{union} for allele frequencies and for test statistics.

Singlewise statistics mapping. From the m_1 and m_2 maps in Figure 5.2 an adversary can compute three other maps, m_{add}, m_{sub} and m_{union}, whose elements are defined over $L_1 \cup L_2$ as follows:

$$m_{add}[i] = \begin{cases} m_1[i] & \text{if } i \in L_1 \backslash L_2 \\ m_2[i] & \text{if } i \in L_2 \backslash L_1 \\ m_1[i] + m_2[i] & \text{if } i \in L_2 \cap L_1 \end{cases}$$

$$m_{sub}[i] = \begin{cases} m_1[i] & \text{if } i \in L_1 \backslash L_2 \\ -m_2[i] & \text{if } i \in L_2 \backslash L_1 \\ m_1[i] - m_2[i] & \text{if } i \in L_2 \cap L_1 \end{cases}$$

$$m_{union} = \{(m_1, m_2) \in \{0,1\}^{(L_1 \cdot N_1)} \cdot \{0,1\}^{(L_2 \cdot N_2)} \\ | \, m_1[i,j] = m_2[i,j] \text{ for } (i,j) \in (N_{ovl}, L_{ovl})\}$$

Pairwise statistics mapping. Following the same idea, from m_1 and m_2, one can compute other maps, m_{add}, m_{sub} and m_{union}, whose elements are defined for $(i,j) \in N_1$ or $(i,j) \in N_2$ as follows:

$$m_{add}[i,j,p,q] = \begin{cases} m_1[i,j,p,q] & \text{if } (i,j \in L_1 \backslash L_2) \vee (i \in L_1 \backslash L_2 \wedge j \in L_{ovl}) \\ m_2[i,j,p,q] & \text{if } (i,j \in L_2 \backslash L_1) \vee (i \in L_2 \backslash L_1 \wedge j \in L_{ovl}) \\ m_1[i,j,p,q] + m_2[i,j,p,q] & \text{if } i,j \in L_{ovl} \end{cases}$$

$$m_{sub}[i,j,p,q] = \begin{cases} m_1[i,j,p,q] & \text{if } (i,j \in L_1 \backslash L_2) \vee (i \in L_1 \backslash L_2 \wedge j \in L_{ovl}) \\ -m_2[i,j,p,q] & \text{if } (i,j \in L_2 \backslash L_1) \vee (i \in L_2 \backslash L_1 \wedge j \in L_{ovl}) \\ m_1[i,j,p,q] - m_2[i,j,p,q] & \text{if } i,j \in L_{ovl} \end{cases}$$

$$m_{union} = \{(m_1, m_2) \in \{0,1\}^{(L_1 \cdot N_1)} \cdot \{0,1\}^{(L_2 \cdot N2)} \\ | \, m_1[i,j,p,q] = m_2[i,j,p,q] \text{ for } (i,j,p,q) \in (N_{ovl}, L_{ovl})\}$$

5.2.2 Singlewise allele frequencies search space analysis

In this case, $\mathbf{GWAS}_1$ and $\mathbf{GWAS}_2$ release two maps m_1 and m_2 that associate a SNP to its minor allele counts (V_1, V_2) in N_1 and N_2. The space of possible solutions for addition is therefore $|S_{add}| = 2^{L_1 \cdot N_1 + L_2 \cdot N_2 - L_{ovl} \cdot N_{ovl}}$ and $|S_{sub}| = 2^{L_1 \cdot N_1 + L_2 \cdot N_2 - 2 \cdot L_{ovl} \cdot N_{ovl}}$ for subtraction. The latter formula comes from the fact that values in the intersection of the two matrices are canceled out under subtraction. Intuitively, $|S_{union}| = |S_{add}|$.

For the frequency spaces, we obtain $|D_{add}|$ and $|D_{sub}|$ by computing the product of the number of possible values for each SNP that they contain:
$|D_{add}| = (N_1{+}1)^{L_1 - L_{ovl}} \cdot (N_2{+}1)^{L_2 - L_{ovl}} \cdot (N_1{+}N_2{-}N_{ovl}{+}1)^{L_{ovl}}$.
$|D_{sub}| = (N_1{+}1)^{L_1 - L_{ovl}} \cdot (N_2{+}1)^{L_2 - L_{ovl}} \cdot (N_1{+}N_2{-}2N_{ovl}{+}1)^{L_{ovl}}$.

$|D_{union}|$ is the product of the frequency spaces of both releases divided by the frequency space over the overlapped area: $|D_{union}| = \frac{(N_1+1)^{L_1} \cdot (N_2+1)^{L_2}}{(N_{ovl}+1)^{L_{ovl}}}$.

5.2.3 Pairwise allele frequencies search space analysis

This analysis produces the pairwise statistics mappings that associate tuples (i, j, p, q) to pairwise allele counts in the respective datasets. Here, i, j denote the SNPs and p, q the allele types as in [Zho+11].

The solution space complexities for $|S_{add}|$ and $|S_{union}|$ are the same as for singlewise allele frequencies. In contrast, $|S_{sub}| = 2^{L_1 \cdot N_1 + L_2 \cdot N_2 - (L_{ovl} \cdot N_{ovl})}$, because m_{sub} depends on the values in the intersection (i.e., $(i, j) \in (N_{ovl}, L_{ovl})$), which are canceled out.

Like in the singlewise frequencies case, $|D_{add}|$ and $|D_{sub}|$ are obtained by computing the product of possible values of the SNPs over the released frequencies. Let $|D_{N_1}| = (N_1 + 1)^{\binom{L_1 - L_{ovl}}{2} + (L_1 - L_{ovl}) \cdot L_{ovl} + (L_1 - L_{ovl})}$ and $|D_{N_2}| = (N_2 + 1)^{\binom{L_2 - L_{ovl}}{2} + (L_2 - L_{ovl}) \cdot L_{ovl} + (L_2 - L_{ovl})}$. Then,

$|D_{add}| = |D_{N_1}| \cdot |D_{N_2}| \cdot (N_1 + N_2 - N_{ovl} + 1)^{L_{ovl} + \binom{L_{ovl}}{2}}$.

$|D_{sub}| = |D_{N_1}| \cdot |D_{N_2}| \cdot (N_1 + N_2 - 2N_{ovl} + 1)^{L_{ovl} + \binom{L_{ovl}}{2}}$.

$|D_{union}|$ is the product of the frequency spaces divided by the frequency space of the overlapped area: $|D_{union}| = \frac{(N_1+1)^{L_1 + \binom{L_1}{2}} \cdot (N_2+1)^{L_2 + \binom{L_2}{2}}}{(N_{ovl}+1)^{L_{ovl} + \binom{L_{ovl}}{2}}}$.

5.2.4 Test statistics search space analysis

From a GWAS release, apart from observing the sets of SNPs L and genomes N that participated in a study, adversaries can also observe the *p-values* statistics of the χ^2 test along with the r^2 values of linkage disequilibrium. However, the r^2 values encompass fewer information than pairwise frequency statistics from the adversary's perspective [Zho+11; Wan+09], which leads to the safety condition $|R^2| = \frac{(N+1)^{L + \binom{L}{2}}}{2^{\binom{L}{2}}}$ being smaller than $|D| = (N + 1)^{L + \binom{L}{2}}$.

We can therefore derive the solution space analysis for test statistics using the same approach as for pairwise frequency space analysis (which is actually the theoretical upper bound since other allele frequencies, such as singlewise frequencies, can be derived from them): $|S_{add}| = |S_{union}| = 2^{L_1 \cdot N_1 + L_2 \cdot N_2 - L_{ovl} \cdot N_{ovl}}$ and $|S_{sub}| = 2^{L_1 \cdot N_1 + L_2 \cdot N_2 - (L_{ovl} \cdot N_{ovl})}$.

Test statistics space complexities are derived from the product of possible values for the SNPs over the released test statistics with $|D_{N_1}|$, $|D_{N_2}|$ as above: $|R^2_{add}| = \frac{|D_{N_1}|}{2^{\binom{L_1 - L_{ovl}}{2} + (L_1 - L_{ovl}) \cdot L_{ovl}}} \cdot \frac{|D_{N_2}|}{2^{\binom{L_2 - L_{ovl}}{2} + (L_2 - L_{ovl}) \cdot L_{ovl}}} \cdot \frac{(N_1 + N_2 - N_{ovl} + 1)^{L_{ovl} + \binom{L_{ovl}}{2}}}{2^{\binom{L_{ovl}}{2}}}$.

$|R^2_{sub}| = \frac{|D_{N_1}|}{2^{\binom{L_1 - L_{ovl}}{2} + (L_1 - L_{ovl}) \cdot L_{ovl}}} \cdot \frac{|D_{N_2}|}{2^{\binom{L_2 - L_{ovl}}{2} + (L_2 - L_{ovl}) \cdot L_{ovl}}} \cdot \frac{(N_1 + N_2 - 2N_{ovl} + 1)^{L_{ovl} + \binom{L_{ovl}}{2}}}{2^{\binom{L_{ovl}}{2}}}$.

$|R^2_{union}|$ is computed as the product of the test statistics spaces of both releases

over the overlapping area: $|R^2_{union}| = \frac{\frac{(N_1+1)^{L_1+\binom{L_1}{2}}}{2^{\binom{L_1}{2}}} \cdot \frac{(N_2+1)^{L_2+\binom{L_2}{2}}}{2^{\binom{L_2}{2}}}}{\frac{(N_{ovl}+1)^{L_{ovl}+\binom{L_{ovl}}{2}}}{2^{\binom{L_{ovl}}{2}}}}$.

In addition, although estimating correct values of r^2 (to measure associations between SNPs that can be used to facilitate attacks) from GWAS test statistics is NP-hard [Zho+11], **I-GWAS** not only conducts the searching space analysis (assuming the full SNP-set L) but also certifies that SNPs found to be in linkage disequilibrium do not have their statistics released (during the LR-test phase). Such a more conservative approach impedes adversaries from leveraging LD to mount attacks.

5.2.5 Protecting interdependent GWASes against recovery attacks

As analyzed before, releases of interdependent GWASes need to satisfy new safety bounds that can lead to unsafe situations if ignored. In particular, these new bounds enforces that the conditions of Equation 4.1 also holds for overlapping releases so that the solution space of combinations of releases is assured to be sufficiently large.

Next sections detail how **I-GWAS** protects the release of statistics against recovery attacks using sequential releases, which assumes that studies are dynamically updated as new genomes are sequenced and/or removed. It starts by providing the intuition behind the release of GWAS statistics using an example where only one interdependent GWAS has been previously released, before generalizing to G GWASes.

5.2.6 Sequential releases of GWASes

Let us assume that a first GWAS — $\mathbf{GWAS}_1$ —has released statistics over L_1 SNPs with N_1 genomes, and that a second, $\mathbf{GWAS}_2$, aims at releasing statistics over L_2 SNPs. Furthermore, let us recall $LtoN_{single}(L)$ function from **DyPS**, which represents the minimum number of genomes one should combine to release the results of an independent GWAS on L SNPs. Our approach consists in increasing the number N_2 of genomes that $\mathbf{GWAS}_2$ uses so that the new safety bounds for interdependent releases are verified. Particularly, it is discovered that releasing $\mathbf{GWAS}_2$ with $LtoN_{single}(L_2)$ genomes is not safe when overlapping genome data is present.

Figure 5.3 shows the smallest safe value for N_2 when L_1=1,000 and N_1=$LtoN_{single}(L_1)$, and when $L_{ovl} \in [0, L_1]$ and $N_{ovl} \in \{\frac{N_1}{2}, N_1\}$. In this figure, the default line represents $LtoN_{single}(L_2)$, i.e., the formula that assumes individual releases. By observ-

ing the behavior of N_2, one can notice that depending on the attack (i.e., targeting the addition, subtraction, or union), N_2 exceeds $LtoN_{single}(L_2)$. By observing the behavior of N_2, one can notice that depending on the attack (i.e., targeting the addition, subtraction, or union), N_2 exceeds $LtoN_{single}(L_2)$. In such situations, it is identified that interdependent releases are not safe if relying on $LtoN_{single}(L_2)$. In addition, it can be noticed that protecting against the union and addition attack always requires more genomes than for the other cases.

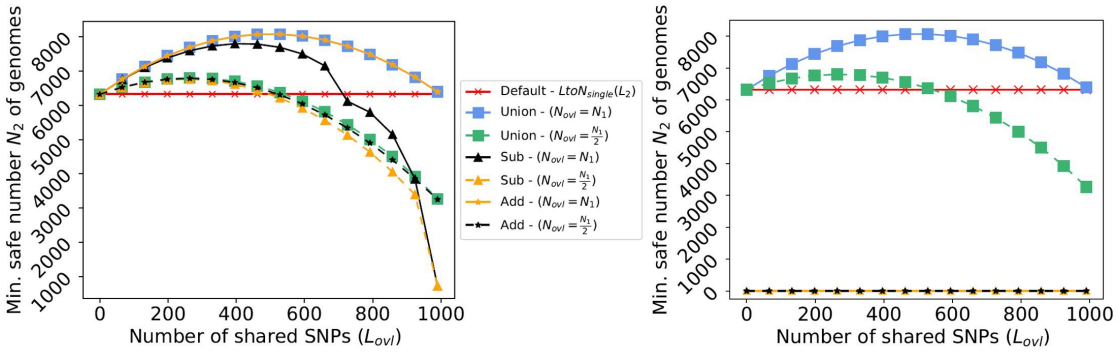
On the other hand, it can be noticed that the addition and subtraction attack lines against r^2 releases (Fig. 5.3 (b)) always stay below the default line. Note that the lines plotted for the addition and subtraction mappings overlap each other at the bottom of the chart). This means that adding or subtracting r^2 values to launch a privacy attack is not practical and matches previous works findings [Zho+11; Wan+09].

Another finding is that the addition and subtraction attack lines against r^2 releases (Fig. 5.3 (b)) always stay below the default line (note that the lines plotted for the addition and subtraction mappings overlap each other at the bottom of the chart). This means that adding or subtracting r^2 values to launch a privacy attack is not practical and matches previous works findings [Zho+11; Wan+09].

Interestingly, one can also notice that N_2 decreases when L_{ovl} increases for all type of attacks. This downwards behavior comes from the fact that the solution spaces (e.g., $|S_{add}|$) grow faster than the frequency spaces (e.g., $|D_{add}|$) with L_{ovl}, and because a recovery attack is deemed possible depending on their ratio (e.g., $|S_{add}|/|D_{add}|$) [Zho+11].

In summary, interdependent releases are safe when the space analysis for each type of attack is kept within safe boundaries, i.e., the combined solution space between the releases is sufficiently larger than their combined frequency spaces. In particular, the number N_2 of genomes required to protect interdependent releases depends on L_{ovl} and N_{ovl}. For instance, for these experiments, if such conditions are not enforced, up 28.6% of the genomes are vulnerable to recovery attack, i.e., their genotype sequence might be inferred if overlapping data is not properly considered by the privacy-protection mechanism.

Therefore, to mitigate recovery attacks, **I-GWAS** identifies N_2 such that $|S_{op}|>|D_{op}|$ for $op \in \{add, sub, union\}$ given the overlaps a study has with previous GWAS. For pairwise allele frequencies and for test statistics, the union attack provides the theoretical bound for interdependent releases. Hence, it is sufficient to check that N_2 verifies $|S_{union}|>|D_{union}|$. For singlewise allele frequencies (not illustrated in Fig. 5.3 for space reasons), the subtraction attack defines the safety bound. In summary, **I-GWAS** selects the largest bound as the safety threshold and apply its conditions when selecting safe batches of genomes for the creation of safe releases.



(a) Pairwise releases

(b) *R*-square releases

Figure 5.3: Smallest number of genomes N_2 that a GWAS that overlaps with a previous GWAS should use for a safe release depending on their overlapping SNP-set size (L_{ovl}) and genomes set size (N_{ovl}).

5.2.7 Scaling with the number of GWASes

This section shows how **I-GWAS** method extends to the case where G GWASes have previously released statistics. Let us note $|S_i|$ the solution space for a given $\textbf{GWAS}_i$, and $|D_i|$ its frequency set space. It is presented only the analysis for the pairwise frequency space. The analysis for the other statistics (i.e., singlewise and r^2) is similar.

Before introducing **I-GWAS**' solution for multiple releases of interdependent GWASes, let us first discuss an intuitive solution to this problem. Indeed, to enforce that several interdependent releases are safe, one could compute all possible combinations of existing releases. Then, measure and evaluate if the solution space is sufficiently large given the released statistics space. Such a brute force approach is secure but has exponential complexity and is not reasonable in practice [Pas+21]. Motivated by that, **I-GWAS** offers a novel solution with linear complexity. **I-GWAS** relies on the following Theorem that defines how and when multiple interdependent GWASes can be dynamically released without infringing on the genomic privacy of their participants.

Theorem 5 (Safe releases of interdependent GWASes)**.** *For every $GWAS_i$ and $GWAS_j$, if (1) $|S_i|>|D_i|$ and (2) $\frac{|S_j|}{|D_j|} > \prod_{i \neq j} |S_i \cap S_j|$, then any combination of releases from $GWAS_i$ and $GWAS_j$ involves enough genomes to prevent recovery attacks.*

Proof. The very first release of a GWAS in the federation does not overlap with any other GWAS, and meets the condition for independent GWAS releases, i.e., $|S_i|>|D_i|$. The release is safe.

Let us assume that any combination of $i \geq 1$ GWAS releases is safe. Let further j be the ID of the $(i+1)$-th GWAS, $\{S_1, S_2, ..., S_i\}$ the sets of the i previous GWAS solution spaces, and $\{D_1, D_2, ..., D_i\}$ their corresponding frequency spaces. All these GWASes also contain enough genomes to meet the single-GWAS safe condition (i.e., $|S_j|>|D_j|$). Then the inclusion-exclusion formula states that

$$|\bigcup_{j=1}^{i} E_j| = \frac{\prod_{j=1}^{g} |E_j|}{\frac{\prod_{1\leq j<k\leq g} |E_j \cap E_k|}{\frac{\prod_{1\leq j<k<l\leq g} |E_j \cap E_k \cap E_l|}{\cdots \over (-1)^{g-1}|E_j \cap \cdots \cap E_g|}}}$$

where all $|E_j|$ can be substituted by either $|S_j|$ or $|D_j|$ to compute the sizes of the combination solution and frequency spaces, respectively. Given this formula, one easily obtains that $|\bigcup_{j=1}^{i} S_j| \geq \frac{\prod_{j=1}^{g} |S_j|}{\prod_{1\leq j<k\leq g} |S_j \cap S_k|}$ and that $1 \leq |\bigcup_{j=1}^{i} D_j| \leq \prod_{j=1}^{g} |D_j|$. Therefore, if one ensures that $\frac{\frac{\prod_{j=1}^{g} |S_j|}{\prod_{1\leq j<k\leq g} |S_j \cap S_k|}}{\prod_{j=1}^{g} |D_j|} \geq 1$ then $|\bigcup_{j=1}^{i} S_j| > |\bigcup_{j=1}^{i} D_j|$. This condition is equivalent to $\prod_{j\leq g-1} \frac{|S_g|}{|D_g|} \cdot \left(\frac{|S_j|}{|D_j|} \cdot \frac{1}{\prod_{k\leq g} |S_g \cap S_k|}\right) \geq 1$, which is provided since $\prod_{j\leq g-1} \frac{|S_g|}{|D_g|} \geq 1$ because of condition (1), and since $\frac{|S_j|}{|D_j|} \cdot \frac{1}{\prod_{k\leq g} |S_g \cap S_k|} \geq 1$ because of condition (2). □

I-GWAS relies on Theorem 5 to verify that a GWAS can release or update its results, given that other GWASes have already released theirs. This verification has a complexity that is linear with the number of GWASes. Moreover, this analysis can also be applied to other GWAS statistics (i.e., singlewise and r^2), once the relation $|S|>|D|$ for the corresponding type of statistics is also kept. To illustrate this process, **I-GWAS**' algorithms (pseudocode) are discussed in the following:

I-GWAS' pseudocode for selecting a safe batch of genome requests. Algorithm 3 illustrates the operations that **I-GWAS** performs in order to select a safe batch of genome operations considering overlapping GWASes. Every time a safe batch of requests for single-GWAS is found, its requests are simulated (line 5), and then the solution spaces are checked over all existing releases, if the release has safe boundaries, it can proceed (line 6 to 10).

5.2.8 Allowing safe genome removals

I-GWAS provides dynamic processing of genomes and their safe removal, similarly to **DyPS**, and extends it to interdependent GWASes. For the update of a given (single) GWAS, DyPS assembles a batch of genome additions and removals, respectively represented by A and R, such that $|A| + |R| \geq LtoN_{single}(L)$ and $|A| \geq |R|$.

ALGORITHM 3 Verification of a set of genome requests to prevent recovery attacks.

procedure $isSafe_{\text{I-GWAS}}(g, G, Add\_Req, Rmv\_Req)$
  Input: g: candidate GWAS with Add_Req and Rmv_Req genome additions and removals, respectively; G: set of released GWASes
  Output: set of selected genome addition and removal requests for interdependent GWASes ($LtoN_{\text{I-GWAS}}$)
  Uses: S_i and D_i respectively return the solution and frequency space sizes for a GWAS i (or an intersection of GWASes); $copyAndApplyRequests(g, Add\_Req, Rmv\_Req)$ applies a batch of request to a copy of a GWAS and returns it.
  $g' = copyAndApplyRequests(g, Add\_Req, Rmv\_Req)$
  $tmp = \frac{S(g')}{D(g')}$
  for i in G **do**
    $tmp = tmp.\frac{S_i}{D_i}.\frac{1}{S(g' \cap i)}$
  end for
  return $tmp \geq 1$
end procedure

The rationale behind this approach is that even if some genomes that participated in a GWAS are removed, the solution space an adversary has to explore only increases with time. **I-GWAS**, however, offers a method that acknowledges interdependent GWASes. Before a GWAS can release its results, **I-GWAS** determines the minimum number of genomes it should use, which might also depend on other GWASes and which is denoted as $LtoN_{\text{I-GWAS}}$, to satisfy the conditions of Theorem 5.

Let us consider an example with two (possibly overlapping) releases of two interdependent GWASes. The first GWAS, $\textbf{GWAS}_i$, consists of genome additions A_i and removals R_i, and the second GWAS,$\textbf{GWAS}_j$, consists of genome additions A_j and removals R_j. Since these releases were orchestrated by **I-GWAS**, we have $|A_i| + |R_i| \geq LtoN_{\text{I-GWAS}}(i)$ and $|A_j| + |R_j| \geq LtoN_{\text{I-GWAS}}(j)$. Let us then assume that $\textbf{GWAS}_i$ is updated with new genome operations $A_{i'}$ and $R_{i'}$, such that $A_{i'} > R_{i'}$. Then, each GWAS considered alone stays safe. The number of remaining genomes in $\textbf{GWAS}_i$ is $|R_i| + |A_i \setminus R_{i'}| + |R_{i'} \setminus A_i| + |A_{i'}| > |R_i| + |A_i| > LtoN_{\text{I-GWAS}}(i) \geq LtoN_{single}(i)$. The combination of $\textbf{GWAS}_i$ and $\textbf{GWAS}_j$ is also safe because Theorem 5 enforces that it contains more than $LtoN_{\text{I-GWAS}}(i)$ genome operations.

5.3 Membership attacks on interdependent GWASes

As with novel recovery attacks presented in Section 5.2, an adversary can leverage the fact that some genomes might have been used in multiple interdependent studies for succeeding in membership attacks. In particular, an adversary can launch membership attacks over release combinations, which might increase the identification power of the attack.

DyPS combines SG's LR-test with an exhaustive verification process to dynamically update the statistics of a single GWAS. In particular, every SNP in a candidate release is checked against existing releases to identify if it statistics has been released before by another study. Compared to **I-GWAS**, DyPS cannot cope with the presence of interdependent studies. In practice, it is identified that an adversary could combine statistics across several overlapping releases and mount a membership attack (as shown by the experiments in Section 5.5). Therefore, **I-GWAS** encompasses additional required verifications to support dynamic releases of interdependent GWASes.

Like previous works [Hal+21; San+09b; Pas+21] **I-GWAS** also leverages SG's LR-test for membership protection. However, in contrast to existing solutions, which evaluate the conditions of safe releases considering studies separately, **I-GWAS** offers a novel pipeline able to protect the privacy of participating genomes by implementing an exhaustive verification step that acknowledges all possible sets of genome and SNPs combinations among existing studies. In particular, **I-GWAS** applies an exhaustive local verification (on a single-GWAS level) combined with a global verification that considers all existing combinations of GWASes on a per-SNP basis.

5.4 Protecting interdependent GWASes against membership attacks

After selecting a safe batch of genomes using the conditions to prevent recovery attacks on interdependent GWASes presented in Section 5.2.5, **I-GWAS** identifies data that can be released without posing membership privacy risks. **I-GWAS** leverages SG's to run membership inference tests over the selected genomes of the candidate GWAS. Thus, identifying the set of safe SNP positions regarding a candidate release selected data.

However, the above verification is not enough. Indeed, **I-GWAS** also needs to enforce that the selected SNPs can be safely released or updated within a single GWAS previous releases (i.e., verifying which SNPs are "locally" safe). This task consists in executing additional LR-test verifications over all possible combinations of releases (and so genome distributions) within previous releases of a particular study (depicted on the left side of Figure 5.4). Recall that rare alleles and SNPs in LD are primarily blocked from participation, and therefore such information could not be leveraged by adversaries.

In addition, to prevent an adversary from leveraging the combination of released statistics from other studies, **I-GWAS** retains in the candidate GWAS only those SNPs of the previous step that remain "globally" safe. For this, **I-GWAS**

first identifies overlapping SNPs and then executes additional LR-tests over the combination of genomes from heterogeneous studies that shared SNPs with the candidate GWAS (depicted on the dashed lines coming from the right side of Figure 5.4). After this procedure, **I-GWAS** has identified a list of SNPs that survived all verifications and therefore can be used for a safe release.

Let us consider the example illustrated in Figure 5.4, where $\mathbf{GWAS}_1$ releases take place first, and SNPs are selected following the local verification only (represented by solid lines) because there was only one study. The notation N_{i_r} represents the genome set selected for $\mathbf{GWAS}_i$, release r. Let us discuss the different scenarios **I-GWAS** considers when a candidate release is found for $\mathbf{GWAS}_2$:

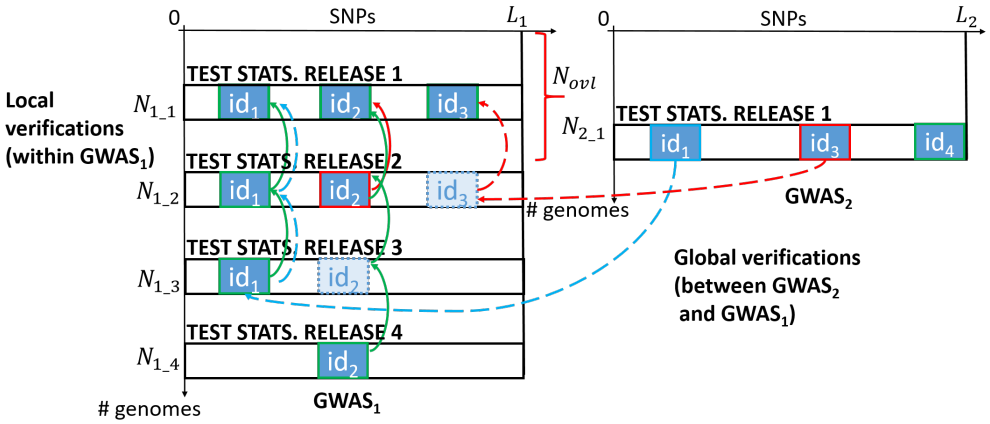


Figure 5.4: Exhaustive verification process to protect interdependent GWASes releases against membership attacks.

Local verifications. A first situation occurs when a SNP that is labeled as safe by **I-GWAS** and has never been studied before can be released without global verification because an adversary cannot combine releases. This is the case for all selected SNPs in N_{1_1} of $\mathbf{GWAS}_1$ and id_4 in N_{2_1} of $\mathbf{GWAS}_2$. When SNPs have been considered in previous releases of a study, e.g., id_1, id_2 and id_3 of $\mathbf{GWAS}_1$ at N_{1_2} and N_{1_3}, those releases are combined by **I-GWAS** and extra LR-tests are conducted over them to certify that candidate SNPs are also identified as safe when the combinations are tested. In this example, id_1 succeed in all local verifications of $\mathbf{GWAS}_1$ releases (green lines). In contrast, id_2 failed (red lines) when tested against the (N_{1_2}, N_{1_1}) combination. Thus, id_2 is not considered in N_{1_2} release.

Local and global verifications. In this case, a SNP is found to be safe for a single GWAS release, but has been released in another study before. For example, id_1 is safe over the N_{2_1} genome set of $\mathbf{GWAS}_2$, but id_1 statistics has been released in $\mathbf{GWAS}_1$ before (over the overlapping genome set N_{ovl}). In this case, **I-GWAS** evaluates new LR-test rounds to certify that a SNP is also identified as safe over

the combinations of genomes among the two studies. This is represented by the dashed line coming from the selected SNPs of $\mathbf{GWAS}_2$. In our example, id_3 is detected as safe over the N_{2_1} set in $\mathbf{GWAS}_2$, but when combined with the other genome sets from $\mathbf{GWAS}_1$, the LR-test identified that this SNP cannot have its statistics safely released anymore (dashed red lines). On the other hand, id_1 passed all tests when $\mathbf{GWAS}_1$ releases are combined with those of $\mathbf{GWAS}_2$ (dashed blue lines), and therefore might have its statistics published. SNP id_4 is not tested since it has never been released before by any study.

Furthermore, to avoid that previously released SNPs (and potentially not considered in the future) being leveraged by adversaries, **I-GWAS** keeps track of all released SNPs so that when combining releases for further verification, those SNPs are also checked. These "ghost" SNPs are represented by dashed boxes (id_2 and id_3) in Figure 5.4, which allow **I-GWAS** to apply the exhaustive verification step for all released SNPs regardless if they can currently be observed or not. For instance, even though SNP id_2 was not released in N_{1_2} of $\mathbf{GWAS}_1$, it is checked anyway once it is a candidate SNP selected for N_{1_4} release. In that example, id_2 has been labeled as safe over all runs of membership verification, and therefore is allowed to have its statistics released at N_{1_4} of $\mathbf{GWAS}_1$.

In summary, **I-GWAS**'s exhaustive verification certifies that only SNPs selected as safe after multiple LR-test runs over the combinations of overlapping releases are allowed to release statistics. Thus, certifying that the identification power of individuals is kept within safe boundaries even in cases where an adversary is mounting membership attacks leveraging combination of releases.

After selecting a safe batch of genomes (that would allow a safe release against recovery attacks), **I-GWAS** evaluates which SNPs might have their GWAS statistics safely released without allowing membership inference. Algorithm 4 details such a process. First, **I-GWAS** identifies which SNPs can be released over the bath of selected genomes for the single-GWAS candidate release (line 5) using the $SNPSelection$ function that represents the SecureGenome's [San+09a] step, which is used to find SNPs that can be safely exposed while avoiding membership attacks. Nevertheless, the standard $SNPSelection$ function (i.e., SecureGenome) only works in a static GWAS release setting. To enable dynamic GWASes releases under the presence of overlapping data, **I-GWAS** conducts additional verifications explained in Section 5.4 and described in the following paragraph.

I-GWAS enforces that each $\mathbf{SNP}_l$ (from the original SNP-set L of a study g) that is identified as safe by the $SNPSelection$ function is to be tested over all possible combinations of existing releases (lines 6 to 24). In particular, **I-GWAS** first identifies and collects all releases where $\mathbf{SNP}_l$ has previously participated (lines 7 to 13). Then, **I-GWAS** loops and run the $SNPSelection$ function for each possible combination of intersected releases that used $\mathbf{SNP}_l$ (lines 14 to 16). If $\mathbf{SNP}_l$

is labeled as safe in all verifications (i.e., over all combinations), it means that it can be safely released. Otherwise, SNP_l is withhold from the candidate release (line 20). In the end of this loop, **I-GWAS** has identified a list of SNPs that survived (i.e., was labeled as safe when checked against existing combinations of releases) and therefore can have their statistics safely released (line 25). Additionally, recall that in this step, SNPs that presents rare allele frequencies are in LD are also identified and blocked from participation.

Algorithm 4 Selection of SNPs to prevent membership attacks.

```
1:  procedure checkInterdependentMembership(Add_Reqs, Rmv_Reqs, G)
2:      Input:Add_Req and Rmv_Reqs of candidate study g and G set of released GWASes
3:      Output: set of selected SNPs for safe interdependent GWASes
4:      Uses: AllCombinations(relsToCombine) creates combinations of releases in relsToCombine;
    SNPSelection(Add_Reqs, Rmv_Reqs, g) runs the LR-test and returns the safe SNP-set for a single GWAS
    g
5:      selected_SNPs := SNPSelection(Add_Reqs, Rmv_Reqs, g)
6:      for SNP_l in selected_SNPs do // for each selected safe SNP in a single GWAS, isSafeSingleBiocList
7:          relsToCombine := ∅
8:          for g in G do //check each existing GWAS g
9:              for rel in g do //check each release of GWAS g
10:                 if (SNP_l == rel.s) then // SNP position SNP_l has been released in a release rel
11:                     relsToCombine.add(rel)
12:                 end if
13:             end for
14:             for combRel in AllCombinations(relsToCombine) do
15:                 testSet := combRel.Add_Reqs + combRel.Rmv_Reqs + Add_Reqs + Rmv_Reqs // merge
    genomes requests
16:                 checkSafeSNPs := SNPSelection(testSet)
17:                 if (SNP_l in checkSafeSNPs) then
18:                     continue // this SNP can be released
19:                 else
20:                     safe_SNPs.del(SNP_l) // this SNP cannot be released
21:                 end if
22:             end for
23:         end for
24:     end for
25:     return selected_SNPs // set of safe SNPs for candidate release
26: end procedure
```

Algorithm 5 details the full pipeline of **I-GWAS**' framework for privacy-preserving releases of interdependent GWASes. From lines 6 to 13, **I-GWAS** check if there exists a safe batch of requests for a single-GWAS. If this is the case, those requests are checked now considering interdependent GWASes (line 14, which is the combination of Algorithms 3 and 4). If this evaluation succeeds, aggregate GWAS statistics can be computed over the selected genomes and SNPs and are publicly published. Only the SNPs identified as safe by Algorithm 4 will have both aggregate and test statistics released. On the other hand, the others (unsafe) SNPs are secluded from having their GWAS statistics released. Note that only the **I-GWAS**'s TEE enclave has access to the identified unsafe SNP ids and their respective GWAS statistics, and therefore these SNPs cannot be leveraged by adversaries to mount genomic

Algorithm 5 Full I-GWAS workflow.

```
1: procedure I-GWAS workflow for interdependent GWASes(G)
2:     Input: set G of GWASes
3:     Output: updated statistics of a safe GWAS g
4:     Uses: NtoL_single(g) returns the list of selected biocenters and their corresponding batch of requests to
       update a single-GWAS g; and NtoL_I-GWAS(g, G, Add_Req, Rmv_Req) returns the list of selected biocenters
       and their corresponding batch of requests to update GWAS g
5:     isSafeSingleBiocList := ∅
6:     isSafeFinal := False
7:     selected_SNPs := ∅
8:     for g in G do //check each existing GWAS g
9:         isSafeSingleBiocList := LtoN_single(g)
10:        if (isSafeSingleBiocList ≠ ∅) then // assemble the requests from selected biocenters
11:            for b in isSafeSingleBiocList do
12:                Add_Reqs := isSafeSingleBiocList.addRequests
13:                Rmv_Reqs := isSafeSingleBiocList.rmvRequests
14:            end for
15:        end if
16:        isSafeFinal = isSafe_I-GWAS(g, G, Add_Reqs, Rmv_Reqs)
17:        if (isSafeFinal) then // update the requests from selected biocenters
18:            selected_SNPs := checkInterdependentMembership(Add_Reqs, Rmv_Reqs, G)
19:        end if
20:    end for
21:    computeTestStats(Add_Reqs, Rmv_Reqs) // update test statistics over selected requests
22:    computeAggregateStats(selected_SNPs) // update aggregate statistics over selected SNPs
23: end procedure
```

privacy attacks.

The complexity of **I-GWAS**' verification increases with the number of releases and studies. The computational complexity for the verification is $O(L' \cdot 2^{LocalRel \cdot OverlappedRel})$, where L' is the number of selected SNPs after the first run of the LR-test on the candidate release set, and $LocalRel$ and $OverlappedRel$ are the number of releases within a study and the number of overlapping releases from other GWASes, respectively. Furthermore, as GWASes often aim at determining only the K most highly ranked SNPs [Bar+12; Che+16b], **I-GWAS** could limit the value of L' to be faster.

5.5 Experimental evaluation

I-GWAS was evaluated under the same settings as of **DyPS**, i.e., **I-GWAS** is implemented in C++ and runs inside a Intel SGX enclave using Graphene SGX [TPV17]. For **I-GWAS**, it was also used real genomes from the phs001039.v1.p1 dbGAP dataset [Wal+11], which consists of 14,860 case and 13,035 control genomes to better measure the effect of interdependency between genomes belonging to different studies. Furthermore, it was adopted the same standard settings of SecureGenome for the LR-tests to decide which SNPs might have their allele frequencies safely released: 0.1 false-positive rate, 0.05 MAF cut-off, 10^{-5} LD cut-off and a 0.9 true-positive rate for the identification threshold.

I-GWAS performance is evaluated along 3 metrics: *privacy*, *data utility* and *running time*. **I-GWAS** is compared against **DyPS** so that the implications of releasing GWASes without being concerned about overlapping genomic data are clearly showed. Besides, **I-GWAS** is compared with ϵ-DP using Laplace mechanism for the releases of GWAS statistics. It is considered several ϵ and privacy budgets (*pvb*) spent over releases to evaluate trade-offs. *pvb* is used to keep DP properties over multiple releases. *pvb* starts at 1 (100% of ϵ) and each release consumes a fraction of ϵ. When *pvb* is exhausted, DP cannot release data with original privacy guarantees. It was utilized PyDP, a Python wrapper for Google's Differential Privacy C++ library [Goo22] to create differentially private releases.

A comparison between **I-GWAS** with a method based on local DP is not presented since it introduces higher perturbation than a centralized DP scheme [Cor+18; LS17]. Moreover, the experiments showed that the system-side performance (e.g., bandwidth, memory, and CPU consumption) of **I-GWAS** is very similar to that of other TEE-based solutions [Sad+18; Che+16b; BAZ20; CT18; Pas+21; Koc+19]. Additionally, **I-GWAS** only imposes a penalty of requiring extra genomes to protect overlapping releases (cf. Table 5.2) when compared to DP-based releases. Such a limitation depends on the assumed workload, e.g., the rate at which new genomes are added and the frequency of overlapping data (cf. Figure 5.3).

To be fair when comparing **I-GWAS** against DP-based releases, it was created a release utility metric that acknowledges both data coverage (amount of data allowed to be used in a release) and accuracy loss. Otherwise, DP utility score would always perform worse than **I-GWAS**. The utility of a release is evaluated as follows:

$$\sum_{l=0}^{L} \frac{\mathbf{SNP}_{lrel} \cdot \mathbf{SNP}_{lacc}}{L} \tag{5.1}$$

In this formula, L is the original SNP-set of the GWAS, $\mathbf{SNP}_{lrel} \in \{0, 1\}$, i.e., 1 if statistics over SNP l has been released and 0, otherwise, and $\mathbf{SNP}_{lacc}$ is the accuracy of the released statistics of SNP l compared to its original (unperturbed – noise-free) result. Note that we still evaluate **I-GWAS** along other traditional metrics mentioned before.

5.5.1 Privacy and data utility

Figure 5.5 illustrates the privacy at the cost of data utility when having to prevent the release of GWAS results of some SNP positions. This experiment used all 14,860 case genomes and consider two GWASes over 10,000 SNPs and varied the fraction of overlapping genomes among studies between 1% to 50%. The vulnerable SNPs are positions that would put participating genomes at risk of being identified in a membership attack. These SNPs need to be identified and secluded from public

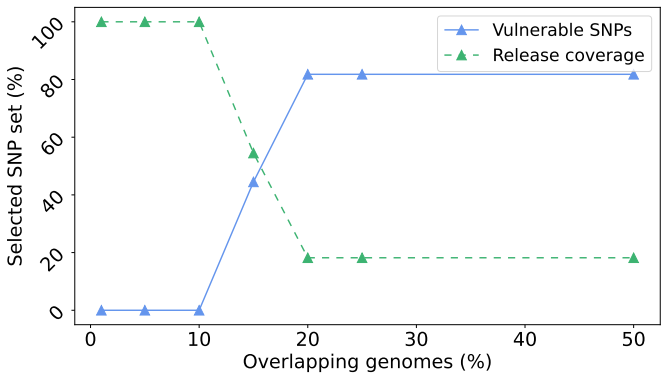


Figure 5.5: Vulnerable SNPs and release coverage when protecting interdependent GWASes against membership attacks.

releases. The GWAS results are only known by the **I-GWAS**'s trusted enclave, and only the results of SNPs that can be safely exposed are publicly shared.

It can be noticed that over the dbGAP dataset considered, when the number of overlapping genomes increases, the release coverage decreases because more SNPs at risk are found and withheld by **I-GWAS**. In the worst case, 81.8% of SNPs might be used to identify individuals and therefore are prohibited from having their statistics released. Interestingly, in this use case, it can be noticed that as long as only very few genomes are shared between studies (between 1% – 10%), all SNPs could be released. It is worth mentioning that **I-GWAS** is able to identify vulnerable SNPs independently of the dataset.

Table 5.1: I-GWAS protection against membership attacks with four GWASes.

# SNPs	**Vulnerable SNPs using DyPS (%)**	**Release coverage w/ I-GWAS (%)**
1,000	80	20
2,500	81.8	18.2
5,000	84.6	15.4
10,000	92.3	7.7

Table 5.1 presents the results of our second scenario where 4 GWASes are considered. The first 3 GWASes used disjoint sets of 4,953 genomes each, while the last one shares each of its 14,860 genomes with the other three GWASes. Each experiment/line assumes a different number of SNPs. Using **DyPS**, which cannot protect releases of interdependent studies, the number of SNP positions at risk increases with the overall number of SNPs, from 80% to 92.3%, which is aligned with the findings of Simmons et al. [SBS19]. As a result, **I-GWAS** presents a smaller release coverage since it refrains statistics of vulnerable SNPs from being

released. In this experiment, **I-GWAS** released statistics over 20% to 7.7% of the original SNP-set according to the scenario.

5.5.2 Comparison to Differential Privacy

The next experiments considered a scenario where a GWAS ($\mathbf{GWAS}_1$) has already released single allele frequencies over 1,000 SNPs using 7,430 genomes (note that $LtoN(1,000) = 6,320$). Now, a second study ($\mathbf{GWAS}_2$) aims at releasing single allele frequencies over 1,000 SNPs using 14,860 genomes from which 7,430 was used in ($\mathbf{GWAS}_1$) and share half of the SNPs. Therefore, $N_{ovl} = 7,430$ and $L_{ovl} = 500$. The accuracy of the single allele frequencies released by $\mathbf{GWAS}_2$ using **I-GWAS** or DP are compared. Setting a privacy budget with DP is necessary to support a given number of releases. The privacy budget interferes on the noise applied to protect a release. This experiment was repeated 100 times and report the average results. Table 5.2 presents the results of this experiment.

I-GWAS detects that compared to the state-of-the-art *LtoN* formula, which indicates 6,320 genomes would be enough to protect $\mathbf{GWAS}_2$, the second study would need at least 350 additional genomes (i.e., 5.53% more genomes) in order to prevent recovery attacks on overlapping studies. Relying on additional genomes to enforce privacy slightly delays releases. Future work could be to use synthetic genomes while preserving statistical properties for this purpose [Hua+15; Rai+17b].

ϵ-DP releases statistics over all SNPs and presents a better release utility score in several scenarios (compare blue and green scores in Table 5.2), but it perturbs the results (accuracy loss column). The accuracy loss metric corresponds to how distant the DP result is from the original statistics one would obtain without privacy guarantees. While **I-GWAS** release utility score is equal to 66 irrespective of the number of subsequent releases, DP release utility scores varied from 76.41 to 94.11 in the best cases. Nevertheless, DP showed poor performance in settings that would allow more future releases (red scores in Table 5.2). In fact, the limited privacy budget of DP-based releases restricts the number of conceivable releases, whereas **I-GWAS** can afford an unlimited number of releases by virtue of its exhaustive verification methods. To allow more releases, ϵ-DP should use smaller *pvb* over releases so that some privacy budget is kept to protect future releases. Nevertheless, smaller *pvb* means increased accuracy loss. Intuitively, using larger values of *pvb* over releases would increase the release utility of ϵ-DP releases but reduces the number of allowed safe releases. Therefore, when adopting DP, GWAS federations have to carefully select the privacy budget that will be spent over dynamic releases. For instance, considering $pvb = 0.12$ per release, ϵ-DP could afford only up to 8 safe releases with decreased utility. In contrast, using half of the total privacy budget per release ($pvb = 0.50$) would only allow two safe releases but higher utility.

Table 5.2: Comparison between I-GWAS and the standard ϵ-DP using Laplace mechanism under several settings. The results represent the average of 100 repetitions.

Approach	**Maximum # releases**	**Accuracy loss (%)**	**Coverage (%SNPs)**	**Required additional genomes (%)**	**Release utility score**
ϵ=0.5 (pvb=0.12)	8	290.98	100	0	1.86
ϵ=0.5 (pvb=0.25)	4	116.46	100	0	52.77
ϵ=0.5 (pvb=0.33)	3	85.09	100	0	63.85
ϵ=0.5 (pvb=0.50)	2	52.62	100	0	76.41
ϵ=1 (pvb=0.12)	8	122.15	100	0	50.8
ϵ=1 (pvb=0.25)	4	52.48	100	0	76.46
ϵ=1 (pvb=0.33)	3	38.29	100	0	82.37
ϵ=1 (pvb=0.50)	2	24.95	100	0	88.2
ϵ=1.5 (pvb=0.12)	8	75.67	100	0	84.32
ϵ=1.5 (pvb=0.25)	4	33.90	100	0	88.97
ϵ=1.5 (pvb=0.33)	3	25.02	100	0	90.66
ϵ=1.5 (pvb=0.50)	2	16.52	100	0	92.87
ϵ=2 (pvb=0.12)	8	55.95	100	0	86.08
ϵ=2 (pvb=0.25)	4	25.15	100	0	90.64
ϵ=2 (pvb=0.33)	3	18.78	100	0	92.27
ϵ=2 (pvb=0.50)	2	12.33	100	0	94.11
I-GWAS	∞	0	66	5.53	66

Analyzing the impact of noise with ϵ-DP. Now, assuming the same scenario, the accuracy loss impact of ϵ-DP on allele frequencies of each SNP is evaluated in more detail. The error bar plots show the average (black circles, ideally at "0", i.e., without noise), standard deviation (black rectangles), minimal and maximal values (grey lines) for accuracy loss of the DP-based releases over 100 repetitions. Figure 5.6 presents a cut-off of the first 200 SNP positions of the release for the second study using $\epsilon = 2$ and $pvb = 0.12$ (the setting with highest utility score with DP) repeated 100 times.

Allele frequencies were applied 4.18% of noise on average, with a 0.41% average standard deviation, which kept the perturbation applied over SNPs results in the 3.77 - 4.59% accuracy loss interval. For some SNPs, original statistics were distorted above 40%, e.g., SNP ID 3, 88 and 97. Using **I-GWAS**, the same study released statistics over 66% of the original SNP-set without any noise addition.

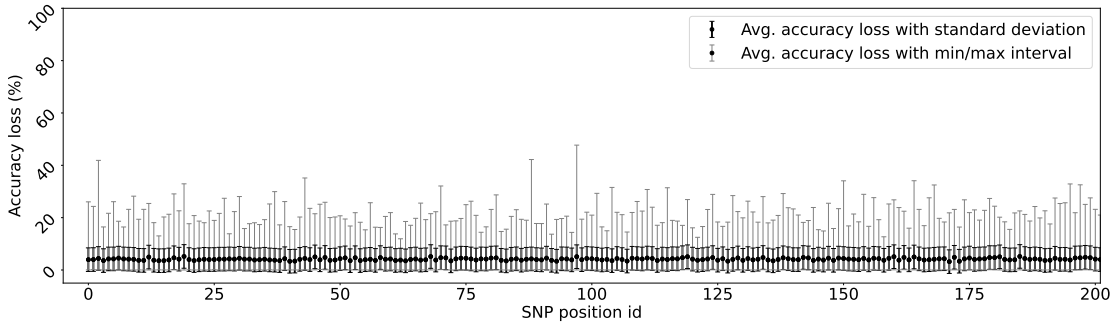


Figure 5.6: Results of GWAS releases using ϵ-DP releases using ($\epsilon = 0.1$ and $pvb = 0.12$). Cut-off of the first 200 of 1,000 SNPs.

5.5.3 Running time and complexity

Figure 5.7 shows the running time of **I-GWAS** for a GWAS that overlaps with other 5 GWASes that consider 500 (left side) or $1,000$ SNPs (right side).

Half of the genomes and SNPs that participated in the G previous GWASes releases were re-used by the current GWAS. First, a batch of requests for the candidate release is selected individually, and then its requests are checked against previous overlapping studies. Evaluating all possible combinations of existing releases with the brute force method does not scale as the associated complexity increases exponentially. However, thanks to Theorem 5, **I-GWAS**'s running time scale linearly with the number of existing releases and are shorter than those of a brute force approach. In addition, **I-GWAS** has a linear complexity even when considering a larger number of SNPs. For instance, with 1,000 SNPs, **I-GWAS** running time varied from 111 seconds for 2 GWASes to 157 seconds for 5

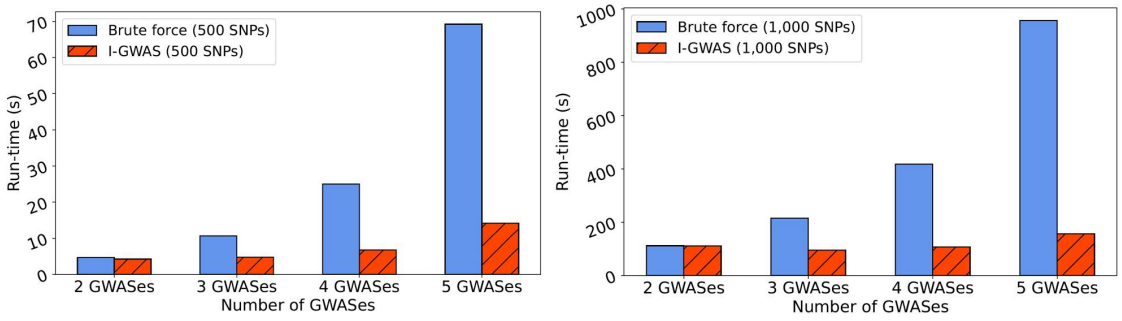


Figure 5.7: Running times of the brute force and I-GWAS approach for protecting recovery attacks on interdependent GWASes.

GWASes, whereas the brute force approach lasted from 111 seconds to 955 seconds, respectively. A similar behavior also happened with the experiments over 500 SNPs. While **I-GWAS** varied from 4 to 14 seconds, the brute force needed 4 to 69 seconds according to the number of GWASes, respectively. For the membership inference protection, it was measured the average for one analysis over the largest dataset (i.e.,14,860 genomes and 10,000 SNPs). **I-GWAS** computes this verification in 12.73 seconds. Keeping the number of SNPs and considering fewer genomes, **I-GWAS** takes on average 6.45 seconds for 9,906 genomes, 4.61 seconds for 7,430 genomes, and 2.96 s for 4,953 genomes. Lastly, the average memory consumption of **I-GWAS** in the enclave was 2 MB. Hence, respecting SGX's memory limitations.

Chapter 6

Genome Distributed Private Release (GenDPR)

Even though some works are concerned with conducting secure and privacy-preserving federated GWAS in a distributed manner, they did not certify that GWAS releases might be vulnerable to genomic privacy attacks. Motivated by that, previous chapters presented solutions to reconcile privacy-preserving *processing* and *releasing* of GWASes. Notwithstanding, existing solutions that allow the creation of privacy-preserving GWAS *releasing* require genome data to be pooled (usually in a centralized location) by a trusted curator or computing module. While DP-based mechanisms need to sample and access actual genome data to determine the perturbation levels to be applied on the original data to create differentially private releases, statistical inference methods execute computations to measure the probability of identifying the presence of individuals in the dataset.

This chapter offers **GenDPR**, a novel workflow that enables members of a GWAS federation to distributively verify and create safe releases of GWAS results without requiring genome data outsourcing and not accessing centralized genome data. The members of **GenDPR** jointly perform the statistical privacy-protection mechanisms to impede membership inference attacks from the observation of GWAS releases. In addition, **GenDPR** can also cope with the presence of colluding members trying to gain knowledge of genomic data of other members of the federation.

The previous chapters presented solutions to protect GWAS releases against membership attacks using statistical inference methods, more specifically using SecureGenome [San+09a; San+09b] introduced in Section 4.2. However, to conduct SG's privacy-protection analysis, genome data needs to be pooled, which are located in a centralized location and, in our case, in a TEE-enabled centralized server. Note that existing DP-based mechanisms also needs genome data to be pooled by a trusted curator/aggregator responsible for defining the perturbation

levels needed to protect the results. Therefore, this thesis identifies the following drawbacks of existing centralized workflows:

1. Requires outsourcing of actual genome data: the existing solutions need access to the genome sequences of the individuals to compute the LR-test or define perturbation levels. In particular, such data is kept in a centralized TEE-enabled machine that operates on the received data.

2. Single point of failure vulnerability: lack of availability of the centralized TEE server disrupts the whole workflow, and as a consequence the liveness of the GWAS.

3. Scalability/computational resources limitation: the centralized TEE-enabled server needs to store all genomic data from biocenters in the federation; TEEs usually suffer from limited memory, which might limit its overall performance.

Aiming at surpassing the above-mentioned issues, this chapter offers a novel mechanism to improve the manner that the assessments of privacy-preserving GWAS are enforced. In particular, this chapter strives for a solution that:

- Keeps all genomic data inside the entrusted institutions' premises by avoiding centralization and actual genome data outsourcing.

- Ensures protection despite possible the collusion of all-but-one federation members, by concealing data even if peer data is known.

- Produces at least the same privacy guarantees as centralized solutions, by correctly identifying the same data in need for protection that the centralized architecture would identify.

GenDPR is comprised of an untrusted part (from the perspective of other federation members) that exclusively accesses local genomic data and a trusted part that combines intermediary information from peer members to identify which subset of variants can be safely used for the subsequent secure GWAS computation. By exchanging only intermediate data, such as allele count vectors and local correlation metrics instead of the genomic variants in relation to a reference genome (recall that a VCF data file can easily amount to 100 GB), GenDPR significantly reduces the secure storage requirements on the central computing device, respectively in our case of the member TEE that is elected to assess safety based on this intermediate data.

GenDPR outsources and communicates intermediate data in encrypted forms and only to properly authenticated TEEs, as a release of such information would still enable membership and inference attacks, albeit with a much reduced chance of success.

6.1 GenDPR' system and threat models

System model. GenDPR considers a similar system models as of previous ones approaches. Nevertheless, now it is assumed that each federation member has its own TEE-enabled server. Thus, creating a multi-enclave setting. In summary, GenDPR considers a federation comprised of B biocenters $\{bioc_1, \ldots, bioc_B\}$, each being entrusted with genomic information and authorized to use and release this information in GWAS studies. Federation members and biocenters are used interchangeably. On premise, each biocenter maintains a database with genomes, servers to perform operations over this data and a TEE-enabled server that is mutually trusted by other federation members, including, after remote attestation, the authenticity of the trusted part of GenDPR. The goal of GenDPR is to secure the privacy of individuals that have entrusted a correct federation member with their genomes even when the federation releases GWAS results or when other members become curious or get compromised.

To remain compliant with regulations, such as GDPR, GenDPR ensures that no genomic information is communicated across TEEs and all communication of intermediate data is encrypted and linked to the current instance of the trusted part of GenDPR in remote TEEs. Members have access to a same reference set public genome data set (e.g., a public genome dataset [Con+15a; Wal+11]) used in the LR-test.

Given a desired starting set of SNP positions L_{des}, GenDPR returns a reduced set $L_{safe} \subseteq L_{des}$ of SNPs that are safe to be considered in a subsequent GWAS. For this final GWAS, existing privacy-perserving federated GWAS approaches [Pas+21; Sad+18; BAZ20; Koc+19; Rai+18] may be used, by considering the risk for SNPs in $L_{des} \setminus L_{safe}$ or by leaving out these SNP positions in the first place.

Worth recalling that TEE-enabled data sealing mechanism is used by local enclaves to persist secret data outside the trusted zone, which can be retrieved later for further processing. Sealed data can only be encrypted/decrypted by the enclave using its unique key. Additionally, it is assumed that appropriate countermeasures are in place to mitigate potential weaknesses of a concrete TEE limitations presented in Section 2.7.1. Figure 6.1 illustrates GenDPR's system and threat model.

Threat model. Like previous work on secure GWAS releases [Zha+18; Sad+18; Rai+18; Pas+21], GenDPR assumes adversaries capable of mounting membership attacks, by observing released GWAS statistics and metadata. In addition, GenDPR assumes up to f federation members might become faulty (e.g., as a result of a compromise). Collusion allows members to increase their knowledge, which increases their chance to mount membership attacks. GenDPR allows f to become as large as $B - 1$, but leave ensuring liveness in situations where feder-

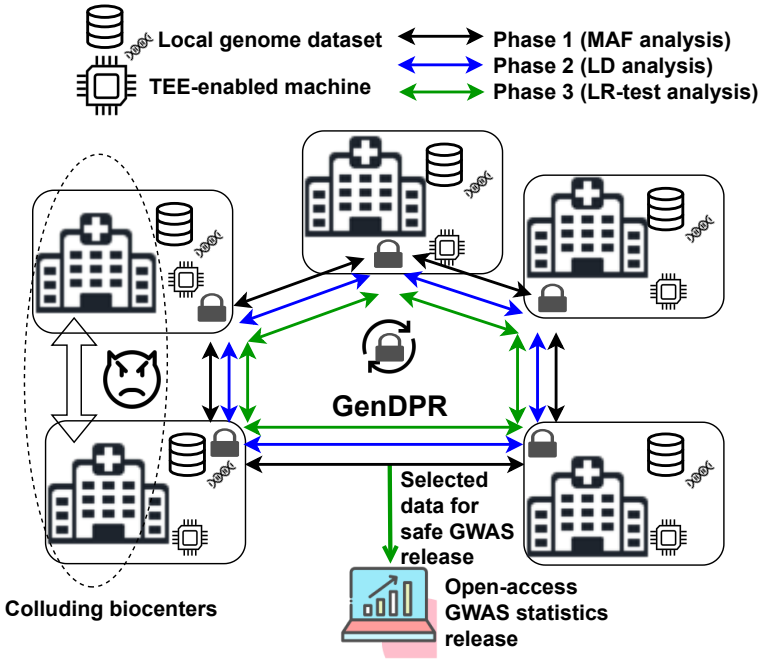


Figure 6.1: GenDPR system and threat model.

ation members refuse to execute GenDPR a question for future work. Also, as mentioned before, GenDPR does not consider leakage of genome information from the premises of compromised members, as this is an orthogonal problem.

While GenDPR considers that federation members might become honest-but-curious to mount attacks using collusion, it assumes that the integrity and confidentiality of TEEs remains intact. Moreover, GenDPR assumes the trusted part of the protocol is able to detect whether a federation member has tampered with the genome data and its accuracy (e.g., by checking the signature of signed .vcf files using hierarchical signature schemes).

Under the above assumptions, we show that as long as no TEE crashes, GenDPR produces a selection of SNP positions (L_{safe}) that is safe to be used for actual GWAS computation while protecting the privacy of individuals even if up to $f \leq G - 1$ federation members collude. Thus, the GWAS federation is properly considering the risks of including genetic variations that might compromise the privacy of its members.

6.2 Genome Distributed Private Release (GenDPR)

6.2.1 Architecture and overview

GenDPR's protocol starts when the federation agrees on conducting a particular GWAS aiming at releasing statistics over L_{des} SNP positions with specific MAF, LD and LR-test cutoff parameters. The protocol coordinates multiple enclaves hosted at each biocenter's premises. The coordination tasks are performed by the randomly elected leader that also performs aggregation and computations tasks leveraging intermediate inputs mutually shared by the members of the federation in a secure and private manner through its TEE-based architecture. There are two types of biocenters, regular biocenters and leader biocenter. Besides executing GWAS-specific computations (using the trusted MAF, LD and LR-test modules) inside the enclave (like regular biocenters), the leader biocenter also executes GenDPR's coordination algorithm. As participating enclaves have attested each other, they can trust on the code and the data outsourced by the members.

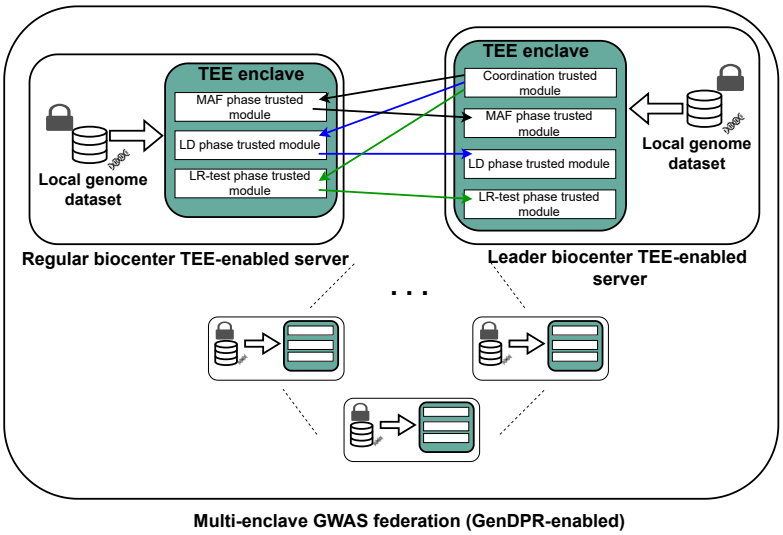


Figure 6.2: GenDPR architecture components.

Encrypted local genome datasets are used to feed local enclaves so that each biocenter enclave can produce and outsource genomic intermediate data requested by the leader biocenter according to the phase it starts. Biocenters only exchange data that is transmitted encrypted over the network. The TEE-enabled encryption scheme adopted by GenDPR allows decryption and encryption only inside and by mutually trusted enclaves. In particular, biocenters agree on the keys used among existing enclaves during a remote attestation phase.

Figure 6.2 presents GenDPR's multi-enclave architecture. For simplicity sake, we refrained from detailing the other regular biocenter enclaves in this figure.

6.2.2 Verification for mitigating recovery attacks

As introduced in previous chapters, DyPS and I-GWAS leverage theoretical complexity analysis to select a safe batch of genomes that can produce safe GWAS releases against recovery attacks. Such a verification is performed using the genome operations (addition or removal from studies) of the federation members. Thus, no genome data needs to be shipped during this phase. As a result, only the integrity and confidentiality of the biocenters genome requests need to be protected, which is achieved by leveraging the TEE-based architecture.

Hence, before running the distributed membership inference tests, GenDPR selects the genomes that are safe against recovery attacks. For that purpose, the randomly elected leader biocenter receives all genome operations from the other biocenters and execute the collusion-tolerant solution space analysis introduced in Section 4.5 to select the genomes allowed to advance to the next verification.

After this step, GenDPR starts the distributed algorithm to verify the data that can be used to produce safe releases against membership attacks, which are detailed in the next sections.

6.2.3 Workflow

Distributing the task of determining the data (genomes and genetic variations) over which GWAS statistics can be safely released while sending the minimum amount of data, is not trivial. Several works rely on the existence of a centralized server that stores actual genomes and is responsible for running such verifications. Therefore, conducting such analysis in a distributed fashion and only leveraging summary and intermediate genomic data from federation members is a challenge. Indeed, if not designed correctly, it can lead to inaccurate verification (as shown later in Section 6.3).

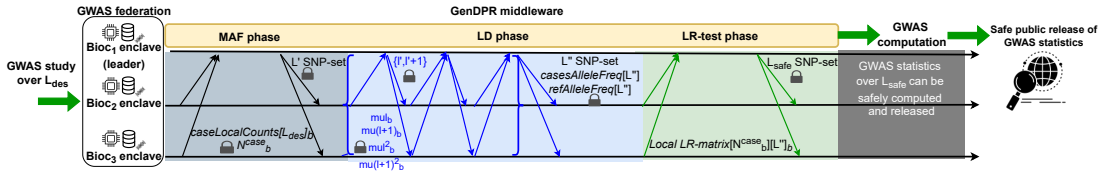


Figure 6.3: GenDPR workflow.

Figure 6.3 presents an overview of GenDPR's workflow. GenDPR is divided into several consecutive phases. Biocenters compute and outsource different intermediate computation results depending on the phase. One of the biocenters is randomly chosen and functions as an aggregator of inputs from the other biocenters and acts as the coordinator of the protocol. In particular, at the beginning of a GWAS and before the start of the distributed computations, GenDPR initiates two essential *pre-processing* tasks. Namely, (i) the leader enclave selection, which consists of randomly choosing one of the registered enclaves among the participants of the federation, which (ii) requests the local computation of summary statistics, e.g., allele counts vector of each biocenter over the original SNP-set L_{des} of the GWAS. Therefore, biocenters locally compute ${N_1^{case}}_l$ for each $l \in L_{des}$. Such a vector is identified as $caseLocalCounts[L_{des}]_b$ of size L_{des} and is sent by each biocenter b to the leader biocenters's enclave. The biocenters also share the number of individuals in their local case population (${N^{case}}_b$).

Biocenters compute and outsource different intermediate computation results depending on the phase. One of the biocenters is randomly chosen and functions as an aggregator of inputs from the other biocenters and acts as the coordinator of the protocol. In particular, at the beginning of a GWAS and before the start of the distributed computations, GenDPR initiates two essential *pre-processing* tasks. Namely, (i) the leader enclave selection, which consists of randomly choosing one of the registered enclaves among the participants of the federation, which (ii) requests the local computation of summary statistics, e.g., allele counts vector of each biocenter over the original SNP-set L_{des} of the GWAS. Therefore, biocenters locally compute ${N_1^{case}}_l$ for each $l \in L_{des}$. Such a vector is identified as $caseLocalCounts[L_{des}]_b$ of size L_{des} and is sent by each biocenter b to the leader biocenter's enclave. In addition, the biocenters share the number of individuals in their local case population, i.e., ${N^{case}}_b$.

Note that the leader biocenter does not need to outsource its intermediate results as it can compute its local summary statistics while aggregating the other biocenters inputs locally. In addition, a new leader can be elected at the beginning of each phase since crucial data for the progress of the protocol is broadcast by the leader at the end of a phase, which enables other biocenter's enclave to assume the leader position. Moreover, it is important to recall that all inputs and outputs of GenDPR are encrypted so that only authenticated (due to remote attestation) enclaves are able to encrypt/decrypt them. The following details each phase of GenDPR.

6.2.4 MAF analysis (Phase 1)

GenDPR's MAF analysis is straightforward. First, the leader enclave locally computes the allele counts vector of the reference population ($referenceLocalCounts[L_{des}]$

also of size L_{des}) and size $N^{reference}$. Then, after receiving the encrypted $caseLocalCounts_b$ and ${N^{case}}_b$ from each biocenter, the leader enclave decrypts and starts the MAF verification. In particular, the leader enclave sums all ${N^{case}}_b$ received from the biocenters with $N^{reference}$ into N_T. Then, the leader biocenter goes over the received inputs to calculate the allele counts of SNPs in both populations (case and reference) and then computes the global MAF of each SNP. More specifically, for each l in the original SNP-set L_{des} and for each biocenter's b allele counts vector, the leader biocenter computes $totalGlobalCounts[l] = caseLocalCounts[l]_g + referenceLocalCounts[l]$. The aggregated result is then divided by N_T in order to get the MAF for SNP l, i.e., $globalAlleleFreq[l] = totalGlobalCounts[l]/N_T$. Finally, the leader checks if $\mathbf{MAF}_l < MAF_{cutoff}$. If so, SNP l is placed on a blacklist and therefore will not be considered for release. In that way, GenDPR manages to perform the removal of rare MAF SNP positions without demanding the outsourcing of actual genomes from the biocenters. Intuitively, the leader biocenter identifies a list of retained SNPs $L' \in L_{des}$ that is broadcast to the federation in order to continue the next steps of the protocol.

GenDPR's MAF analysis is straightforward. First, the leader enclave locally computes the allele counts vector of the reference population ($referenceLocalCounts[L_{des}]$ also of size L_{des}) and size $N^{reference}$. Then, after receiving the encrypted $caseLocalCounts_b$ and ${N^{case}}_b$ from each biocenter, the leader enclave decrypts and starts the MAF verification. In particular, the leader enclave sums all ${N^{case}}_b$ received from the biocenters with $N^{reference}$ into N_T. Then, the leader biocenter goes over the received inputs to calculate the allele counts of SNPs in both populations (case and reference) and then computes the global MAF of each SNP. More specifically, for each l in the original SNP-set L_{des} and for each biocenter's b allele counts vector, the leader biocenter computes $totalGlobalCounts[l] = caseLocalCounts[l]_b + referenceLocalCounts[l]$. The aggregated result is then divided by N_T in order to get the MAF for SNP l, i.e., $globalAlleleFreq[l] = totalGlobalCounts[l]/N_T$. Finally, the leader checks if $\mathbf{MAF}_l < MAF_{cutoff}$. If so, SNP l is placed on a blacklist and therefore will not be considered for release. In that way, GenDPR manages to perform the removal of rare MAF SNP positions without demanding the outsourcing of actual genomes from the biocenters. Intuitively, the leader biocenter identifies a list of retained SNPs $L' \in L_{des}$ that is broadcast to the federation in order to continue the next steps of the protocol.

6.2.5 LD analysis (Phase 2)

The next step consists of executing the LD verification over the retained L' SNPs, so that all SNPs that will be potentially released are independent from each other. To compute LD, allele information between two SNPs needs to be pooled. It is easily achievable in a centralized TEE-based architecture because all genomes are

locally available. GenDPR, however, cannot benefit from such availability as it can only rely on intermediate data from biocenters and is thus not able to pool the allele sequences for LD computation. One could naïvely let each biocenters conduct the LD analysis locally and share their locally retained SNPs. Nevertheless, assuming this approach, each biocenter would inaccurately select different SNPs because biocenters own different genomes, implying heterogeneous distributions that lead to different correlation statistics.

GenDPR employs the following adaptations for removing SNPs in LD. When computing the LD between every pair of SNP l and $l+1 \in L'$, local allele sequences of individuals need to be pooled to compute correlation statistics. Therefore, each biocenter b enclave locally produces and outsources the following correlation statistics over their genomes: μ_{l_b} += $\mathbf{SNP}_{l_b}$, μ_{l+1_b} += $\mathbf{SNP}_{l+1_b}$, $\mu_{(l,l+1)_b}$ += $\mathbf{SNP}_{l_b} * \mathbf{SNP}_{l+1_b}$, $\mu_{l_b^2}$ += $\mathbf{SNP}_{l_b} * \mathbf{SNP}_{l_b}$, $\mu_{(l+1)_b^2}$ += $\mathbf{SNP}_{l+1_b} * \mathbf{SNP}_{l+1_b}$, and N_T (acquired during the previous phase). The leader biocenter computes the same correlation statistics over the reference set.

Upon reception, the leader enclave aggregates biocenters inputs with the correlation metrics obtained over the reference set. This way, GenDPR can collectively absorbs the correlation statistics from each biocenter so that the aggregated correlation metrics reflect the global genome distribution of the federation for the right computation of LD. After that, the leader enclave can proceed with the computation of the p-value on the r^2 test to measure the level of correlation between the two SNPs. If $\mathbf{LD}_{(l,l+1)} < LD_{cutoff}$, then SNPs l and $l+1$ are dependent, and therefore GenDPR keeps the most ranked one (in terms of p-value on χ^2) for the next iterations of the algorithm. When this evaluation ends, the leader biocenter has identified a new SNP-subset $L'' \in L'$ that is broadcast for supporting the next phase. This process is repeated at most $(L')^2$ times, considering a very rare case where all pairs of SNPs are found to be independent, which is not a common event in the human genome [BM12; Bar+12].

6.2.6 LR-test analysis (Phase 3)

To perform the LR-test verification the actual allele information of SNPs of each participant is needed (i.e., $x_{n,l}$ in Equation 4.3). Therefore, to successfully conduct the LR-test, existing solutions rely on the availability of genomes in a centralized enclave. On the other hand, GenDPR overcomes such constraint by demanding each biocenter to compute and outsource their local LR-matrices. However, biocenters cannot correctly compute these matrices leveraging their local genome dataset distribution, otherwise conclusions of the test are incorrect. Indeed, using local frequencies would lead to wrong LR-matrices because the LR-test needs to be drawn considering SNP frequencies of the whole population (i.e., genomes belonging to all biocenters). Therefore, allele frequencies over the full cohort is

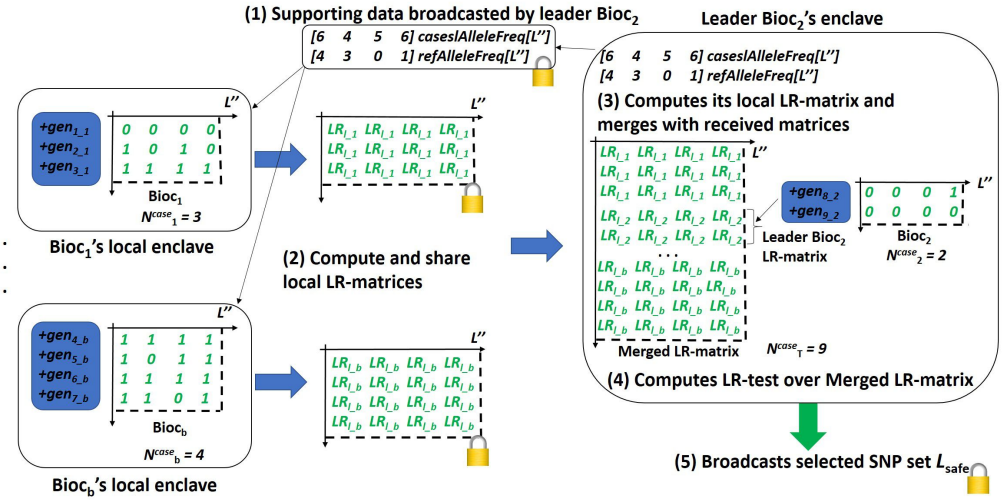


Figure 6.4: GenDPR distributed LR-test phase scheme.

needed so that each biocenter can accurately compute their local LR-matrix. The local LR-matrix consists of the LR values (recall Equation 4.3) for each SNP l and allele value of individual n at SNP position l represented by $x_{n,l}$ in each biocenter dataset.

The complete scheme enforced by GenDPR for the distributed LR-test evaluation is illustrated in Figure 6.4, where $\mathbf{Bioc}_2$ has been selected as the leader. In Step (1), the leader $\mathbf{Bioc}_2$ broadcasts the allele frequencies vector of the case and reference populations over the retained SNPs L'' (note that these vectors are already available inside the leader enclave since the MAF phase). Therefore, the $casesAlleleFreq[L'']$ and $refAlleleFreq[L'']$ vectors, both of size L'', are shared with all biocenters. These vectors represent p_l and $\hat{p}_l$ of Equation 4.3, respectively. In Step (2), after the reception of the allele frequencies vectors, each biocenter ($\mathbf{Bioc}_1$ to $\mathbf{Bioc}_b$) can correctly build their LR-matrices since the received vectors encompass the frequencies over the complete cohort of participating genomes. Therefore, their local LR-matrices can be correctly computed. After completion, biocenters encrypt and send their local LR-matrices to the leader biocenter. In Step (3), upon the reception of biocenter' LR-matrices, the leader $\mathbf{Bioc}_2$ first computes its local LR-matrix, and then merges all matrices received. Thus, creating a LR-matrix that covers all biocenters LR metrics. This matrix is used throughout the LR-test verification performed in Step (4) inside $\mathbf{Bioc}_2$'s enclave. This verification consists of empirically checking several subsets of SNPs in L'' that satisfies the conditions presented in Section 4.2. When the LR-test ends, the leader enclave has identified a new subset of SNPs $L_{safe} \in L''$, which is encrypted and broadcast to the members of the federation (Step (5)). The list of SNPs in L_{safe} can be

safely used for the computation and release of the GWAS.

Additionally, GenDPR can be combined with Differential Privacy (DP) [Dwo11] mechanisms to increase the data utility of releases. Particularly, the SNPs in L_{safe} can be released in a noise-free manner (i.e., without any data perturbation like DP applies), while SNP positions not present in L_{safe} but in L_{des} are released with DP-based perturbation. Thus, allowing GWASes to release statistics over all initially desired SNP positions (L_{des}) in a privacy-preserving manner. I plan to investigate such a technique in future work.

Algorithm 6 describes GenDPR's pseudocode. This algorithm reflects the behavior explained previously. Data encryption operations were not discussed during the workflow because it is standardized knowledge when leveraging TEE-based architectures. Therefore, only the rationale of the algorithm is presented.

GenDPR starts randomly selecting a biocenter in the federation that will behave as the leader of the protocol in line 6. Then, in line 9, the leader biocenter starts computing its local GWAS summary statistics, e.g., case allele counts and the number of individuals. It does the same over the genomes in the reference set. From that moment on, the leader can receive the summary statistics data of the other biocenters (locally computed when the federation agrees on starting a study). After collecting biocenters' intermediate data, the leader biocenter starts the MAF analysis by first aggregating local counts over the original SNP-set L_{des} of the study. It does the same for calculating the total number of individuals in the federation. Then, the leader biocenters finally computes the MAF of each SNP and checks the MAF cut-off, keeping only SNP positions with MAF above or equal to the MAF cut-off (MAF_{cutoff}). These steps are described in lines 10 to 24. At the end of this analysis, the leader biocenter has acquired a new SNP-subset L' consisting of the list of SNPs that survived this phase. Such a list of SNPs is broadcast to all biocenter in line 25.

Next, the leader biocenter initiates the LD analysis after receiving the correlation metrics of each biocenter of a pairwise combination of SNPs in L'. In particular, the LD verification algorithm (from lines 26 to 55) aggregates local correlation statistics from each biocenter and the ones corresponding to the reference set for SNPs pair l and $l+1$. In addition, the leader biocenter computes allele frequencies over L'' for the reference and global population (note that is achievable using the allele counts shared in the MAF analysis) that is further aggregated with the correlation metrics of all biocenters. After aggregation, the leader biocenter calculates the p-value for the correlation between the two SNPs. If SNPs are high-correlated, i.e., p-value below the LD cut-off (LD_{cutoff}), the leader biocenter keeps the most ranked SNP and proceeds the loop. SNPs that do not present a high pairwise correlation with others are retained in L'', which is also broadcast at the end of this phase. Finally, the allele frequencies vectors ($casesAlleleFreq[L''$] and

Algorithm 6 GenDPR's full workflow pseudocode

procedure GenDPR(GWAS g, B set of biocenters, original SNP set L_{des} of g, α, β, $ref\_population$)
  Inputs: (1) Local allele counts vector from biocenters of size L_{des}; (2) Local statistics of SNP$_l$ and SNP$_{l+1}$; (3) Local LR-matrix each of size ${N^{case}}_b$ x L''
  Outputs: (1) Selected SNP subset L'; (2) Selected SNP subset L''; (3) Selected SNP subset L_{safe}, which can be used to create private GWAS release
  Uses: $randomLeaderSelection(B)$: select and returns a random biocenter $b \in B$ to be considered as leader; $startLocalComputations()$: computes local statistics of biocenter; $computeR^2(\mu_l, \mu_{l+1}, \mu_{(l,l+1)}, \mu_{l^2}, \mu_{(l+1)^2}, N_T)$: returns p-value on r^2 between SNPs l and $l+1$; $getMostRanked(l, l+1, s)$: returns index of most ranked SNP (p-value on χ^2 of study g; $LRtest(LRMatrix, \alpha, \beta)$: returns a set of SNPs that keeps individuals identification power below given threshold

  $leader_{bioc} = randomLeaderSelection(B)$ //randomly selects a leader in B
  $leader_{bioc}.startLocalComputations()$ // computes leader and biocenters local allele statistics
  $leader_{bioc}.listenToInputs()$ // collects intermediate data from other biocenters

  (1) //MAF analysis
  for b in B **do** //retrieves local allele counts vector from each biocenter
    N_T += ${N^{case}}_b$
  end for
  N_T += $N^{reference}$
  for SNP l in L_{des} **do**
    for b in B **do** //retrieves local allele counts vector from each biocenter
      $totalGlobalCounts[l] = caseLocalCounts[l]_b + referenceLocalCounts[l]$
    end for
    $globalAlleleFreq[l] = totalGlobalCounts[l]/N_T$
    if $globalAlleleFreq[l] < MAF_{cutoff}$ **then** //SNP l cannot be retained
      continue
    else
      $L'.push(l)$
    end if
  end for
  $leader_{bioc}.broadcast(L')$ // leader biocenter broadcast message

  (2) //LD analysis
  $last_{index} = L'[-1]$ // get index of the last SNP in L'
  $aux_{index} = L'[0]$ // get index of the first SNP in L'
  while $aux_{index}! = last_{index}$ **do** // starts greedy algorithm for LD computation
    for SNP l in L' **do**
      $leader_{bioc}.listenToInputs()$ // collects intermediate correlation statistics from biocenters
      $leader_{bioc}.startLocalComputations()$ // computes leader local correlation statistics
      for b in B **do** //retrieves local LD statistics for SNP$_l$ and SNP$_{l+1}$ from each biocenter
        μ_l += ${\mu_l}_b$
        μ_{l+1} += ${\mu_{l+1}}_b$
        $\mu_{(l,l+1)}$ += ${\mu_{(l,l+1)}}_b$
        μ_{l^2} += $\mu_{l^2_b}$
        $\mu_{(l+1)^2}$ += $\mu_{(l+1)^2_b}$
      end for
      μ_l += $\mu_{l_{ref}}$
      μ_{l+1} += $\mu_{{l+1}_{ref}}$
      $\mu_{(l,l+1)}$ += $\mu_{(l,l+1)_{ref}}$
      μ_{l^2} += $\mu_{l^2_{ref}}$
      $\mu_{(l+1)^2}$ += $\mu_{(l+1)^2_{ref}}$
      $pval = computeR^2(\mu_l, \mu_{l+1}, \mu_{(l,l+1)}, \mu_{l^2}, \mu_{(l+1)^2}, N_T)$
      if $pval > LD_{cutoff}$ **then** //independent SNPs
        $aux_{index} = l + 1$
        continue
      else //dependent SNPs, keep most ranked one
        $l_{index} = getMostRanked(l, l+1, s)$
        $L''.push(l_{index})$
      end if
    end for
    $aux_{index} = l + 1$
  end while
  $leader_{bioc}.broadcast(L'', casesAlleleFreq[L''], refAlleleFreq[L''])$ // leader biocenter broadcast message

  (3) //LR-test analysis
  $leader_{bioc}.listenToInputs()$ // collects local LR-matrices from biocenters
  $leader_{bioc}.startLocalComputations()$ // computes leader local LR-matrix
  for b in B **do** //retrieves and concatenates local LR-matrix from each biocenter
    for SNP l in L'' **do**
      $FullLRMatrix[l]+ = LRmatrix_b[l]$
    end for
  end for
  $L_{safe} = LRtest(FullLRMatrix, \alpha, \beta)$ //runs LR-test analysis over full matrix
  return L_{safe} //final subset of SNPs for safe GWAS g release
end procedure

$refAlleleFreq[L'']$) are broadcast to the biocenters in line 56.

Lastly, the leader biocenter needs to perform the LR-test to find the final list of safe SNPs. This verification starts in line 58, where the leader biocenter receives the local LR-matrices from each biocenter that are locally computed by each biocenters using $casesAlleleFreq[L'']$ and $refAlleleFreq[L'']$ shared in the previous phase. Upon the reception of the local LR-matrices, the leader biocenter loops over L'' to merge all received LR-matrices with its local matrix (lines 60 to 64). Next, in line 65, the leader biocenter runs the LR-test function over the merged matrix that empirically finds a subset $L_{safe} \in L''$ of which releases over these SNPs do allow membership inference attacks to succeed. Finally, leader biocenter broadcasts L_{safe} SNP-set list in line 66.

6.2.7 Collusion-tolerant GenDPR

To protect the GWAS federation against collusion among biocenters, GenDPR's leader enclave needs to certify that the outcome of the private analysis is valid for the cases where up to $f \leq B-1$ colluding biocenters attempt to attack the honest ones. For this purpose, GenDPR employs a collusion-tolerant algorithm. For each phase of GenDPR's pipeline discussed above, GenDPR generates the $\binom{B}{B-f}$ combinations of intermediate results received from the biocenters to simulate the case where f biocenters would launch an attack. Each of these combinations has a unique identifier and goes through the various phases of GenDPR, which identify a list of safe SNPs. At the end of each phase, GenDPR computes the intersection of the SNPs chosen for each combination, thus preventing any f biocenters to to compromise the data of honest biocenters. Let us discuss an example for Phase 3 (the most complex phase).

During the LR-test phase, the leader enclave generates and provides an unique id, and broadcasts $\binom{B}{B-f}$ allele frequency vectors of L'' SNPs selected in the previous LD analysis phase. As a consequence, the leader receives $\binom{B}{B-f}$ local matrices (each one computed using its corresponding frequency vector) from each biocenter. Each combination of sub-matrices forms a unique merged matrix that is used for the actual LR-test evaluation inside the leader enclave. As a result, GenDPR collects several lists of selected SNPs L_{safe} executed over each matrix, i.e., one SNP list is output for each LR-test completed over combinations. Finally, the leader enclave computes the intersection among the lists of SNPs, and finally outputs only the intersected SNPs, i.e., SNPs that were labeled as safe in every combination. This way, GenDPR certifies that no combination of genome data can be isolated and become vulnerable to colluding biocenters.

During the LR-test phase, the leader enclave generates and provides an unique id, and broadcasts $\binom{B}{B-f}$ allele frequency vectors over L'' SNPs selected in the

previous LD analysis phase. As a consequence, the leader receives $\binom{B}{B-f}$ local matrices (each one computed using its corresponding frequency vector) from each biocenter. Each combination of sub-matrices forms a unique merged matrix that is used for the actual LR-test evaluation inside the leader enclave. As a result, GenDPR collects several lists of selected SNPs (L_{safe}), i.e., one for each LR-test completed over each combination of matrices. Finally, the leader enclave computes the intersection among the lists of SNPs, and finally outputs only the intersected SNPs, i.e., SNPs that were mutually labeled as safe in every combination. This way, GenDPR certifies that no combination of genome data can be isolated and become vulnerable to colluding biocenters.

GenDPR can also adhere to a more conservative approach assuming all possibilities of collusions instead of considering a static f, i.e., $f = \{1, ..., B-1\}$. GenDPR would then perform evaluations over $\sum_{f=1}^{f=B-1} \binom{B}{B-f}$.

As one would expect, this scheme demands GenDPR to execute extra rounds of computations, which in practice can be efficiently conducted in parallel inside the leader enclave as it already stores all necessary data. The following details the extension applied by GenDPR to enable collusion-tolerance.

To avoid repetition, Algorithm 6 should be recalled to understand the required modifications that GenDPR enforces to accommodate collusion-tolerance.

To enable collusion-tolerance, GenDPR needs to execute the analysis over each combination of data that can be actually isolated by colluding biocenters to mount membership attacks against honest biocenters or to distort the correct output of selected SNPs. To that extent, after retrieving intermediate data from each biocenter in each phase, GenDPR forms $\binom{B}{B-f}$ combination with the received inputs to simulate the fraction of data that could be isolated by the colluding biocenters depending on f. Therefore the original set B consisted of b biocenters becomes a new set of combination of biocenters so that the verification can be computed for every combination, represented as $combBiocSet = combineBioc(B)$. $combineBioc(B)$ is a function that receive the set of biocenters B and outputs a new set consisted of $\binom{B}{B-f}$ combinations.

As a result, the loop to acquire biocenters data is done throughout this new $combBiocSet$. For instance, the loop for MAF analysis in line 10 of Algorithm 6 is performed over $combBiocSet = \binom{B}{B-f}$ instead of the original set B. The same behavior is applied to the other phases of GenDPR's protocol. Namely, in line 33 for the LD analysis and line line 60 for the LR-test.

Besides that, GenDPR also need to keep a data structure to store the list of selected SNPs of each iteration. This is needed so that GenDPR can compute the intersection of SNPs selected as safe in all combinations. In fact, at the end of each phase, only SNPs present in all lists are going to be broadcast to the federation because they are safe independently of the presence of colluders. For example, con-

sidering the MAF phase again, GenDPR appends each L' to a new data structure called $L'\_ListSet$ after line 23. Once the loop over $combBiocSet$ ends, GenDPR computes $finalL' = getIntersection(L'\_ListSet)$ function that receives a set of SNP lists and returns a list of SNPs mutually chosen in all combinations. The SNPs in $finalL$ guarantees that no combination of intermediate results leveraged by colluding parties can be used to launch successful membership attacks.

This method to compute the intersection of SNPs is performed at the end of each phase before data is broadcast by the leader biocenter. More specifically, the $getIntersection(L)$ function to find the SNPs intersection over the list outputted for each iteration is executed before line 25 for MAF analysis, line 56 for the LD phase and before line 66 after the LR-test verification, and then acquiring the final intersected list of SNPs L_{safe} that can be safely used in a release.

6.3 Experimental evaluation

GenDPR is also implemented in C/C++ using the Graphene SGX library [TPV17] and evaluated its performance on an Intel i7-8650U processor with 16 GB RAM, running Ubuntu 18.04. The experiments used 27,895 genomes from the dbGaP (phy001039.v1.p1) dataset for an Age-Related Macular Degeneration study [Wal+11]. The dataset contains of 14,860 case genomes and 13,035 control genomes. The control population set was used as a reference for the LR-test. Additionally, the genomes were divided equally among federation members. GenDPR is evaluated using SecureGenome's suggested settings [San+09b] – 0.05 MAF cut-off, 10^{-5} LD cut-off, 0.1 false-positive rate and 0.9 identification power threshold – which is also used as baseline (centralized version). All exchanged data is encrypted using AES 256. The experiments assume from 2 to 7 federation members (biocenters) and from 1,000 to 10,000 SNP positions. We report the average of 5 five repetitions. We also compared GenDPR with a centralized approach that runs SecureGenome inside a centralized TEE enclave, which we use as Baseline.

6.3.1 Bandwidth, memory and CPU usage

Table 6.2 shows the average resource demands for GenDPR for different configurations of the federation and the GWAS it performs. As can be seen, all scenarios remain below 1% of CPU utilization and below 2 MB of data that needs to be exchanged on average among the federation members. Biocenters exchange vectors of integers that require 32 bits for each SNP in the original dataset L_{des}. Hence, the overall size of data that needs to be exchanged is $(4 \cdot L_{des})$ Bytes, which increases by approximately 30% after encryption due to padding.

Table 6.1: Average resource demand of GenDPR.

Configuration	Avg. CPU utilization	Avg. Memory demand
2 Biocenters / 1,000 SNPs	< 1%	2,068 KB
2 Biocenters / 10,000 SNPs	< 1%	2,164 KB
3 Biocenters / 1,000 SNPs	< 1%	2,068 KB
3 Biocenters / 10,000 SNPs	< 1%	2,172 KB
5 Biocenters / 1,000 SNPs	< 1%	2,074 KB
5 Biocenters / 10,000 SNPs	< 1%	2,148 KB
7 Biocenters / 1,000 SNPs	< 1%	2,052 KB
7 Biocenters / 10,000 SNPs	< 1%	2,180 KB

With GenDPR, biocenters do not need to outsource genome sequences, which saves $2 \cdot L_{des}$ bits for every genome and $2 \cdot L_{des} \cdot N_T$ bits in total.

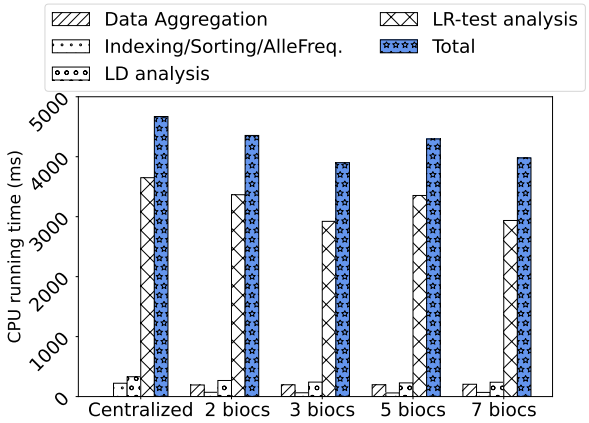
Notice, the data that need to be exchanged in subsequent steps becomes even lower as they operate only on a subset of the initially desired SNPs. Indeed, for the LR-test phase, each biocenter shares smaller data, i.e., over $L'' \cdot {N^{case}}_b$, which is a magnitude order smaller than complete genome sequences.

In summary, it can be seen that GenDPR's performance scales well with an increasing number of biocenters and SNPs considered and that it remains well within the resource limitations found in today's TEEs.

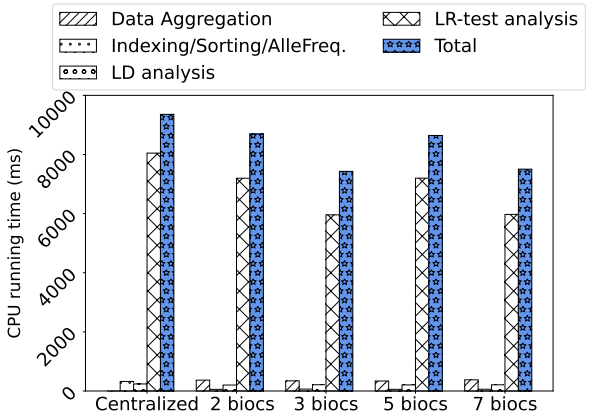
6.3.2 Running time

In Figures 6.5 and 6.6, we report GenDPR's running time compared to the Baseline approach for each task performed during each phase while considering several GWAS settings. Firstly, we can notice that even though not demanding any data aggregation tasks, the centralized solution is not relatively quicker than GenDPR. Particularly, the running times of both directly depend on the size of the data that needs to be evaluated. Comparing Figures 6.5a with 6.5b, and Figures 6.6a with 6.6b, we can notice that the number of genomes and SNPs considered increases the magnitude order of the running time of both approaches. Therefore, we claim that GenDPR is scalable since that doubling the number of genomes considered at first (7,430) and considering 10 times more SNPs in a study have not imposed a burden to the distributed protocol. Overall, GenDPR finishes in reasonable time.

Moreover, we can see that increasing the number of biocenters to more than two, actually decreases the running time of the protocol since the computational tasks are distributed among members, which reduces running time compared to the centralized architecture. In contrast, the centralized version cannot take advantage of such a feature, and therefore needs to process all the data at once. Hence, we claim that GenDPR also benefits from the workload distribution achieved thanks



(a) 7,430 genomes / 1,000 SNPs.



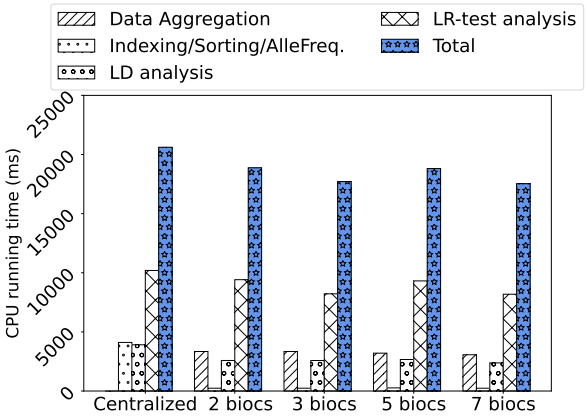
(b) 14,860 genomes / 1,000 SNPs.

Figure 6.5: Running time comparison (1,000 SNPs).

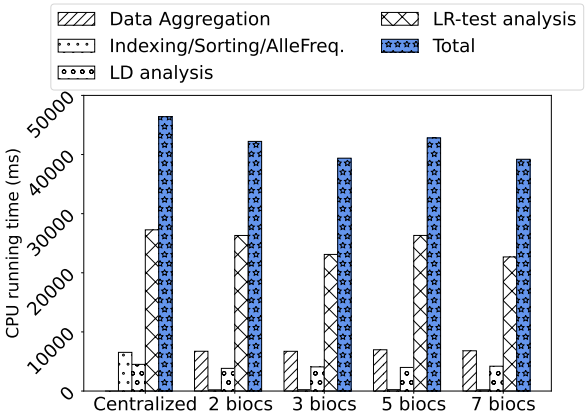
to its distributed protocol.

Comparing the running times of each phase we can notice that the LR-test analysis is the longest due to the fact that besides operating on larger data structures (2D matrix instead of 1D vectors as in previous phases), GenDPR uses an empirical approach when selecting the safe SNP-subset among the available SNPs, which require some iterations over several sets of SNPs. In general, GenDPR only imposes slightly longer running time due to the extra coordination and aggregation tasks performed by the leader, but the presence of more of more members in the federations can actually improves GenDPR's running time.

An interesting phenomenon identified is that although the scenario with 5 biocenters presented a longer running time compared to the scenarios with 3 and 7 biocenters, it is approximately as long as the scenario with 2 biocenters. De-



(a) 7,430 genomes / 10,000 SNPs.



(b) 14,860 genomes / 10,000 SNPs.

Figure 6.6: Running time comparison (10,000 SNPs).

spite that, GenDPR's distributed protocol is faster than the centralized (Baseline) approach in all settings.

It is important to note that GenDPR's running time depends on the distribution of the genome data being assessed in the analysis. For instance, for some populations more or fewer SNPs are removed at each phase. In particular, a higher number of retained SNPs through the phases means increased running time since statistics need to be computed over a larger space.

6.3.3 Correctness

To assert correctness, it is compared the SNP positions selected as safe by GenDPR, by the Baseline (centralized TEE-enabled SecureGenome approach), and by a lim-

ited distributed protocol that uses naïve aggregation (where the LD and LR-test analyses – steps that require access to allele information – are run locally by each biocenter leveraging allele frequency vectors shared by the leader). While the LD verification needs to pool pairwise allele statistics over all individuals, the LR-test requires pooling all genomes to produce the LR-matrix used in the test. In the naïve approach, each biocenter computes the LD and LR-test independently and shares an encrypted vector with selected SNP indexes, of which an intersection is later computed to obtain the final list of SNPs.

Table 6.2 presents the number of SNPs retained as safe after each phase of the privacy-protecting evaluation obtained considering 7,430 or 14,860 case genomes and several number of SNPs. First, it can be noticed that changing the number of biocenters in the federation does not affect the outcome of the verification. In addition, it is noted that GenDPR imitates the behavior of the Baseline over all verification phases, which shows that GenDPR is correct and does not suffer from perturbation throughout its execution.

Moreover, if intermediate data is not aggregated and considered correctly, i.e., using a naïve aggregation algorithm, it can lead to wrong SNP selection. Indeed, it was detected that even though such a scheme is able to retain the same SNPs during the MAF evaluation, it is not able to correctly perform the LD and LR-test analyses since these latter verifications need to consider the global genome distribution to correctly identify safe SNPs, which is not enforced with a naïve aggregation. This behavior is identified in the bold lines of Table 6.2, where the naïve protocol inappropriately identified a smaller and disjoint set of SNPs. The release of such SNPs would allow membership inference of participants in the study. On the other hand, the adjustments rendered in GenDPR thwart such issues, i.e., GenDPR selects the same set of SNPs as Baseline, which shows its accuracy.

Table 6.2: Comparison of the selected SNPs after each phase of the privacy-protecting verification.

Original number of SNPs	Baseline	GenDPR	Distributed with limited aggregation
7,430 genomes	Number of retained SNPs		
1,000	MAF 731 / LD 44 / LR 44	MAF 731 / LD 44 / LR 44	MAF 731 / **LD 29** / **LR 29**
2,500	MAF 1,559 / LD 107 / LR 107	MAF 1,559 / LD 107 / LR 107	MAF 1,559 / **LD 66** / **LR 12**
5,000	MAF 2,666 / LD 208 / LR 208	MAF 2,666 / LD 208 / LR 208	MAF 2,666 / **LD 127** / **LR 29**
10,000	MAF 4,584 / LD 375 / LR 375	MAF 4,584 / LD 375 / LR 375	MAF 4,584 / **LD 240** / **LR 240**
14,860 genomes			
1,000	MAF 303 / LD 25 / LR 25	MAF 303 / LD 25 / LR 25	MAF 303 / **LD 11** / **LR 11**
2,500	MAF 1,032 / LD 50 / LR 50	MAF 1,032 / LD 50 / LR 50	MAF 1,032 / **LD 22** / **LR 22**
5,000	MAF 2,021 / LD 105 / LR 105	MAF 2,021 / LD 105 / LR 105	MAF 2,021 / **LD 44** / **LR 44**
10,000	MAF 3,767 / LD 187 / LR 187	MAF 3,767 / LD 187 / LR 187	MAF 3,767 / **LD 80** / **LR 80**

6.3.4 Collusion-tolerant GenDPR

This section discuss the experiments to evaluate the impact of the collusion-tolerant version of GenDPR in terms of privacy protection (detecting SNPs that would become vulnerable given the presence of colluders) and performance (running time and release coverage). Table 6.3 presents the results of collusion-tolerant GenDPR. It can be noticed that from 20.9% to 28.3% of the SNPs could have their statistics unsafely released when collusions happen and are not protected against. These vulnerable SNPs are secluded and refrained from being released. Thus, there is an expected impact on the number of SNPs being released proportional to the number of vulnerable SNPs. Despite that, collusion-tolerant GenDPR is still able to release from 71.7% to 79.1% of the data when compared to the experiments without collusion ($f = 0$) presented in Table 6.2.

Table 6.3: Collusion-tolerant GenDPR results considering 10,000 SNPs and 14,860 genomes.

Settings	# safe released SNPs **with** collusion detection	# vulnerable SNPs **without** collusion detection	Running time (ms)
$B = 3, f = 1$	141 (75.4%)	46 (24.6%)	123,338.5
$B = 3, f = 2$	143 (76.5%)	44 (23.5%)	76,362.5
$B = 3, f = \{1, 2\}$	138 (73.8%)	49 (26.2%)	158,059.5
$B = 4, f = 1$	143 (76.5%)	44 (23.5%)	159,293.2
$B = 4, f = 2$	139 (74.3%)	48 (25.7%)	156,569.9
$B = 4, f = 3$	145 (77.5%)	44 (22.5%)	80,681.4
$B = 4, f = \{1, 2, 3\}$	136 (72.7%)	51 (27.3%)	309,032.3
$B = 5, f = 1$	144 (77.1%)	43 (22.9%)	215,347.1
$B = 5, f = 2$	135 (72.1%)	52 (27.9%)	255,071.8
$B = 5, f = 3$	137 (73.3%)	50 (26.7%)	181,159
$B = 5, f = 4$	148 (79.1%)	39 (20.9%)	79,300.4
$B = 5, f = \{1, 2, 3, 4\}$	134 (71.7%)	53 (28.3%)	605,281.8

Overall, there is an increase in running time of the collusion-tolerant GenDPR due to the extra verifications conducted over biocenters' isolated data. Comparing the most conservative setting of GenDPR (in which all possible combination of colluders are considered, i.e., $f = \{1, ..., B - 1\}$) with the $f = 0$ case, it is noticed longer running times. For instance, the $B = 5, f = \{1, 2, 3, 4\}$ setting took 605 seconds while for $f = 0$, 44 seconds. Nevertheless, such an increment of running time is a reasonable trade-off to bring higher levels of privacy.

Another interesting result is the shorter running times achieved in the $f = B-1$ setting when compared to other values of f as presented in the last line of Table 6.3. This is explained by the fact that in this scenario the additional rounds of LR-tests only need to be performed considering each biocenter dataset individually, and therefore over fewer genomes. It is noticed that the number of safe SNPs

depends on how the distribution of the genome data impacts the identification power of participants during the LR-test evaluation. Therefore, there is no direct correlation between the number of genomes/SNPs with the number of safe SNPs. The experiments considering 1,000 SNPs showed a similar behavior.

Chapter 7

A Holistic Approach (combining DyPS, I-GWAS and GenDPR)

This chapter provides an overview of a federated GWAS framework to conduct *practical GWASes*, which consists of enforcing all the functionalities described in the previous chapters simultaneously. In particular, it describes how to reconcile all proposed solutions of this thesis in a homogeneous form.

The detailed specifications for each phase of the framework can be found in their respective chapters. However, this chapter refresh some descriptions and the main operations and goals of federated *practical GWASes*.

7.1 Holistic architecture

Figure 7.1 illustrates the multi-enclave TEE architecture used to support the operations of DyPS, I-GWAS and GenDPR.

The goal of the federation is to produce safe releases of GWASes that are evaluated in a per-round basis. A round represents the moment a collection of genome requests is gathered. The biocenters are responsible for sequencing donors' genome data and to generate genome operations according to the desire of the individual, i.e., willing to participate in a GWAS g or asking for removal. Each biocenter has its unique key that is used to encrypt and sign its local messages/requests before outsourcing. Recall that during the attestation phase, the enclaves authenticity are checked and they mutually agree on their keys. Therefore, only attested enclaves can decrypt data of each other. In addition, all shipped messages are produced inside attested enclaves. As a result, the federation members can trust the inputs/outputs from the other enclaves.

The type of the message sent by the enclaves depends on the current phase of the framework (described in Figure 7.2 between fewer – two enclaves – for

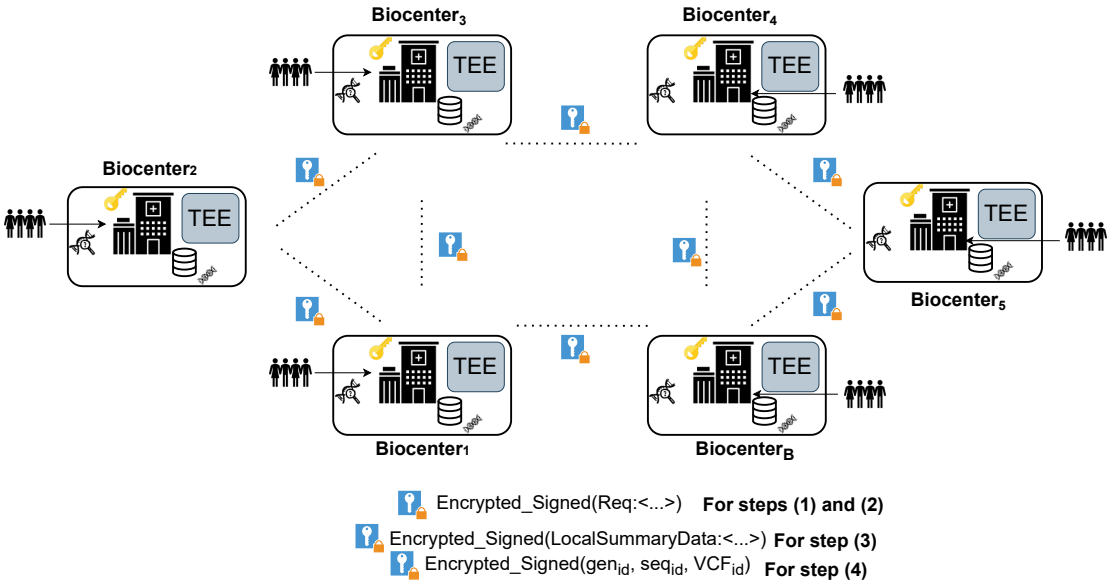


Figure 7.1: Holistic multi-enclave system architecture. Steps (1), (2), (3) and (4) are presented in Figure 7.2

simplicity's sake). The next section describes the pipeline of the holistic framework.

7.2 Holistic framework

For steps (1) and (2), which consist of receiving and analyzing genome operation requests (addition or removal), biocenters only need to send their request <Req> in an encrypted and signed form. <Req> contains $\langle bioc_{id}, gen_{id}, seq_{id}, gwas_g, pop, op\rangle$, previously defined in Section 4.4.1. Recall that seq_{id} (the operation sequence number of the biocenter) is used to analyze the genome operation in a FIFO manner. However, it does not necessarily means that the request is performed in FIFO. For instance, some genomes might not be allowed to participate in a release, while other (even if sent later) might be able to participate depending on the privacy-preserving analysis.

The operations performed in step (2) consists of running DyPS and I-GWAS algorithms to select a safe batch of genomes for a candidate GWAS g release while impeding recovery attacks under the presence of dynamic and overlapping releases. These algorithms measure the theoretical complexity of the solution space of a candidate release in relation to existing (already released studies) to check if a candidate release is within safe boundaries. In other words, if the solution space an internal adversary(ies), colluding biocenters, or an external adversary has to

infer is sufficiently large (recall conditions of Chapters 4 and 5).

Therefore, this step can be executed leveraging only the genome ID (gen_{id}), its respective operation (pop), biocenter ID $bioc_{id}$ and study $gwas_g$ information. Consequently, no actual genome data needs to be shipped.

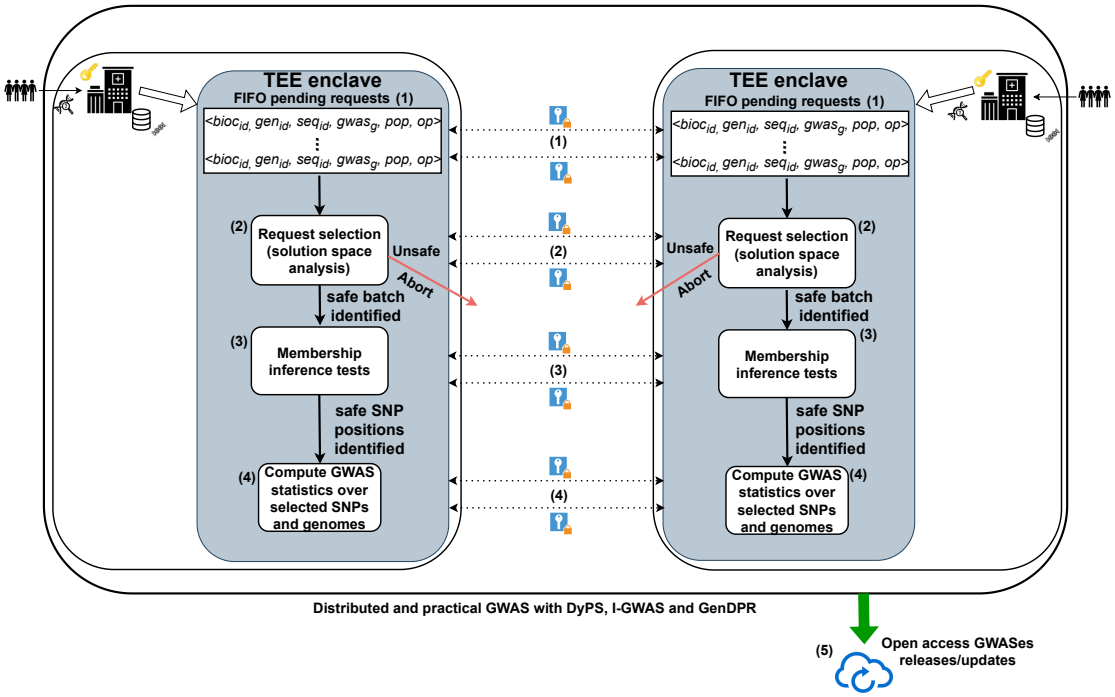


Figure 7.2: Holistic framework for federated practical GWASes.

There are two possibilities after step (2) ends. Either a safe batch of genomes has been found, and then the framework progress to step (3) or the candidate release cannot take place, and therefore it is aborted and evaluated later in a subsequent round. To be noted that pending requests are reassessed in the next rounds. Furthermore, a release of GWAS g will eventually take place as biocenters sequence more genomes over time.

In case step (3) succeeds, the federation needs to assure that the release is not vulnerable against membership inference attacks. For that, the biocenters jointly run the GenDPR algorithm that allows the SecureGenome tests to be performed without genome data shipping. In this step, each biocenter's enclave produce local summary data, e.g., local allele frequencies and local LR-matrix as specified in Section 6.2 for running the distributed LR-test. Such summary data is also encrypted and signed before being broadcast to the other enclaves.

Upon successfully finishing step (3), the federation has identified a safe set of SNP positions that can be used to generate a private GWAS release. Step

(4), the actual GWAS statistics computation, can be performed using any existing privacy-preserving federated GWAS approach or using a similar TEE-based schemed offered in this thesis. However, biocenters might need to share the actual genome data of selected participants ($\texttt{VCF}_{id}$) along with genome ID (gen_{id}) and request sequence ID (seq_{id}) or send local summary data, depending on the assumed approach as discussed in previous chapters.

Finally, the GWAS statistics produced over the data selected in steps (2) and (3) can be publicly published since no potentially colluding federation member or external adversary can compromise genomic privacy of participants given the statistical model protection guarantees described in this work.

Step (5) represents the continuous open-access releases or updates of GWASes statistics over time as soon as more donors are sequenced and their respective genome requests are generated by the biocenters.

7.3 Comparison between SecureGenome, Differential Privacy and the solutions offered in this thesis

The solutions of the thesis rely on statistical-based methods and offer extensions to enable the properties of *practical* GWASes. Due to the importance and popularity of Differential Privacy, this section offers an overall comparison between the statistical inference method chosen (SecureGenome) and Differential Privacy to allow private releases of GWAS statistics. Table 7.1 compares the features, properties, commonalities, advantages, and disadvantages of SG and DP, while relating them to the functionalities enabled by the solutions of this work.

DP is a generic approach to protect data releases against membership inference, and therefore has also been used to protect GWAS releases. DP relies on mathematically proven privacy guarantees (recall DP properties and definition introduced in Section 2.7) to create private statistics releases according to a specified privacy budget used to determine the perturbation level required to protect private data of every single individual. In contrast, SG impedes membership inference attacks by preventing the release of statistics over SNPs that have a low frequency, are highly associated with another SNP (linkage disequilibrium), or whose release would lead to a high identification power of an individual through a LR-test.

In particular, both SG and DP have been used to protect records against membership inference attacks. While DP has a general applicability, SG takes genomic data particularities into account, such as the presence of dependent SNPs and rare allele frequencies. In fact, these two methods have been successfully applied in static GWAS scenarios [Azi+21; Rai+18; Fro+17; San+09b; Zho+11].

Both approaches need a trusted (centralized or distributed) curator responsible for receiving and conducting the analysis for the creation of a private release. In the case of DP, to assess and decide the required perturbation level to protect a release according to the specified privacy budget. For SG, to conduct statistical tests over the data to quantify membership inference risks of individuals according to the specified confidence levels of the LR-test.

DP has the advantage of not requiring a reference set (as SG does) to generate private GWAS releases. Besides, DP does not make any assumption on the background knowledge of adversaries and its cost is cheap once the privacy budget is calibrated (in case of static setting). A clear difference among the two approaches is that while DP produces safe statistics over the entire dataset (but applying data perturbation on the results), SG release statistics only over the SNP positions identified as safe after the verification (i.e., they would keep membership risks of participants below a specified power), but without noise addition.

Unfortunately, despite DP benefits, Liu et al. [LCM16] have identified that the presence of correlated data within a dataset can be exploited by adversaries to breach DP guarantees. Similarly, Almadhoun et al. [AAU20a; AAU20b] recently showed that this issue also impacts genomic privacy. In particular, they demonstrated that inference attacks might become possible when adversaries leverage dependencies (e.g., statistically linked genomic variations or kinship) among the genomes in a study. In contrast, SG's LR-test can be adapted to cope with the presence of relatives of individuals in a study, which can be determined by setting a γ variable to determine the probability of identifying relatives in a cohort [San+09b].

Furthermore, managing the privacy budget of DP mechanisms for protecting continuous observation [Dwo+10; CSS11] and growing databases [Cum+18] are at an early development stage, and the author is unaware of a practical implementation of these concepts. Equivalently, SG also suffers from lack of support for dynamic releases and overlapping data. Indeed, our findings show that new privacy issues arise when considering a dynamic setting, where data is released (queried) multiple times and the presence of overlapping studies is possible. Indeed, these novel privacy issues impact the privacy guarantees of both SG and DP.

For instance, the presence of overlapping data among several studies and the observation of how statistics evolve across releases facilitate adversaries work when mounting genomic privacy attacks. Hence, how to practically (in terms of privacy guarantees and accuracy loss) use DP under dynamic settings is still a challenge. In contrast, the present thesis tackles this challenge building on methods of statistical nature, by providing extensions to enable safe releases of dynamic and potentially overlapping GWASes, whose results are continuously updated as additional genomes become available or are removed, while allowing individuals to remove their consent of participation in ongoing studies.

Table 7.1: Comparison between SecureGenome, Differential Privacy and this work solutions for practical GWAS.

Approach	Description and guarantees	Requires background knowledge	Requires trusted curator	Result perturbation	Release coverage	Privacy budget	Can cope with dependent records	Allows consent withdraw	Dynamic and Interdependent releases	Collusion-tolerance
SecureGenome	Genome-oriented privacy protection mechanism. SG quantifies the membership inference risks of individuals in a study and withhold the release of GWAS statistics over SNP positions that would allow re-identification of participants at a specified identification power rate threshold $1-\beta$.	Yes. SG uses a reference set as the adversary background knowledge during the statistical verification.	Yes.	No.	SG releases statistics over partial data. Only selected SNP positions have GWAS statistics released.	No. The selected data depends on the statistical confidence levels specified for the test.	Yes. SG can support the presence of individuals' relatives in the population.	No.	No. SG cannot produce private releases over continuous and potentially overlapping GWASes releases.	No.
Differential Privacy	Generic approach to provide differentially private releases. DP ensures the privacy of every single individual in a study according to a specified privacy protection level (ϵ).	No.	Yes.	Yes. The perturbation applied to DP releases depends on the value of ϵ.	DP releases statistics over the full data.	Yes. DP has a limited privacy budget that exhausts in a per-release basis (i.e., each release expends a fraction of ϵ).	No. Previous works have shown that DP properties cannot be fully guaranteed under the presence of dependent records.	No.	Limited. Number of releases depends on the specified ϵ and remaining *pvb*. Besides, it is yet not clear if DP guarantees can be kept under the presence of overlapping releases (similarly to the dependent records case).	No.
This thesis solutions	Extends statistical-based inference methods and SG for private releases of dynamic and overlapping GWAS.	Yes. Need a reference set for the SG's LR-test.	Partially. DyPS and I-GWAS need centralized data. However, GenDPR offers a workflow to conduct SG's LR-test analysis in a distributed manner.	No.	Partially. Similar to SG.	No.	Yes. Similar to SG.	Yes.	Yes.	Yes.

7.4 Assessing the limitations of the proposed solutions

This section provides further discussions on the potential limitations of the offered solutions and directions for future research directions.

As covered before, there are two valid approaches to prohibit membership inference attacks on GWAS releases, namely (i) leveraging Differential Privacy or (ii) utilizing statistical inference methods. Each one of these methods has its own particularities and privacy guarantees. In fact, they should be used according to the expectations of the federation in terms of precision of the results, foreseen number of releases and privacy-protection guarantees.

Unfortunately, due to the unavailability of DP-based mechanisms for continuous dynamic releases of GWASes, giving up accuracy (to the scale as presented in Table 5.2) for privacy when using standard DP, might not be reasonable, especially when dealing with high-precision studies, such as GWAS [SB16; SBS19].

The solutions of the present thesis build on SecureGenome [San+09b] to cope with dynamic and overlapping releases given its genome-oriented nature, for being used in previous TEE-based privacy-preserving architectures [Pas+21], and because it is a noise-free approach. Indeed, the offered solutions can afford an infinite number of releases thanks to its exhaustive verification scheme, which is not the case when using DP because limited privacy budget. However, the utility score might be impacted depending on the number of vulnerable SNPs found in the original SNP-set. In addition, the solutions needs additional genomes to protect releases against recovery attacks, which depends on how large are overlapping regions among studies. At the same time, the experiments indicate that the solutions are scalable and has linear complexity depending on the number of existing overlapping studies. Notwithstanding, it is important noticing that the exhaustive verification method conducted by the solutions of this thesis is computationally more expensive than DP-based algorithms.

Regrettably, statistical inference-based methods and DP might be vulnerable to the presence of dependent records [AAU20b; AAU20a; Dwo+10; Cum+18]. Despite that, Sankararaman et al. [San+09b] demonstrated that SG can remediate such an issue by allowing the detection of relatives in a cohort using a parameter (γ) to represent the probability of detecting relatives of individuals in a cohort. As a consequence, the solutions offered in this work can accommodate such a feature. Furthermore, the author is not ware of existing approaches that offer collusion-tolerance in terms of private releasing. To the best of my knowledge, the present thesis is the first to consider the presence of colluding federation members trying to facilitate or circumvent GWAS private release conditions.

DyPS, I-GWAS and GenDPR consider contemporary genomic privacy risks

derived by known existing attacks and their countermeasures. Therefore, our approaches might be vulnerable to future genomic discoveries, e.g., new correlations between SNPs found at a later time. Nevertheless, these type of findings are becoming more unusual with time [Cog+15] and the offered privacy-protection algorithms can be adapted to changing control population statistics but only a posteriori. In particular, even though it is true is that such discoveries might compromise previous releases conducted with our solutions, new findings can be easily integrated to their protocols, e.g., by adding an additional verification to remove specific new dependencies. As a result, past studies releases would have to be made inaccessible or imposed tougher access restrictions as previously conducted by NIH [ZN08; Hom+08]. After potential updates on the protocol, studies can be relaunched so that releases are now conducted following up-to-date genomic privacy protection standards. Lastly, we note that the ability to update releases depends on the current number of overlapping studies and how high their data are correlated, which can impact the release coverage of studies, as verified in our experiments.

In addition, as this thesis extends the Zhou et al.'s [Zho+11] conditions to protect releases against recovery attacks, which considers SNP correlations up to the level of linkage disequilibrium and pairwise correlations as several others [Wan+09; Tra+15; Hum+14]. Several works studies the privacy risks when considering additional genomic-related correlations under different threat models such high-order correlations [KFZ08; VAC19; Dez+17; Sam+15] and kinship [AH17; Dez+17; Hum+17]. Recent research on genome-wide LD identified that: (i) using k^{th} Markov Chain Models (MCM) to identify higher-order correlations might not scale, whereas leveraging recombination models is linear with the number of SNPs [Dez+17]; (ii) inference power of an attack does not increase much when considering MCM with $k > 3$ [Sam+15]; (iii) relying on a Hidden Markov Model (HMM) presents better accuracy than MCM [Dez+17; Sam+15]; and (iv) analysis of how the assumed windows size, number of SNPs in a window, and using sliding-window schemes interfere with the identification of higher-order LD regions [KFZ08; Kem+15]. Besides, it is still unclear how common and powerful high-order LDs are in the genome [ZFC18]. Therefore, our solutions can be extended to remove SNPs that are involved in k^{th} higher-order correlations or SNPs that could be inferred using HMM. However, future work would be required to determine the minimum order of correlations to consider to prevent privacy attacks under dynamic and overlapping settings.

Besides, this thesis envisions that genomic-oriented privacy protection methods can be combined with DP mechanisms to provide better release utility. In particular, such a hybrid mechanism would conduct presented methods as a first step to select a safe batch of genomes and then detect vulnerable SNPs. Currently,

these SNPs are secluded from the release, i.e., they do not have their statistics released. Nevertheless, these SNPs might be released using further protection that can be enforced with Differential Privacy, while the statistics of the remaining (safe) SNPs can be released without perturbation. As a consequence, increasing the overall utility score of releases. However, the limited privacy budget of DP would impede the continuous releases of the SNPs that were released leveraging DP. Therefore, such a study to offer a practical hybrid mechanism is left for future work.

Finally, even though the experiments evaluated in the present thesis uses the suggested privacy parameters adopted in previous works [San+09b; San+09b; AH17], the offered solutions can also support stricter privacy-protection levels, for example, by specifying more conservative thresholds and confidence levels (false-positive and detection rates of the LR-test).

Chapter 8

Conclusions and Future Work

8.1 Conclusions and outcomes of the thesis

To the best of author's knowledge, this thesis is the first that reconciles secure genomic data analysis (in the sense of securely and in a privacy-preserving manner conducting federated GWAS), with privacy-preserving open releasing of GWAS statistics results. In addition, the present thesis enables several novel functionalities toward *practical GWAS*, which requires novel mechanisms to support additional features. This thesis therefore, identified and provided solutions to enable them.

In particular, it shows that on one hand, an adversary might launch privacy attacks during the computation of GWASes. On the other hand, GWASes releases must preserve genomic privacy against known attacks in the literature, i.e., recovery and membership attacks. For instance, these attacks raise genome privacy concerns due to the fact that genome data is very personal and therefore should remain private. Indeed, genomic data contains sensitive data that would allow, for instance, individuals' predisposition to diseases, and even be used to leak sensitive data from individuals' family members or relatives. As a result, nowadays, GWASes results are not allowed to be publicly shared anymore, which impacts the spreading of the benefits brought by GWASes, such as early disease detection, drug developments, and personalized medicine

Unfortunately, existing state-of-the-art solutions are not able to combine both aspects of a fully privacy-preserving federated GWAS (processing and releasing) environment in a homogenized form. Moreover, given the reduced cost of sequencing DNA nowadays, and also to allow federations to comply with data-privacy regulations, such as GDPR (that oblige institutions, e.g., biocenters, to enable consent withdrawal from genome data donors at any time), there is now a need to update GWASes results in a dynamic manner. As a consequence, enabling dynamic update of GWASes brings novel genomic privacy challenges. For instance,

adversaries might observe several GWASes updates, and compare how statistics have evolved from time to time. As shown in this thesis, such an observation can compromise private genomic data of participants in new ways.

Additionally, due to economic reasons or to surpass competitors, for example, some biocenters might conjointly exchange data in order to circumvent safe conditions of releases and so infringe private data of other biocenters. Indeed, HbC biocenters can collude aiming to isolate sufficiently enough data of other's biocenters that will become vulnerable to genomic privacy attacks. Furthermore, the existence of overlapping genome data being used by multiple studies is also an open issue. Indeed, a number of the same genomes and/or SNP positions might be used in different studies. Such a scenario might decrease the solution space adversaries have to infer when launching recovery attacks or enable successful membership inference by the combination of multiple overlapping GWASes' data.

To cope with such issues, this thesis offers solutions combined in form of a unique framework capable of enabling dynamic privacy-preserving releases of overlapping GWASes results according to the evolution of the considered number of genomic variations, individuals, or involved biocenters while enforcing privacy. The solution leverages Intel SGX to securely receive and process genome data from biocenters whilst evaluating safe release conditions to allow safe publication of GWASes results without compromising privacy of both donors and data holders. It proposes a genome operation (i.e., addition and removal of participation in studies) requests selection mechanism that selects a safe batch of genomes even when up to $f = B - 1$ biocenters are colluding to attack others. Furthermore, it provides a scaling mechanism to speed up the release of results by progressively increasing the number of considered genomic variations. Moreover, the thesis is the first to acknowledges the new privacy issues that arise under the presence of interdependent GWASes, which might share genomes and SNPs. As identified, this scenario obliges new safe release conditions to be enforced.

The performance of the solutions were extensively evaluated considering up to 300,000 SNPs and more than 6 million simulated genomes, and using two datasets made of 2,000 and 35,000 real genomes, respectively. It is experimentally shown and theoretically proved that the solution can scale with the number of studies and genomes while enabling dynamic releases. It was successfully demonstrated the practicability of both test and aggregate statistics release mechanisms, and compared them against baseline approaches. For test statistics, the thesis compared the solution to an approach that would immediately update the results whenever a sufficiently large batch of requests could be assembled. This latter approach puts up to 8% of the genomes at risk, while this work prevents all privacy leaks. For aggregate statistics, the solution was compared to a static approach that would wait for all requests to have been received before releasing statistics. The exper-

iments show that this work is able to provide earlier aggregate statistics releases over the same number of SNPs. These earlier releases occasionally provided statistics including up to 2.6 times as many SNPs. Moreover, the existence of colluding players only slightly impacts the performance of the solutions by increasing the time that some genome requests are executed.

Compared to safe releases that consider only a single, but possibly dynamically updated GWAS, up to 28.6% of the processed genomic information could be leaked in recovery attacks. Depending on the number of genomes shared in a study, between 81.8% and 92.3% of the SNPs remain vulnerable to membership attacks if disclosed naïvely. It is also demonstrated the benefits of using our noise-free solution instead of relying on DP mechanisms. Thus, achieving no accuracy loss when releasing GWASes. Indeed, when compared to standard ϵ-DP using the Laplace mechanism, DP considerably decreases the accuracy of releases. On the other hand, this thesis' solution only degrades the number of SNPs having statistics releases, but without noise addition. Although it is true that this method decreases the coverage of GWAS releases, it is shown that depending on the settings (expected number of releases, privacy guarantee levels), **I-GWAS** presents a better release utility score that DP-based mechanisms. Notwithstanding, the author is unaware of DP-based solutions that can enable safe dynamic releases of GWAS as endorsed by the approach offered in this work.

Chapter 6, which introduces **GenDPR**, extends the previous solutions to show that the privacy-protection mechanisms enforced by them can also be conducted in a distributed manner. **GenDPR** is a distributed multi-enclave federated framework that allows members of the federation to jointly assess and decide private releases of GWAS, without requiring genome data outsourcing kept at centralized local (as assumed by centralized federated schemes). **GenDPR** accomplishes these goals by enforcing coordination and aggregation tasks performed by an randomly elected trusted leader that receives only intermediate data (e.g., summary statistics) from other members. Thus, federation members genome data does not leave their local premises. **GenDPR** proves that privacy-protection mechanisms (such as to protect releases against membership attacks) can be correctly computed in a distributed fashion and can also cope with the existence of colluding participants. In particular, **GenDPR** is able to generate the same output of the SecureGenome analysis over centralized data. Moreover, **GenDPR** can identifies additional data that would be vulnerable with the presence of colluding attacks. **GenDPR** also presents an efficient performance in terms of computational resources, such as memory and CPU consumption but also in running times. Indeed, as some operations of the privacy-protection verifications are shared among the members, for some settings of federations, the running time of **GenDPR** is smaller than the centralized approach where all computations are to be performed by a single

machine.

As a consequence, the present thesis addresses crucial and recent genomic privacy concerns that will allow further scientific progress in genomics research by enabling secure and privacy-preserving genome data sharing and collaboration. In particular, the proposed solutions offer more trust and privacy to not only people donating their genomic data but also to data holders (e.g., biocenters). Thus, facilitating the creation and adoption of larger-scale GWASes. As a result, accelerating the development of new treatments and advances in medicine. In summary, this thesis advances the literature on secure and privacy-preserving GWAS under collaborative environments by offering an end-to-end privacy-preserving environment for federated GWAS while enabling new features to enable *practical* GWAS that consider 21st-century privacy guidelines.

8.2 Future work

Future work includes extending our solutions to tolerate byzantine biocenters (which might behave arbitrarily) inside the federation. These malicious biocenters could upload fake genomic data, statistics or genome operations requests. To mitigate this, a novel module to check authenticity and genuinity of genome data can be conducted within enclaves so that outsourced data from federation members can be attested. Nevertheless, the creation of algorithms to detect genuine genome data is still a challenge.

Other directions for future work include analyzing the interplay between differential privacy and tolerable privacy budgets for dependent and overlapping records under continual releases. Besides, studying and addressing new privacy issues that might raise under the presence of colluding parties in local-DP based schemes, similarly to what have been identified in this thesis with statistical-based methods.

On the system-side aspect of the work, it is planned to create an data-oblivious version of the protocol and algorithms in order to mitigate the side-channel attack vulnerability of TEE-based schemes.

In addition, I plan to study the feasibility of employing fault-tolerant protocols on top of the multi-enclave privacy-protecting protocol so that our solutions can make progress and keep liveness even given the presence of faulty (e.g., delayed or disrupted members).

Furthermore, I aim at designing a hybrid approach that combines homomorphic encryption (HE) and TEE. For instance, leveraging HE for simpler tasks, such as data aggregation, while relying on TEEs to compute more complex operations, such as running conducting membership inference tests.

Additionally, this work leaves as an open challenge the designing of collusion-tolerant by nature privacy-preserving *releasing* mechanisms. For instance, a collusion-

tolerant Differential Privacy approach to support the presence of colluding members sharing their raw and perturbed data with specific members trying to circumvent private release conditions, similar to what has been assumed in this thesis.

Last but not least, after a fruitful discussion with Prof. Dr. Yves Moreau, I leave as another remaining challenge the creation of privacy-preserving releasing schemes that account for the "meaningfulness" of the SNP positions allowed to have their statistics disclosed. In particular, privacy-preserving releasing approaches might prohibit the participation of genomes/SNPs with a high association with a study, but cannot have their statistics released due to the privacy-protection mechanisms in place. Hence, implementing a method that acknowledges the interplay between privacy and SNP associations within a study is a crucial feature to allow easier adoption of such privacy-preserving schemes by bioinformaticians and researchers.

Terminology

Concept	Description
Deoxyribonucleic acid (DNA)	Elongated polymer composed of genetic information that form all living beings' genetic code.
Gene	The sections of a DNA responsible for accommodating instructions for specific functional molecules, usually a protein. It encodes and express individual characteristics (e.g., hair color) of all living creatures, which is inherited from relatives.
Genome	The complete set of genes (genetic instructions) needed to form all functional characteristics of an organism allowing it to live, grow and develop. Genomes are found in specific DNAS' partitions (chromosomes). individual. It contains all the information the individual requires to function. The genome is usually stored in long molecules of DNA called chromosomes.
Chromosome	A long DNA molecule that stores the genetic material of a creature. It is a threadlike structure wrapped around proteins in a scaffolding manner. The human genome is made of 23 pairs of chromosomes.
Nucleotide	An organic molecule that is composed of sugar, a phosphate group, and a nitrogenous base. It represents the basic unit that assembles the DNA. It is categorized depending on its nitrogenous type. In DNAs, there are four types of nucleotides: adenine (A), cytosine (C), thymine (T), and guanine (G). While for RNAs, uracil (U) substitutes thymine.
Locus	In the biological context, it represents a single region in the genome where genetic variations (the presence of different nucleotides in creatures of a same population) might occur.

Concept	Description
Reference genome	It is a generic genome that contains all common variations found in a population. The human reference genome consists of all known variations present in humans. On average, humans share 99.9% of genetic information from which the remaining 0.1% consists of variations.
Single nucleotide polymorphism (SNP)	The smallest piece of genetic information (i.e., a nucleotide) that varies at a specific position (locus) of the genome in a population. Only variations that are present in more than 1% of the population are considered SNPs. Variations found in less than 1% of the population are recognized as rare mutations or abnormal changes.
Allele	A variant of a nucleotide stored in a chromosome pair. Each cell in a pair of chromosomes has two alleles inherited from each parent per gene. There is a dominant allele, which is the one responsible for coding the functionality of a protein, whereas the recessive allele does not contribute to the encoding of proteins. Furthermore, there are two types of alleles: major and minor alleles according to their incidence at a specific locus of the genome.
Major allele	The most common type of allele found at a given locus of the genome in a population.
Minor allele	The rarest type of allele found at a given locus of the genome in a population.
Allele frequency	It measures the prevalence of genomic variations among individuals. It consists of the sum of all incidences of a given allele at a specific locus, divided by the total number of alleles found in a population.

Concept	Description
Genotype	The full sequence (double-stranded DNA) of alleles present in individuals' SNPs. Since any two genotypes of a person (i.e., a complete set of genes) are mostly identical, one can represent an individual's genotype by the difference of its information compared to a human reference genome. There are three types of genotypes: (i) major homozygous genotype, i.e., the presence of two major alleles; (ii) minor homozygous genotype, i.e., the presence of two minor alleles; and (iii) heterozygous genotype, i.e., the presence of a major and a minor allele.
Haplotype	A single strand sequence of DNA variations (e.g., SNPs) adjacent to each other.
Phenotype	The observable physical properties or characteristic present in a organism. It results from the combination of the environment where the living creature lives and its genotype.
Genome-wide association study (GWAS)	An observational and statistical study that aims at identifying correlations between specific genome variations (SNPs) with a particular phenotype, e.g., a disease. It evaluates the presence of certain genomic variations among two groups of individuals (case and control) that might be linked to the concerned trait.
Federated analysis	A collaborative environment where several parties (that also might do not trust each other, for economical reasons, for example), jointly agrees to contribute by sharing their local data/inputs aiming at computing a given function of interest, e.g. performing GWAS statistics using aggregated data from all participants of the federation, i.e., data holders.
Case population	Individuals participating in a GWAS that does not express the phenotype of interest, e.g., lung cancer.
Control population	Individuals participating in a GWAS that does express the phenotype of interest, e.g., people with lung cancer.

Concept	Description
Minor Allele Frequency (MAF)	It is the frequency of the second most common allele (i.e., a minor allele) of a SNP locus in a population. Usually, GWAS aims at identifying MAF between 1%-5%.
chi-square (x^2) statistic	An association test that determines if there is a statistical correlation between variables. In GWASes, it is used to determine whether or not the null hypothesis, which states that the allele frequencies in the case and control populations follow a similar distribution, can be rejected.
p-value (x^2) statistic	A statistical test to determine the probability of occurrence of the observed data seems consistent or not with the null hypothesis of the *chi*-square test. Usually, a p-value below 10^{-8}) indicates that a genetic variant is highly associated with the phenotype of interest of the GWAS.
Linkage disequilibrium (LD)	The phenomenon when allele frequencies at one or more loci are not independent, i.e., when the frequency of association between alleles at different loci is not random.
Likelihood-ratio test (LR-test)	A LR-test measures the goodness of fit of two competing statistical models (i.e., the null and alternative hypothesis) based on the ratio of their likelihoods observed from supporting data (e.g., case and control populations of a GWAS).
Trusted execution environment (TEE)	It consists of a secured area inside a CPU's processor where loaded code and data cannot be tampered with, even by the operating system. Such isolation enforces confidentiality and integrity of information residing in such regions.
Homomorphic Encryption (HE)	It is a cryptographic method that allows the execution of computation (e.g., arithmetic operations) over encrypted data.

Concept	Description
Secure Multiparty Computation (SMC)	It is a cryptographic method that allows several parties to jointly operate, compute and receive the result of a function of interest without disclosing their private inputs to other parties.
Differential Privacy (DP)	It is an approach for allowing the public sharing of information (e.g., statistical results) about a given dataset while preventing the leak of private information of any record part of the dataset.
Honest-but-Curious (HbC) adversary	In a multiparty setting, a HbC adversary is an entity that follows the protocol honestly but might attempt to learn extra information from the system or other parties.
Probabilistic polynomial-time (p.p.t.) adversary	It is an adversary that can only perform a polynomial amount of operations that can be accomplished by any probabilistic polynomial-time algorithm. Such an algorithm can rely on the results of a random source, e.g., a random function such as tossing a coin.

References

[AAC21] Kerem Ayoz, Erman Ayday, and A Ercument Cicek. "Genome Reconstruction Attacks Against Genomic Data-Sharing Beacons". In: *Proceedings on Privacy Enhancing Technologies* 3 (2021), pp. 28–48.

[AAU20a] Nour Almadhoun, Erman Ayday, and Özgür Ulusoy. "Differential privacy under dependent tuples—the case of genomic privacy". In: *Bioinformatics* 36.6 (2020), pp. 1696–1703.

[AAU20b] Nour Almadhoun, Erman Ayday, and Özgür Ulusoy. "Inference attacks against differentially private query results from genomic datasets including dependent tuples". In: *Bioinformatics* 36.Supplement_1 (2020), pp. i136–i145.

[AH17] Erman Ayday and Mathias Humbert. "Inference attacks against kin genomic privacy". In: *IEEE Security & Privacy* 15.5 (2017), pp. 29–37.

[AKM20] Aref Asvadishirehjini, Murat Kantarcioglu, and Bradley Malin. "A Framework for Privacy-Preserving Genomic Data Analysis Using Trusted Execution Environments". In: *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE. 2020, pp. 138–147.

[Al +17] Md Momin Al Aziz, Reza Ghasemi, Md Waliullah, and Noman Mohammed. "Aftermath of bustamante attack on genomic beacon service". In: *BMC medical genomics* 10.2 (2017), pp. 43–54.

[Ayo+20] Kerem Ayoz, Miray Aysen, Erman Ayday, and A Ercument Cicek. "The effect of kinship in re-identification attacks against genomic data sharing beacons". In: *Bioinformatics* 36.Supplement_2 (2020), pp. i903–i910.

[Aze18] C-A Azencott. "Machine learning and genomics: precision medicine versus patient privacy". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376.2128 (2018), p. 20170350.

[Azi+21] Md Momin Al Aziz, Shahin Kamali, Noman Mohammed, and Xiaoqian Jiang. "Online Algorithm for Differentially Private Genome-wide Association Studies". In: *ACM Transactions on Computing for Healthcare* 2.2 (2021), pp. 1–27.

[Bac+18] Michael Backes, Pascal Berrang, Mathias Humbert, Xiaoyu Shen, and Verena Wolf. "Simulating the large-scale erosion of genomic privacy over time". In: *IEEE/ACM transactions on computational biology and bioinformatics* 15.5 (2018), pp. 1405–1412.

[Bal+11] Pierre Baldi, Roberta Baronio, Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. "Countering gattaca: efficient and secure testing of fully-sequenced human genomes". In: *Proceedings of the 18th ACM conference on Computer and communications security*. 2011, pp. 691–702.

[Bar+12] Gregory S Barsh, Gregory P Copenhaver, Greg Gibson, and Scott M Williams. "Guidelines for genome-wide association studies". In: *PLoS Genet* 8.7 (2012), e1002812.

[BAZ20] Abubakar Bomai, Mohammed Shujaa Aldeen, and Chuan Zhao. "Privacy-Preserving GWAS Computation on Outsourced Data Encrypted under Multiple Keys Through Hybrid System". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2020, pp. 683–691.

[Ber+18] Pascal Berrang, Mathias Humbert, Yang Zhang, Irina Lehmann, Roland Eils, and Michael Backes. "Dissecting privacy risks in biomedical data". In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2018, pp. 62–76.

[Bes+15] Alysson Bessani, Jörgen Brandt, Marc Bux, Vinicius Cogo, Lora Dimitrova, Jim Dowling, Ali Gholami, Kamal Hakimzadeh, Micheal Hummel, Mahmoud Ismail, et al. "BiobankCloud: a platform for the secure storage, sharing, and processing of large biomedical data sets". In: *Workshop on Data Management and Analytics for Medicine and Healthcare*. 2015.

[BG17] Forrest Briscoe and Barbara Gray. "INNOVATIONS IN MEDICAL GENOMICS: HOW TO ENABLE ADVANCES WHILE MANAGING PRIVACY AND SECURITY RISKS?" In: (2017).

[Bla+18] Marina Blanton, Ah Reum Kang, Subhadeep Karan, and Jaroslaw Zola. "Privacy Preserving Analytics on Distributed Medical Data". In: *arXiv preprint arXiv:1806.06477* (2018).

[Bla+20] Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, and Shafi Goldwasser. "Secure large-scale genome-wide association studies using homomorphic encryption". In: *National Academy of Sciences* 117.21 (2020), pp. 11608–11613.

[Blo] Gene Blockchain. *Gene Blockchain English White paper*. https://www.geneblockchain.org/download-view/whitepaper_in_english/. Accessed on: January 10th, 2022.

[BLR19] Cesare Bartolini, Gabriele Lenzini, and Livio Robaldo. "The Security Implications of Data Subject Rights". In: *IEEE SECURITY & PRIVACY* 17.6 (2019), pp. 37–45.

[BLW08] Dan Bogdanov, Sven Laur, and Jan Willemson. "Sharemind: A framework for fast privacy-preserving computations". In: *European Symposium on Research in Computer Security*. Springer. 2008, pp. 192–206.

[BM12] William S Bush and Jason H Moore. "Genome-wide association studies". In: *PLoS computational biology* 8.12 (2012), e1002822.

[Bog+14] Dan Bogdanov, Liina Kamm, Sven Laur, Pille Pruulmann-Vengerfeldt, Riivo Talviste, and Jan Willemson. "Privacy-preserving statistical data analysis on federated databases". In: *Annual Privacy Forum*. Springer. 2014, pp. 30–55.

[Bog+18] Dan Bogdanov, Liina Kamm, Sven Laur, and Ville Sokk. "Implementation and evaluation of an algorithm for cryptographically private principal component analysis on genomic data". In: *IEEE/ACM transactions on computational biology and bioinformatics* 15.5 (2018), pp. 1427–1432.

[Bon+18] Charlotte Bonte, Eleftheria Makri, Amin Ardeshirdavani, Jaak Simm, Yves Moreau, and Frederik Vercauteren. "Towards practical privacy-preserving genome-wide association study". In: *BMC bioinformatics* 19.1 (2018), p. 537.

[Bra+17] Ferdinand Brasser, Urs Müller, Alexandra Dmitrienko, Kari Kostiainen, Srdjan Capkun, and Ahmad-Reza Sadeghi. "Software Grand Exposure:{SGX} Cache Attacks Are Practical". In: *WOOT*. 2017.

[BSB20] Tim Beck, Tom Shorter, and Anthony J Brookes. "GWAS Central: a comprehensive resource for the discovery and comparison of genotype and phenotype data from genome-wide association studies". In: *Nucleic acids research* 48.D1 (2020), pp. D933–D940.

[Byc+18] Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, et al. "The UK Biobank resource with deep phenotyping and genomic data". In: *Nature* 562.7726 (2018), pp. 203–209.

[Cai+15] Ruichu Cai, Zhifeng Hao, Marianne Winslett, Xiaokui Xiao, Yin Yang, Zhenjie Zhang, and Shuigeng Zhou. "Deterministic identification of specific individuals from GWAS results". In: *Bioinformatics* 31.11 (2015), pp. 1701–1707.

[CD16] Victor Costan and Srinivas Devadas. "Intel SGX Explained". In: *IACR Cryptology ePrint Archive* 2016.086 (2016), pp. 1–118.

[Che+16a] Feng Chen, Michelle Dow, Sijie Ding, Yao Lu, Xiaoqian Jiang, Hua Tang, and Shuang Wang. "PREMIX: Privacy-preserving EstiMation of individual admixture". In: *AMIA Annual Symposium* 2016 (2016), p. 1747.

[Che+16b] 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". In: *Bioinformatics* 33.6 (2016), pp. 871–878.

[Che+17a] Feng Chen, Chenghong Wang, Wenrui Dai, Xiaoqian Jiang, Noman Mohammed, Md Momin Al Aziz, Md Nazmus Sadat, Cenk Sahinalp, Kristin Lauter, and Shuang Wang. "PRESAGE: Privacy-preserving genetic testing via software guard extension". In: *BMC medical genomics* 10.2 (2017), p. 48.

[Che+17b] Wang Chenghong, Yichen Jiang, Noman Mohammed, Feng Chen, Xiaoqian Jiang, Md Momin Al Aziz, Md Nazmus Sadat, and Shuang Wang. "SCOTCH: Secure Counting Of encrypTed genomiC data using a Hybrid approach". In: *AMIA Annual Symposium Proceedings*. Vol. 2017. American Medical Informatics Association. 2017, p. 1744.

[Che+21] Zhongsheng Chen, Michael Boehnke, Xiaoquan Wen, and Bhramar Mukherjee. "Revisiting the genome-wide significance threshold for common variant GWAS". In: *G3* 11.2 (2021), jkaa056.

[CKM12] Mustafa Canim, Murat Kantarcioglu, and Bradley Malin. "Secure Management of Biomedical Data With Cryptographic Hardware". In: *IEEE transactions on information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society* 16.1 (2012), pp. 166–175.

[Cla10] David Clayton. "On inferring presence of an individual in a mixture: a Bayesian approach". In: *Biostatistics* 11.4 (2010), pp. 661–673.

[Cog+15] Vinicius V Cogo, Alysson Bessani, Francisco M Couto, and Paulo Verissimo. "A high-throughput method to detect privacy-sensitive human genomic data". In: *14th ACM Workshop on Privacy in the Electronic Society*. ACM. 2015, pp. 101–110.

[Con+15a] 1000 Genomes Project Consortium et al. "A global reference for human genetic variation". In: *Nature* 526.7571 (2015), p. 68.

[Con+15b] Scott D Constable, Yuzhe Tang, Shuang Wang, Xiaoqian Jiang, and Steve Chapin. "Privacy-preserving GWAS analysis on federated genomic datasets". In: *BMC medical informatics and decision making* 15.5 (2015), S2.

[Cor+18] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. "Privacy at scale: Local differential privacy in practice". In: *Proceedings of the 2018 International Conference on Management of Data*. 2018, pp. 1655–1658.

[Cou08] Jennifer Couzin. "Whole-genome data not anonymous, challenging assumptions". In: *Science* (2008).

[Cra+11] David W Craig, Robert Goor, Zhenyan Wang, Justin Paschall, Jim Ostell, Mike Feolo, Stephen T Sherry, and Teri A Manolio. "Assessing and managing risk when sharing aggregate genetic variant data". In: *Nature reviews Genetics* 12.10 (2011), p. 730.

[CSS11] T-H Hubert Chan, Elaine Shi, and Dawn Song. "Private and continual release of statistics". In: *TISSEC* 14.3 (2011), pp. 1–24.

[CT18] Sergiu Carpov and Thibaud Tortech. "Secure top most significant genome variants search: iDASH 2017 competition". In: *BMC medical genomics* 11.4 (2018), p. 82.

[Cum+18] Rachel Cummings, Sara Krehbiel, Kevin A Lai, and Uthaipon Tantipongpipat. "Differential privacy for growing databases". In: *Advances in Neural Information Processing Systems*. 2018.

[CWB18] Hyunghoon Cho, David J Wu, and Bonnie Berger. "Secure genome-wide association analysis using multiparty computation". In: *Nature biotechnology* 36.6 (2018), p. 547.

[Dan+20] Fida K Dankar, Marton Gergely, Bradley Malin, Radja Badji, Samar K Dankar, and Khaled Shuaib. "Dynamic-informed consent: A potential solution for ethical dilemmas in population sequencing initiatives". In: *Computational and Structural Biotechnology Journal* (2020).

[Dec+18] Jérémie Decouchant, Maria Fernandes, Marcus Völp, Francisco M Couto, and Paulo Esteves-Verissimo. "Accurate filtering of privacy-sensitive information in raw genomic data". In: *Journal of biomedical informatics* 82 (2018), pp. 1–12.

[Den+20] Yamin Deng, Tao He, Ruiling Fang, Shaoyu Li, Hongyan Cao, and Yuehua Cui. "Genome-Wide Gene-Based Multi-Trait Analysis". In: *Frontiers in Genetics* 11 (2020), p. 437.

[Der+22] Leonard Dervishi, Xinyue Wang, Wentao Li, Anisa Halimi, Jaideep Vaidya, Xiaoqian Jiang, and Erman Ayday. "Facilitating Federated Genomic Data Analysis by Identifying Record Correlations while Ensuring Privacy". In: *arXiv preprint arXiv:2203.05664* (2022).

[Des+] Giuseppe Desolda, Joseph Aneke, Carmelo Ardito, Rosa Lanzilotti, and Maria Francesca Costabile. "Explanations in Warning Dialogs to Help Users Defend Against Phishing Attacks". In: *Available at SSRN 4127608* ().

[Dez+17] Iman Deznabi, Mohammad Mobayen, Nazanin Jafari, Oznur Tastan, and Erman Ayday. "An inference attack on genomic data using kinship, complex correlations, and phenotype information". In: *IEEE/ACM transactions on computational biology and bioinformatics* 15.4 (2017), pp. 1333–1343.

[DP13] Cynthia Dwork and Rebecca Pottenger. "Toward practicing privacy". In: *Journal of the American Medical Informatics Association* 20.1 (2013), pp. 102–108.

[DR+14] Cynthia Dwork, Aaron Roth, et al. "The algorithmic foundations of differential privacy". In: *Found. Trends Theor. Comput. Sci.* 9.3-4 (2014), pp. 211–407.

[DSZ15] Daniel Demmler, Thomas Schneider, and Michael Zohner. "ABY-A framework for efficient mixed-protocol secure two-party computation." In: *NDSS*. 2015.

[Dwo+10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. "Differential privacy under continual observation". In: *STOC*. 2010, pp. 715–724.

[Dwo11] C Dwork. *Differential privacy*. Springer, 2011, pp. 338–340.

[Edi08] Editors. "DNA databases shut after identities compromised." In: *Nature* 455 (2008), p. 13.

[Eig+14] Fabienne Eigner, Aniket Kate, Matteo Maffei, Francesca Pampaloni, and Ivan Pryvalov. "Differentially private data aggregation with optimal utility". In: *ACSAC*. 2014.

[Eng16] Genomics England. "The 100,000 genomes project". In: *NHS Genomics England* (2016).

[Fer+17] Maria Fernandes, Jérémie Decouchant, Francisco M Couto, and Paulo Esteves-Verissimo. "Cloud-Assisted Read Alignment and Privacy". In: *International Conference on Practical Applications of Computational Biology & Bioinformatics*. Springer. 2017, pp. 220–227.

[Fer+19] Maria Fernandes, Jérémie Decouchant, Marcus Völp, Francisco M Couto, and Paulo Esteves-Verissimo. "DNA-SeAl: sensitivity levels to optimize the performance of privacy-preserving DNA alignment". In: *IEEE Journal of Biomedical and Health Informatics* 24.3 (2019), pp. 907–915.

[Fre+14] Matthew Fredrikson, 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: *USENIX Security*. 2014.

[Fro+17] 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: *PETS* 2017.4 (2017), pp. 232–250.

[Fro+21] David Froelicher, Juan R Troncoso-Pastoriza, Jean Louis Raisaro, Michel A Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. "Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption". In: *Nature communications* 12.1 (2021), pp. 1–10.

[Gen09] Craig Gentry. "Fully homomorphic encryption using ideal lattices". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 2009, pp. 169–178.

[Gib+03] Richard A Gibbs, John W Belmont, Paul Hardenbol, Thomas D Willis, FL Yu, HM Yang, Lan-Yang Ch'ang, Wei Huang, Bin Liu, Yan Shen, et al. "The international HapMap project". In: (2003).

[Gol87] Oded Goldreich. "Towards a theory of software protection and simulation by oblivious RAMs". In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. 1987, pp. 182–194.

[Goo22] Google. *Google Differential Privacy Library*. https://github.com/google/differential-privacy. Accessed on: January 7th, 2022. 2022.

[Gür+18] Gamze Gürsoy, Arif Harmanci, Haixu Tang, Erman Ayday, and Steven E Brenner. "When biology gets personal: hidden challenges of privacy and ethics in biological big data". In: *BIOCOMPUTING 2019: Proceedings of the Pacific Symposium*. World Scientific. 2018, pp. 386–390.

[Hal+21] Anisa Halimi, Leonard Dervishi, Erman Ayday, Apostolos Pyrgelis, Juan Ramon Troncoso-Pastoriza, Jean-Pierre Hubaux, Xiaoqian Jiang, and Jaideep Vaidya. "Privacy-Preserving and Efficient Verification of the Outcome in Genome-Wide Association Studies". In: *PETS* (2021).

[Has+18] Mohammad Zahidul Hasan, Md Safiur Rahman Mahdi, Md Nazmus Sadat, and Noman Mohammed. "Secure count query on encrypted genomic data". In: *Journal of biomedical informatics* 81 (2018), pp. 41–52.

[He+18] Zaobo He, Jiguo Yu, Ji Li, Qilong Han, Guangchun Luo, and Yingshu Li. "Inference attacks and controls on genotypes and phenotypes for individual genomic data". In: *IEEE/ACM transactions on computational biology and bioinformatics* 17.3 (2018), pp. 930–937.

[Hea] U.S. Department of Health & Human Services. *Health Insurance Portability and Accountability Act (HIPAA)*. https://www.hhs.gov/hipaa/index.html. Accessed on: January 12th, 2022.

[Hee+11] Catherine Heeney, Naomi Hawkins, Jantina de Vries, Paula Boddington, and Jane Kaye. "Assessing the privacy risks of data sharing in genomics". In: *Public health genomics* 14.1 (2011), pp. 17–25.

[Her] Matthew Herper. *Illumina Promises To Sequence Human Genome For $100 – But Not Quite Yet*. Ed. by Forbes. https://www.forbes.com/sites/matthewherper/2017/01/09/illumina-promises-to-sequence-human-genome-for-100-but-not-quite-yet/. Accessed on: January 10th, 2022.

[Hom+08] 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". In: *PLoS genetics* 4.8 (2008).

[HTH19] Mathias Humbert, Benjamin Trubert, and Kévin Huguenin. "A survey on interdependent privacy". In: *ACM Computing Surveys (CSUR)* 52.6 (2019), pp. 1–40.

[Hua+15] Zhicong Huang, Erman Ayday, Jacques Fellay, Jean-Pierre Hubaux, and Ari Juels. "GenoGuard: Protecting genomic data against brute-force attacks". In: *Security & Privacy*. 2015.

[Hum+14] Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. "Reconciling utility with privacy in genomics". In: *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. 2014, pp. 11–20.

[Hum+15] Mathias Humbert, Kévin Huguenin, Joachim Hugonot, Erman Ayday, and J-P Hubaux. "De-anonymizing genomic databases using phenotypic traits". In: *Proceedings on Privacy Enhancing Technologies* 2015.2 (2015).

[Hum+17] Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. "Quantifying interdependent risks in genomic privacy". In: *ACM Transactions on Privacy and Security (TOPS)* 20.1 (2017), pp. 1–31.

[Hum+22] Mathias Humbert, Didier Dupertuis, Mauro Cherubini, and Kévin Huguenin. "KGP Meter: Communicating Kin Genomic Privacy to the Masses". In: *IEEE European Symposium on Security and Privacy (EuroS&P)*. 2022, p. 20.

[IBM21] IBM. *IBM4764*. https://www.ibm.com/docs/en/i/7.1?topic=cryptography-4764-4765-cryptographic-coprocessors. 2021.

[Im+12] Hae Kyung Im, Eric R Gamazon, Dan L Nicolae, and Nancy J Cox. "On sharing quantitative trait GWAS results in an era of multiple-omics data and the limits of genomic privacy". In: *The American Journal of Human Genetics* 90.4 (2012), pp. 591–598.

[Ins] NIH - National Human Genome Research Institute. *The Human Genome Project*. Ed. by NIH. https://www.genome.gov/human-genome-project/. Accessed on: January 12th, 2022.

[Jac+09] 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". In: *Nature genetics* 41.11 (2009), p. 1253.

[Jag+17] Karthik A Jagadeesh, David J Wu, Johannes A Birgmeier, Dan Boneh, and Gill Bejerano. "Deriving genomic diagnoses without revealing patient genomes". In: *Science* 357.6352 (2017), pp. 692–695.

[Jai] Ducharme Jaime. *A Major Drug Company Now Has Access to 23andMe's Genetic Data. Should You Be Concerned?* Ed. by Time. http://time.com/5349896/23andme-glaxo-smith-kline/. Accessed on: February 08th, 2022.

[Jia+14] Xiaoqian Jiang, Yongan Zhao, Xiaofeng Wang, Bradley Malin, Shuang Wang, Lucila Ohno-Machado, and Haixu Tang. "A community assessment of privacy preserving techniques for human genomes". In: *BMC medical informatics and decision making* 14.1 (2014), S1.

[JS13] Aaron Johnson and Vitaly Shmatikov. "Privacy-preserving data exploration in genome-wide association studies". In: *SIGKDD*. 2013, pp. 1079–1087.

[Kam+13] Liina Kamm, Dan Bogdanov, Sven Laur, and Jaak Vilo. "A new way to protect privacy in large-scale genome-wide association studies". In: *Bioinformatics* 29.7 (2013), pp. 886–893.

[Kay12] Jane Kaye. "The tension between data sharing and the protection of privacy in genomics research". In: *Annual review of genomics and human genetics* 13 (2012), pp. 415–431.

[Kem+15] Petri Kemppainen, Christopher G Knight, Devojit K Sarma, Thaung Hlaing, Anil Prakash, Yan Naung Maung Maung, Pradya Somboon, Jagadish Mahanta, and Catherine Walton. "Linkage disequilibrium network analysis (LDna) gives a global view of chromosomal inversions, local adaptation and geographic structure". In: *Molecular ecology resources* 15.5 (2015), pp. 1031–1045.

[KFZ08] Yunjung Kim, Sheng Feng, and Zhao-Bang Zeng. "Measuring and partitioning the high-order linkage disequilibrium by multiple order Markov chains". In: *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society* 32.4 (2008), pp. 301–312.

[Kin06] Steven L Kinney. *Trusted platform module basics: using TPM in embedded systems*. Elsevier, 2006.

[KL15] Miran Kim and Kristin Lauter. "Private genome analysis through homomorphic encryption". In: *BMC medical informatics and decision making* 15.5 (2015), S3.

[KM11] Daniel Kifer and Ashwin Machanavajjhala. "No free lunch in data privacy". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 2011, pp. 193–204.

[Koc+19] Can Kockan, Kaiyuan Zhu, Natnatee Dokmai, Nikolai Karpov, M Oguzhan Külekci, David P Woodruff, and Süleyman Cenk Sahinalp. "Sketching Algorithms for Genomic Data Analysis and Querying in a Secure Enclave." In: *RECOMB*. Springer. 2019, pp. 302–304.

[Lam+18] Christoph Lambert, Maria Fernandes, Jérémie Decouchant, and Paulo Esteves-Verissimo. "Maskal: Privacy preserving masked reads alignment using intel sgx". In: *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*. IEEE. 2018, pp. 113–122.

[LCM16] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. "Dependence Makes You Vulnberable: Differential Privacy Under Dependent Tuples." In: *NDSS*. Vol. 16. 2016, pp. 21–24.

[Lew] Tanya Lewis. *Human Genome Project Marks 10th Anniversary*. https://www.livescience.com/28708-human-genome-project-anniversary.html. Accessed on: January 7th, 2019.

[Lip+17] 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". In: *Proceedings of the National Academy of Sciences* 114.38 (2017), pp. 10166–10171.

[Liu+21] Hai Liu, Changgen Peng, Youliang Tian, Feng Tian, and Zhenqiang Wu. "Privacy-Utility Equilibrium Protocol for Federated Aggregating Multiparty Genome Data". In: *Journal of Networking and Network Applications* 1.3 (2021), pp. 103–111.

[LLN14] Kristin Lauter, Adriana López-Alt, and Michael Naehrig. "Private computation on encrypted genomic data". In: *International Conference on Cryptology and Information Security in Latin America*. Springer. 2014, pp. 3–27.

[LS17] Zhigang Lu and Hong Shen. "A New Lower Bound of Privacy Budget for Distributed Differential Privacy". In: *PDCAT*. 2017, pp. 25–32.

[LYS15] Wen-Jie Lu, Yoshiji Yamada, and Jun Sakuma. "Privacy-preserving genome-wide association studies on cloud environment using fully homomorphic encryption". In: *BMC medical informatics and decision making* 15.5 (2015), S1.

[Man+07] Teri A Manolio, Laura Lyman Rodriguez, Lisa Brooks, Gonçalo Abecasis, Dennis Ballinger, Mark Daly, Peter Donnelly, Stephen V Faraone, Kelly Frazer, Stacey Gabriel, et al. "New models of collaboration in genome-wide association studies: the Genetic Association Information Network." In: *Nature genetics* 39.9 (2007).

[Man+18] Avradip Mandal, John C Mitchell, Hart Montgomery, and Arnab Roy. "Data Oblivious Genome Variants Search on Intel SGX". In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2018.

[MC19] Yakubu A Mohammed and YP Chen. "Ensuring privacy and security of genomic data and functionalities." In: *Briefings in bioinformatics* (2019).

[McK+16] Frank McKeen, Ilya Alexandrovich, Ittai Anati, Dror Caspi, Simon Johnson, Rebekah Leslie-Hurd, and Carlos Rozas. "Intel® software guard extensions (intel® sgx) support for dynamic memory management inside an enclave". In: *HASP*. 2016.

[Men+19] Ricardo Mendes, Tiago Oliveira, Vinicius Vielmo Cogo, Nuno Ferreira Neves, and Alysson Neves Bessani. "Charon: A secure cloud-of-clouds system for storing and sharing big data". In: *IEEE Transactions on Cloud Computing* (2019).

[Mot+20] Richard Mott, Christian Fischer, Pjotr Prins, and Robert William Davies. "Private Genomes and Public SNPs: Homomorphic encryption of genotypes and phenotypes for shared quantitative genetics". In: *Genetics* 215.2 (2020), pp. 359–372.

[Ney+17] Peter Ney, Karl Koscher, Lee Organick, Luis Ceze, and Tadayoshi Kohno. "Computer Security, Privacy, and DNA Sequencing: Compromising Computers with Synthesized DNA, Privacy Leaks, and More". In: *USENIX*. 2017.

[Ost+21] Andre Ostrak, Jaak Randmets, Ville Sokk, Sven Laur, and Liina Kamm. "Implementing Privacy-Preserving Genotype Analysis with Consideration for Population Stratification". In: *Cryptography* 5.3 (2021), p. 21.

[Par] EU Parlament. *The EU General Data Protection Regulation (GDPR)*. https://eugdpr.org/. Accessed on: January 17th, 2022.

[Pas+21] Túlio Pascoal, Jérémie Decouchant, Antoine Boutet, and Paulo Esteves-Verissimo. "DyPS: Dynamic, Private and Secure GWAS". In: *Proceedings on Privacy Enhancing Technologies* 2 (2021), pp. 214–234.

[Pas+23] Túlio Pascoal, Jérémie Decouchant, Antoine Boutet, and Marcus Völp. "I-GWAS: Privacy-preserving Interdependent Genome-Wide Asssociation Studies". In: *Proceedings on Privacy Enhancing Technologies* 1 (2023).

[PDV22] Túlio Pascoal, Jérémie Decouchant, and Marcus Völp. "Distributed and secure assessment of privacy-preserving releases of GWAS". In: *ACM/IFIP Middleware*. 2022.

[Pri] iDASH Privacy & Security Challenge. *Secure genome analysis competition*. http://www.humangenomeprivacy.org/2017/competition-tasks.html. Accessed on: March 13rd, 2020.

[PS19] Sandro Pinto and Nuno Santos. "Demystifying Arm TrustZone: A Comprehensive Survey". In: *ACM Computing Surveys (CSUR)* 51.6 (2019), p. 130.

[Pur+07] Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. "PLINK: a tool set for whole-genome association and population-based linkage analyses". In: *The American journal of human genetics* 81.3 (2007), pp. 559–575.

[Rai+17a] Jean Louis Raisaro, Florian Tramer, Zhanglong Ji, Diyue Bu, Yongan Zhao, Knox Carey, David Lloyd, Heidi Sofia, Dixie Baker, Paul Flicek, et al. "Addressing Beacon re-identification attacks: quantification and mitigation of privacy risks". In: *Journal of the American Medical Informatics Association* 24.4 (2017), pp. 799–805.

[Rai+17b] 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*. Tech. rep. EPFL infoscience, 2017.

[Rai+18] 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 Secure and Privacy-Preserving Exploration of Distributed Clinical and Genomic Data". In: *IEEE/ACM transactions on computational biology and bioinformatics* 16.4 (2018), pp. 1328–1341.

[Rei] Robinson Reid. *How big is the human genome?* Ed. by Precision Medicine. https://medium.com/precision-medicine/how-big-is-the-human-genome-e90caa3409b0. Accessed on: February 15th, 2022.

[Sad+18] Md Nazmus Sadat, Md Momin Al Aziz, Noman Mohammed, Feng Chen, Xiaoqian Jiang, and Shuang Wang. "SAFETY: secure gwAs in federated environment through a hYbrid solution". In: *TCBB* 16.1 (2018), pp. 93–102.

[Sam+15] Sahel Shariati 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: *2015 IEEE Security and Privacy Workshops*. IEEE. 2015, pp. 32–40.

[San+09a] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. "Genomic privacy and limits of individual detection in a pool". In: *Nature genetics* 41.9 (2009), pp. 965–967.

[San+09b] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. "Genomic Privacy and Limits of Individual Detection in a Pool: Supplementary Material". In: *Nature genetics* (2009).

[SAW13] Latanya Sweeney, Akua Abu, and Julia Winn. "Identifying participants in the personal genome project by name (a re-identification experiment)". In: *arXiv preprint:1304.7605* (2013).

[SB15a] Suyash S Shringarpure and Carlos D Bustamante. "Privacy risks from genomic data-sharing beacons". In: *The American Journal of Human Genetics* 97.5 (2015), pp. 631–646.

[SB15b] Sean Simmons and Bonnie Berger. "One size doesn't fit all: Measuring individual privacy in aggregate genomic data". In: *Security & Privacy Workshops*. 2015.

[SB16] Sean Simmons and Bonnie Berger. "Realizing privacy preserving genome-wide association studies". In: *Bioinformatics* 32.9 (2016), pp. 1293–1300.

[SBS19] Sean Simmons, Bonnie Berger, and Cenk S Sahinalp. "Protecting Genomic Data Privacy with Probabilistic Modeling". In: *PSB*. 2019.

[Sha79] Adi Shamir. "How to share a secret". In: *Communications of the ACM* 22.11 (1979), pp. 612–613.

[SSB16] Sean Simmons, Cenk Sahinalp, and Bonnie Berger. "Enabling privacy-preserving GWASs in heterogeneous human populations". In: *Cell systems* 3.1 (2016), pp. 54–61.

[ST18] Thomas Schneider and Oleksandr Tkachenko. "Towards Efficient Privacy-Preserving Similar Sequence Queries on Outsourced Genomic Databases". In: *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*. ACM. 2018, pp. 71–75.

[ST19] Thomas Schneider and Oleksandr Tkachenko. "EPISODE: efficient privacy-preserving similar sequence queries on outsourced genomic databases". In: *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*. 2019, pp. 315–327.

[Ste+18] Emil Stefanov, Marten Van Dijk, Elaine Shi, T-H Hubert Chan, Christopher Fletcher, Ling Ren, Xiangyao Yu, and Srinivas Devadas. "Path oram: An extremely simple oblivious ram protocol". In: *Journal of the ACM (JACM)* 65.4 (2018), pp. 1–26.

[Swe02] Latanya Sweeney. "k-anonymity: A model for protecting privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.

[SWT19] Giorgio Sirugo, Scott M Williams, and Sarah A Tishkoff. "The missing diversity in human genetic studies". In: *Cell* 177.1 (2019), pp. 26–31.

[Tir] Meg Tirrell. *Unlocking my genome: Was it worth it?* Ed. by CNBC. https://www.cnbc.com/2015/12/10/unlocking-my-genome-was-it-worth-it.html. Accessed on: January 10th, 2022.

[Tka+18] Oleksandr Tkachenko, Christian Weinert, Thomas Schneider, and Kay Hamacher. "Large-scale privacy-preserving statistical computations for distributed genome-wide association studies". In: *Asia CCS*. 2018.

[TPV17] Chia-Che Tsai, Donald E Porter, and Mona Vij. "Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX". In: *USENIX ATC*. 2017.

[Tra+15] Florian Tramèr, Zhicong Huang, Jean-Pierre Hubaux, and Erman Ayday. "Differential privacy with bounded priors: reconciling utility and privacy in genome-wide association studies". In: *SIGSAC*. 2015, pp. 1286–1297.

[UEL17] Chibuike Ugwuoke, Zekeriya Erkin, and Reginald L Lagendijk. "Privacy-safe linkage analysis with homomorphic encryption". In: *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE. 2017, pp. 961–965.

[USF13] Caroline Uhlerop, Aleksandra Slavković, and Stephen E Fienberg. "Privacy-preserving data sharing for genome-wide association studies". In: *The Journal of privacy and confidentiality* 5.1 (2013), p. 137.

[VAC19] Nora Von Thenen, Erman Ayday, and A Ercument Cicek. "Re-identification of individuals in genomic data-sharing beacons via allele inference". In: *Bioinformatics* 35.3 (2019), pp. 365–371.

[Vai+13] Jaideep Vaidya, Basit Shafiq, Xiaoqian Jiang, and Lucila Ohno-Machado. "Identifying inference attacks against healthcare data repositories". In: *AMIA Summits on Translational Science Proceedings* 2013 (2013), p. 262.

[Van06] Leendert Van Doorn. "Hardware virtualization trends". In: *ACM/Usenix International Conference On Virtual Execution Environments: Proceedings of the 2 nd international conference on Virtual execution environments*. Vol. 14. 16. 2006, pp. 45–45.

[VB13] Paulo Esteves Verissimo and Alysson Bessani. "E-biobanking: What have you done to my cell samples?" In: *Security & Privacy* 11.6 (2013), pp. 62–65.

[VG16] Effy Vayena and Urs Gasser. "Between openness and privacy in genomics". In: *PLoS medicine* 13.1 (2016), e1001937.

[VH09] Peter M Visscher and William G Hill. "The limits of individual identification from sample allele frequencies: theory and statistical analysis". In: *PLoS genetics* 5.10 (2009), e1000628.

[Vim+15] Sabrina De Capitani Di Vimercati, Sara Foresti, Stefano Paraboschi, Gerardo Pelosi, and Pierangela Samarati. "Shuffle index: Efficient and private access to outsourced data". In: *ACM Transactions on Storage (TOS)* 11.4 (2015), pp. 1–55.

[Wal+11] Lorelei Walker, Helene Starks, Kathleen M West, and Stephanie M Fullerton. "dbGaP data access requests: a call for greater transparency". In: *Science translational medicine* 3.113 (2011), pp. 113–134.

[Wan+09] 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: *Proceedings of the 16th ACM conference on Computer and communications security*. ACM. 2009.

[Wan+16] Shuang Wang, Yuchen Zhang, Wenrui Dai, Kristin Lauter, Miran Kim, Yuzhe Tang, Hongkai Xiong, and Xiaoqian Jiang. "HEALER: homomorphic computation of ExAct Logistic rEgRession for secure rare disease variants analysis in GWAS". In: *Bioinformatics* 32.2 (2016), pp. 211–218.

[WMC14] Shuang Wang, Noman Mohammed, and Rui Chen. "Differentially private genome data dissemination through top-down specialization". In: *BMC medical informatics and decision making* 14.1 (2014), pp. 1–7.

[Yao82] Andrew C Yao. "Protocols for secure computations". In: *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE. 1982, pp. 160–164.

[Yu+14] 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*. Springer. 2014, pp. 170–184.

[ZBA15] Yihua Zhang, Marina Blanton, and Ghada Almashaqbeh. "Secure distributed genome analysis for GWAS and sequence comparison computation". In: *BMC medical informatics and decision making* 15.5 (2015), S4.

[ZFC18] Yanjun Zan, Simon KG Forsberg, and Örjan Carlborg. "On the relationship between high-order linkage disequilibrium and epistasis". In: *G3: Genes, Genomes, Genetics* 8.8 (2018), pp. 2817–2824.

[Zha+14] Yongan Zhao, Xiaofeng Wang, Xiaoqian Jiang, Lucila Ohno-Machado, and Haixu Tang. "Choosing blindly but wisely: differentially private solicitation of DNA datasets for disease marker discovery". In: *Journal of the American Medical Informatics Association* 22.1 (2014), pp. 100–108.

[Zha+15] Yuchen Zhang, Wenrui Dai, Xiaoqian Jiang, Hongkai Xiong, and Shuang Wang. "Foresee: Fully outsourced secure genome study based on homomorphic encryption". In: *BMC medical informatics and decision making*. Vol. 15. 5. BioMed Central. 2015, pp. 1–11.

[Zha+18] Shifa Zhangy, Anne Kim, Dianbo Liu, Sandeep C Nuckchadyy, Lauren Huangy, Aditya Masurkary, Jingwei Zhangy, Lawrence Pratheek Karnatiz, Laura Martinezx, Thomas Hardjono, et al. "Genie: A Secure, Transparent Sharing and Services Platform for Genetic and Health Data". In: *arXiv preprint arXiv:1811.01431* (2018).

[Zha+22] Yanjun Zhang, Guangdong Bai, Xue Li, Surya Nepal, Marthie Grobler, Chen Chen, and Ryan KL Ko. "Preserving Privacy for Distributed Genome-Wide Analysis Against Identity Tracing Attacks". In: *IEEE Transactions on Dependable and Secure Computing* (2022).

[Zho+11] Xiaoyong 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*. 2011.

[ZN+15] Guy Zyskind, Oz Nathan, et al. “Decentralizing privacy: Using blockchain to protect personal data”. In: *SPW*. 2015.

[ZN08] Elias A Zerhouni and Elizabeth G Nabel. “Protecting aggregate genomic data”. In: *Science* 322.5898 (2008), pp. 44–44.

[Zoo+16] Justin M Zook, David Catoe, Jennifer McDaniel, Lindsay Vang, Noah Spies, Arend Sidow, Ziming Weng, Yuling Liu, Christopher E Mason, Noah Alexander, et al. “Extensive sequencing of seven human genomes to characterize benchmark reference materials”. In: *Scientific data* 3.1 (2016), pp. 1–26.

[ZPS11] Zhenfei Zhang, Thomas Plantard, and Willy Susilo. “Reaction attack on outsourced computing with fully homomorphic encryption schemes”. In: *International Conference on Information Security and Cryptology*. Springer. 2011, pp. 419–436.