Protecting Privacy and Security of Genomic Data in i2b2 With Homomorphic Encryption and Differential Privacy

Jean Louis Raisaro, Gwangbae Choi, Sylvain Pradervand, Raphael Colsenet, Nathalie Jacquemont, Nicolas Rosat, Vincent Mooser and Jean-Pierre Hubaux

Abstract—Re-use of patients' health records can provide tremendous benefits for clinical research. Yet, when researchers need to access sensitive/identifying data, such as genomic data, in order to compile cohorts of well-characterized patients for specific studies, privacy and security concerns represent major obstacles that make such a procedure extremely difficult if not impossible. In this paper, we address the challenge of designing and deploying in a real operational setting an efficient privacy-preserving explorer for genetic cohorts. Our solution is built on top of the i2b2 (Informatics for Integrating Biology and the Bedside) framework and leverages cutting-edge privacy-enhancing technologies such as homomorphic encryption and differential privacy. Solutions involving homomorphic encryption are often believed to be costly and immature for use in operational environments. Here, we show that, for specific applications, homomorphic encryption is actually a very efficient enabler. Indeed, our solution outperforms prior work by enabling a researcher to securely compute simple statistics on more than 3,000 encrypted genetic variants simultaneously for a cohort of 5,000 individuals in less than 5 seconds with commodity hardware. To the best of our knowledge, our privacy-preserving solution is the first to also be successfully deployed and tested in a operation setting (Lausanne University Hospital).

Index Terms—Genomic privacy, homomorphic encryption, differential privacy, i2b2.

1 Introduction

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

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

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

For this reason, in the last few years, researchers from both the computer science and medical fields have started collaborating to design new advanced solutions that could enable the outsource of sensitive data while protecting individuals' medical privacy and, in particular, genomic privacy [1]. However, to obtain acceptance and to be adopted in the real world, these solutions need to be deployed and assessed in concrete operational scenarios.

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

Indeed, in order to be used in the real world, we built

- *J.L. Raisaro and G. Choi are co-first authors.*
- *J.L. Raisaro, G. Choi and J.-P. Hubaux are with the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.*
- *S. Pradervand, R. Colsenet, N. Jacquemont, N. Rosat and V. Mooser are with Service de Soutien à la Recherche Clinique, Centre Hospitalier Universitaire Vaudois (CHUV), Lausanne, Switzerland.*

our system on top of the most widespread open-source framework for exploring clinical research data-warehouses, namely *Informatics for Integrating Biology and the Bedside* (i2b2) [2]. The i2b2 framework was jointly developed by the Harvard Medical School and Massachusetts Institute of Technology to enable clinical researchers to use existing de-identified clinical data and only IRB-approved genomic data for discovery research and design of target therapies. i2b2 is currently used by more than 300 medical institutions worldwide for translational research or academic purposes [3] and its software is maintained and constantly upgraded by the i2b2 tranSMART Foundation. Yet, as most cohort explorers, i2b2 is designed to be primarily used in trusted environments as it does not provide any specific protection mechanism for genomic data, or other types of identifiable information, apart from standard access control and data de-identification [4], both proven to be ineffective [5], [6], [7], [8], [9], [10], [11]. This limitation substantially restricts the scope of potential feasibility studies that could be conducted in less controlled and protected environments for accelerating medical research.

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

We extensively tested the performance of our solution in a real operational setting for different cohort sizes, and we found the overhead introduced by privacy-preserving techniques to be entirely acceptable thanks to the ciphertext packing technique enabled by the *somewhat* homomorphic encryption scheme on which our solution is based. The response time is linear in the number of selected patients and always in the order of a few seconds for standard queries and cohort sizes, which outperforms the state of the art [12], [13], [14], [15], [16]. The rest of this paper is organized as follows. In the next section, we summarize the related work. In Section 3, we briefly describe the i2b2 frameworks and the main concepts in genomics and cryptography used throughout the paper. In Section 4, we introduce the system and threat models and describe the proposed solution in detail. In Section 5, we describe its implementation and evaluate the performance using real genetic data in the operational setting of CHUV. In Section 6, we discuss our findings and we conclude the paper in Section 7. In the Appendix, we discuss a benchmark of some state-of-the-art homomorphic cryptosystems that we have evaluated for our solution.

2 Related Work

For many years, researchers have assumed that releasing anonymized genomic data for research purposes would not compromise participants' privacy. However, it has been shown at multiple rounds [5], [6], [7], [8], [9], [10], [11] that standard anonymization techniques are ineffective on genomic data. As a response to these concerns, many solutions were proposed in the past few years, with the goal of protecting genomic data and enabling their utility for medical research. We can put them in two main categories: (i) approaches that focus on the protection of data confidentiality against illegitimate access and (ii) approaches that mitigate the risk of sensitive attribute inference from legitimate accessed genomic data.

The first category includes works such as the one by Canim *et al.* [13] where the authors describe how to outsource analyses on genomic data to a commercially available cryptographic hardware. Also in this category, a more recent study by Kamm *et al.* [14] proposes a new scheme for generating aggregated statistics on genomic data by using secure multi-party computation based on homomorphic secret sharing. Their technique requires the presence of multiple non-colluding servers. Xie *et al.* introduce in their work [15] a novel cryptographic strategy based on secure multi-party computation to securely perform meta-analysis for genetic association studies in large consortia, whereas Wang *et al.* [16] also rely on secure multi-party computation techniques to securely compare genomes across institutions. Finally, several other recent works [12], [17], [18], [19], [20] propose using homomorphic encryption to protect genomic information in order to allow researchers to perform some statistics directly on the encrypted data and decrypt only the final result.

The second category includes the work by Uhler *et al.* [21] that proposes new methods for releasing aggregate results from genome-wide association studies (GWAS) without compromising a participant's privacy, by focusing on the differentially private release of minor allele frequencies. A similar approach is also adopted by Johnson and Shmatikov [22], Yu *et al.* [23] and Simmons and Berger [24], [25], [26], whereas Tramer *et al.* [27] investigated a relaxation of differential privacy providing a better tradeoff between privacy and utility.

In our work, contrarily to those mentioned above that address a single problem, we mitigate both risks simultaneously. Our solution makes use of both homomorphic encryption (HE) to protect patients' genomic data confidentiality against illegitimate access at rest and during processing, and differential privacy (DP) to protect against re-identification attacks. Whereas for the latter we propose a straightforward application of DP, for the former we designed a new optimized solution based on HE that outperforms the state-of-the-art. Moreover, to the best of our knowledge, our solution is the first to be deployed and tested in a real medical research environment. Another work by McLaren *et al.* [28] describes a successful deployment in a medical operational environment. Yet, in that work HE is

used to protect data for personalized medicine and not for research purposes.

Two research contributions from the i2b2 academic community describe how to use genomic data within the i2b2 framework: Phillips *et al.* [29] propose to use a genomic ontology directly in the i2b2 native framework, whereas Gabetta *et al.* propose BigQ [30], an i2b2 plugin handling genomic variants and their annotations. Yet, both works do not at all address privacy problems due to the use of genomic information. They simply rely on the i2b2 standard security protections that are based on simple access control mechanisms as described in the work by Murphy *et al.* [4]. To the best of our knowledge, our work is the first to combine i2b2 with advanced privacy-enhancing technologies for genomic data protection.

3 PRELIMINARIES

In this section, we briefly summarize the main concepts in cryptography and genomics that are used in the paper.

3.1 Notation

In this paper, we denote the cardinality of a set A by $|A|$. We denote x uniformly chosen from the set X as $x \xleftarrow{U} X$. Moreover, we use boldface letters to represent vectors and regular letters for polynomials.

3.2 Cryptographic Background

In this section, we briefly explain the cryptographic concepts necessary for understanding the rest of the paper.

Homomorphic Encryption. Homomorphic encryption (HE) is a special type of encryption that supports computation on encrypted data. In 2009, Craig Gentry [31] introduced for the first time a special type of HE enabling for *arbitrary computations* on ciphertexts called fully homomorphic encryption (FHE).

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

$$\mathcal{D}(h(\mathcal{E}(m_1, k_p), \mathcal{E}(m_2, k_p)), k_s) = f(m_1, m_2) \quad (1)$$

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

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

Our proposed solution is based on the Fan and Vercauteren (FV) cryptosystem [32] which is the state-of-the-art lattice-based leveled homomorphic encryption scheme based on the *Ring Learning With Errors* (RLWE) problem. The FV scheme ensures indistinguishability against chosen plaintext attack if the standard RLWE problem is hard. Moreover, as other lattice-based cryptosystems, it is supposed to be quantum-resistant. Let ℓ be a power of 2 and a polynomial degree, q be a coefficient modulus, t be a plaintext modulus, $\mathcal{X}$ be a noise distribution over a polynomial ring $\mathbb{Z}_q[x]/(x^\ell + 1)$, and $m \in \mathbb{Z}_t[x]/(x^\ell + 1)$ be a plaintext polynomial. Let $s \xleftarrow{U} \mathbb{Z}_q[x]/(x^\ell + 1)$ be the secret key and $\boldsymbol{p} = (p_0, p_1) = (-(a \cdot s + e) \bmod q, a)$, be the public key where $e \leftarrow \mathcal{X}$ and $a \xleftarrow{U} \mathbb{Z}_q[x]/(x^\ell + 1)$. Then the FV scheme works as follows:

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

$$Enc(m, \boldsymbol{p}) = (c_0, c_1) = \quad (2)$$
$$((p_0 \cdot u + e_1 + \lfloor \frac{q}{t} \rfloor \cdot m) \bmod q, (p_1 \cdot u + e_2) \bmod q),$$

- *Decryption:*

$$Dec(\boldsymbol{c}, \boldsymbol{s}) = \left\lfloor \frac{t}{q} \cdot ((c_0 + c_1 \cdot s) \bmod q) \right\rceil \bmod t \quad (3)$$

- *Homomorphic addition:*

$$Add(\boldsymbol{c}, \boldsymbol{c'}) = ((c_0 + c'_0) \bmod q, (c_1 + c'_1) \bmod q) \quad (4)$$

We do not report the definition of homomorphic multiplication as it is not used in our solution. For further details we refer the reader to the original paper [32]. Note that we chose the FV scheme because, to the best of our knowledge, it provides the best performance in terms of homomorphic computations and storage overhead for the operations required in the proposed solution. We also compared the FV scheme with the Elliptic curve ElGamal (EC-ElGamal) cryptosystem [33] and Yet Another Somewhat Homomorphic Encryption (YASHE) scheme [34]. Benchmark details can be found in the Appendix.

Ciphertext Packing. Ciphertext packing [35] is a technique that can be used to reduce the overall size of the ciphertext and improve the efficiency of homomorphic operations. Despite recent advances, practical HE is still quite expensive. This is because security considerations require ciphertexts to be large, thus slowing down homomorphic computations. Ciphertext packing represents the main technique for dealing with this problem as a vector of plaintext values, and not a single value, can be encrypted in only one ciphertext. Homomorphic operations are applied to these vectors component-wise.

More formally, let $\mathsf{CS}(K_p, K_s, P, C, \mathcal{E}, \mathcal{D})$ be a cryptosystem, and m_0, m_1, ..., be the messages to be encrypted where $m_i \in M, \ \forall i$. Let also $n = \left\lfloor \frac{|P|}{|M|} \right\rfloor$ and P_1, P_2, ..., P_n be n independent subspaces of P where $|P_j| \geq |M|, \ \forall j$. When $|P| \geq 2 \cdot |M|$, we can encrypt at most n messages into one ciphertext by encrypting $m' = m_1 p_1 + m_2 p_2 + \cdots + m_n p_n$, where p_j is the basis of the subspace P_j.

For example, when $P = \mathbb{Z}_q[x]/(x^\ell + 1)$ and $M = \mathbb{Z}_t$, we can encrypt at most ℓ messages into one ciphertext with $m' = m_0 + m_1 x + \cdots + m_{\ell-1} x^{\ell-1}$.

Differential Privacy. Differential privacy is an approach to privacy-preserving reporting of results, introduced by Cynthia Dwork [36], that guarantees that a given randomized statistic, $f(D) = R$, computed on a dataset D_1 behaves almost the same when computed on the neighbor dataset D_2 that differs from D_1 in exactly one element. More formally we have that

$$\Pr\left[f(D_1) = R_0\right] \leq \exp(\epsilon) \cdot \Pr\left[f(D_2) = R_0\right], \qquad (5)$$

where the parameter ϵ is a privacy parameter: the closer it is to 0 the more privacy is ensured. The most straightforward method [37] for achieving ϵ-differential privacy consists in perturbing the output of the statistic with noise drawn from the Laplace distribution with mean 0 and scale $\frac{\Delta f}{\epsilon}$, where Δf is known as the *sensitivity* of f:

$$\Delta f = \max_{D_1, D_2} ||f(D_1) - f(D_2)||_1. \qquad (6)$$

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

3.3 Genomic Background

In this section, we briefly introduce the basic genomic concepts used in this paper.

Genomic Variant. The human genome consists of almost 3 billion base pairs made of a 4-letter alphabet (A, C, T and G). Around 99.9% of it is identical between any two individuals and the remaining part ($\sim$0.1%) is referred to as *genetic variation*. Most commonly, this genetic variation comes from differences in single nucleotides, called *single nucleotide variants* (SNVs). Yet, there exist many other types of genetic differences also involving multiple nucleotides such as *insertions/deletions* (INDELs), *duplications* (DUPs), *copy number variants* (CNVs) and more complex *structural variants* (SVs). In the human population, a given genetic locus can have several possible versions (or alleles) with different genetic variations. Due to the diploid nature of somatic human cells, a human genome comprises two sets of chromosomes – one inherited from each parent. Therefore, each individual either has two copies of the same allele/variant (*homozygous*) or two copies of different alleles/variants (*heterozygous*).

Variant Call Format (VCF). The Variant Call Format (VCF) [38] is the main format for storing genetic variants of one or more individuals with respect to the reference genome. The VCF consists of two parts: header and content. The header contains the meta-information about the file and data along with the definition of file variables. The content holds the information about the genetic variants for each individual. Each variant is uniquely identified by (i) its chromosomal position (CHR, POS), (ii) the reference allele (REF), and (iii) the alternate allele (ALT) We can separate the content into two parts by the characteristics of the data. Each line of the content corresponds to the information about a variant and the genotype (i.e., the value) of this variant for each individual in the VCF. We call the information about the variant, such as CHR, POS, REF, ALT, meta-data. Meta-data is not sensitive from the privacy perspective as it is public information, as opposed to genotype information that is sensitive and must be protected.

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

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

4 METHODS

In this section, we introduce CHUV's system and threat models. We then outline the functional requirements that our system should satisfy and finally we describe our proposed privacy-preserving solution in detail. Note that these system and threat models can be easily adapted to other similar healthcare providers.

4.1 System Model

The CHUV's information system consists of two physically separated networks, as depicted in Fig.1. Each of them hosts different services: (i) the main network of the hospital, also called *clinical network* and (ii) a *research network* that is shared with the University of Lausanne (UNIL).

- *Clinical network.* The clinical network is used for hospital's clinical daily activities. It hosts all services used for daily healthcare and administration purposes along with the clinical research data-warehouse[1] (DWH-RC) that contains pseudonymized clinical and genomic data of patients. This network is very controlled and protected by a firewall that blocks all incoming network traffic. Authorized users are authenticated and their activities are constantly logged.
- *Research network.* The research network is also protected by a firewall that blocks all incoming network traffic except that coming from the clinical network but the level of control is weaker with respect to the clinical network as users' activities are not logged. It hosts multiple services used by clinical or academic researchers in their research activities; i2b2 is one of these services. The i2b2 service is composed of (i) an i2b2 server to which pseudonymized and de-identified clinical data along with pseudonymized and encrypted genomic data are pushed from the DWH-RC and (ii) a proxy server

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

which is devoted to support the decryption phase and the storage of partial decryption keys. Both servers are physically separated from each other, protected by different firewalls and equipped with an intrusion detection system. Moreover, the two servers cannot communicate with each other. Researchers already in the network access the i2b2 service after authentication through an internal Web-client.

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

4.2 Threat Model

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

The honest-but-curious attacker can be represented by a careless or disgruntled employee of the hospital (i.e., an insider) or a hacker who has illegitimate access to the i2b2 server and tries to obtain patients' private genomic and clinical information without altering the protocol. Note that although this information was pseudonymized and there is no direct link with patients' identities, re-identification would still be possible due to the identifying nature of the genome and to some auxiliary (and often publicly available) information (e.g., public genomic databases, recreational websites, online social networks, etc.) that the attacker might exploit [5], [6], [7], [8], [9], [10], [11]. As a consequence, a potential loss of clinical and genomic data could be extremely dangerous, not only for the patients but also for the reputation of the medical institution itself. Re-identified health-care records are nowadays extremely valuable for hackers as, according to a recent report by IBM [39], their value on the black market is as much as 60 times more than that of stolen credit cards. We assume that the i2b2 server and the proxy server do not collude or, in other words, that they are not simultaneously compromised.

The malicious-but-covert adversary can be represented by a malicious but legitimate user of i2b2 (e.g., a malicious researcher) or hacker who breaks into the research network and uses the i2b2 service to re-identify an individual in a subset of patients with specific clinical characteristics. In particular, an attacker with already some genomic information about the victim (e.g., the value of some of her genetic variants) might repeatedly query the i2b2 service with this genomic information and use the system as an oracle. As such, he could re-identify the presence of the victim in a sensitive subset of individuals (e.g., all cancer patients or all HIV-positive patients, etc.) and infer his/her health status. For example, the attacker could exploit the aggregate information obtained from the cohort explorer as described in well-known attacks such as Homer's attack [5] and the Beacon attack [6].

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

4.3 Functional and Computational Requirements

According to CHUV, the functional requirements of our privacy-preserving cohort explorer should be based on other well-known tools for exploration of genetic cohorts such as the Beacon system of Global Alliance for Genomics & Health (GA4GH) [40] and the ExAC browser of the Broad Institute [41]. For example, through the Beacon system researchers can query a database of genomes for the presence of a specific mutation, whereas researchers using the ExAC browser can also have information about the alternate allele count and frequency of the queried mutation.

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

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

Moreover, the storage overhead introduced by encryption should be kept at the lowest possible level and the query round trip time in the order of seconds for a smooth user-experience.

4.4 Proposed Solution

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

System Initialization. The system initialization phase takes place at the clinical research data-warehouse (DWH-RC) where clinical and genomic data are stored with patients' pseudonyms and can be accessed only by a group of a few trusted and authorized individuals. The first step consists in setting the parameters of the FV cryptosystem

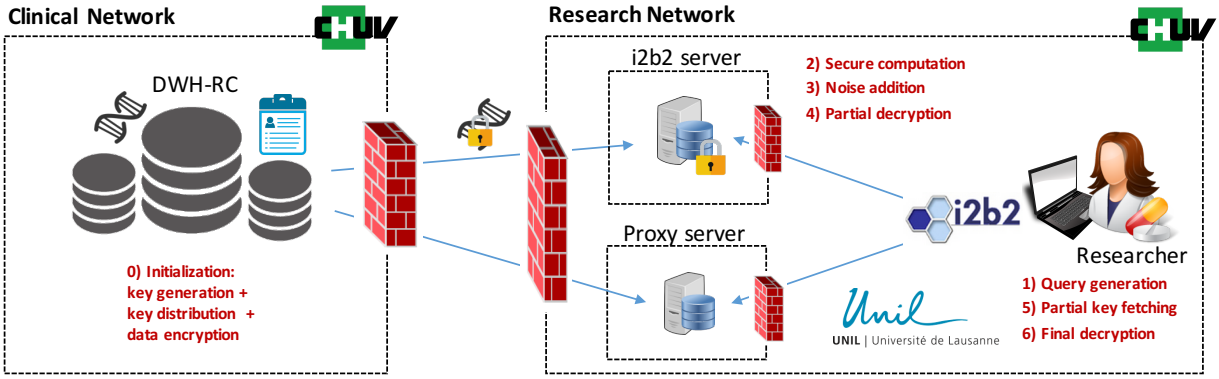


Fig. 1: Architecture of the proposed solution.

(l, t and q) according to the desired security level (e.g., 80 bits security) and the maximum number of additions and multiplications to be supported by the system. In our case, the maximum number of additions should be at least twice the number of individuals in the database because it corresponds to the maximum number of alleles that could be involved in a counting query. Multiplication is not needed. For the optimal selection of the FV parameters, we refer the reader to the original work by Fan and Vercauteren [32]. Then, a public key $\boldsymbol{p}$ and a secret key $\boldsymbol{s}$ are generated as described in Section 3.2.

After key generation, the VCF file is parsed and each genotype is encoded following the scheme described either in Table 1 (for phased genotypes) or in Table 2 (for unphased genotypes). For simplicity, in the rest of the paper we describe only the encoding for unphased genotypes described in Table 2 (but the encoding in Table 1 is equally supported by the proposed solution). As the number of potential genotype values is 6 (we are also including *no-calls* as they can be used when allele or genotype frequencies are computed), we use three values for genotype encoding: the first two values indicate the presence of zero, one or two alternate alleles, whereas the third value reports the number of no-calls in the genotype.

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

TABLE 1: Encoding for phased genotypes.

For each individual in the VCF file, consecutive genotypes are packed into 3 sets of polynomials by using the packing technique described in Section 3.2. Let $(\text{gt1}_i, \text{gt2}_i, \text{no-call}_i)$ be the encoded genotype for variant i.

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

TABLE 2: Encoding for unphased genotypes

Then, we can pack at most ℓ genotypes in three polynomials

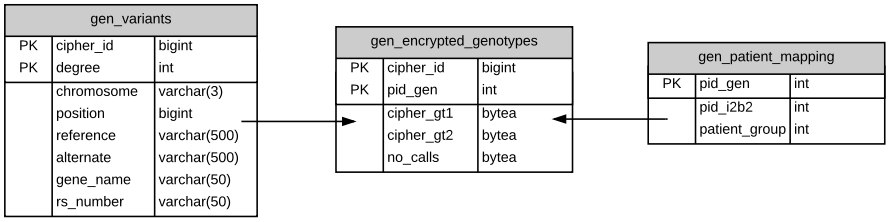
$$\mathbf{gt1_j} = \sum_{k=0}^{\ell-1} \text{gt1}_{j\ell+k} x^k, \tag{7}$$

$$\mathbf{gt2_j} = \sum_{k=0}^{\ell-1} \text{gt2}_{j\ell+k} x^k, \tag{8}$$

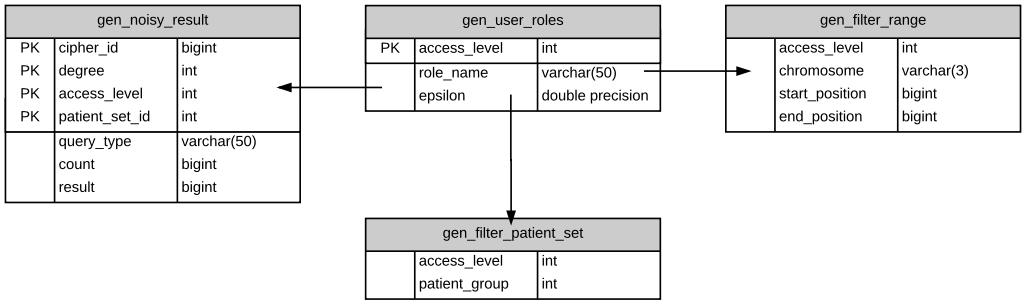
$$\textbf{no-call}_j = \sum_{k=0}^{\ell-1} \text{no-call}_{j\ell+k} x^k, \tag{9}$$

where ℓ is the polynomial degree and j the ciphertext index. Polynomials $\mathbf{gt1_j}$ and $\mathbf{gt2_j}$ are finally encrypted with the public key $\boldsymbol{p}$ at the DWH-RC to obtain $\mathbf{cipher\_gt1_j} = Enc(\mathbf{gt1_j}, \boldsymbol{p})$ and $\mathbf{cipher\_gt2_j} = Enc(\mathbf{gt2_j}, \boldsymbol{p})$ which are pushed to the i2b2 server for storage along with $\textbf{no-call}_j$, patients' pseudonyms, de-identified clinical data, and $\boldsymbol{p}$, as shown in Fig.2a. Note that $\textbf{no-call}_j$ does not need to be encrypted as it does not leak any information on the value of the genotype.

User Assignment. Assignment of a new user also takes place at the DWH-RC. During this phase, for each new user, the secret key $\boldsymbol{s}$ is randomly divided into two shares s_1 and s_2 such that $\boldsymbol{s} = s_1 + s_2$. The two partial secret keys s_1 and s_2 are then sent, for storage, to the i2b2 server and proxy server, respectively. This procedure avoids a single point of failure in the system: if one of the two servers is compromised, data are still protected because only with the full secret key $\boldsymbol{s}$ the attacker can successfully decrypt. Note that ideally, to ensure better security, the i2b2 server and the proxy server should be part of two completely different organizations. Yet, because of the CHUV's infrastructure these two servers are within the same network. Nevertheless, they are phys-



(a) Scheme for storing encrypted genetic variants.



(b) Scheme for storing users' access rights.

Fig. 2: Database schemes for the proposed solution.

ically separated, managed by different administrators and cannot communicate with each other.

Access rights for each new user are also assigned during this phase and stored on the i2b2 server, as shown in Fig. 2b. Different users might have different rights to access patients' private information. Our system provides full customization on three different levels of access: (i) query result, (ii) patients' set, and (iii) variants' set. For example, a junior researcher might have access only to perturbed query results, where some noise has been added on the true result to satisfy the notion of differential privacy, whereas a senior researcher can obtain accurate information. Also, depending on his specialization or on the IRB-approved study protocol, a researcher might have access only to a given subset of patients or a given subset of genetic variants. For example, an oncologist might have access only to data of patients with cancer and might not be allowed to query genetic variants related to other diseases such as diabetes or coronary artery disease.

We acknowledge that if the i2b2 server is compromised by a honest-but-curious adversary, information on users' access rights could leak because they are stored in the clear. Protecting users' access rights is not the focus of this project. Yet, we are planning to address such an issue in future work by exploring even more sophisticated solutions based on *attribute-based somewhat homomorphic encryption (ABSHE)* [42].

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

(i) *Query generation.* In the query generation, after password-based authentication, the user of our privacy preserving cohort explorer (i.e., CHUV researcher) builds his query in two steps through the i2b2 Webclient. In the first step, he selects a set of inclusion and exclusion clinical criteria in order to identify the set of patients for which he wants to explore their genetic data. For example, a researcher might want some aggregate genetic information on some specific variants for patients with cancer who underwent a specific treatment and who had a positive outcome. In the second step, once the patient set has been identified, the user selects the set of variants of interest and the summary statistics he wants to obtain from the ones specified in Section 4.3 and finally submits the query.

(ii) *Query processing.* During the query processing, the i2b2 server verifies the access rights of the querier and his query definition. If the verification is successful, the server retrieves from the database the list of ciphertexts containing the encrypted genotypes of the variants specified by the query. The ciphertexts are then used to homomorphically compute the summary statistics requested by the user. We designed different secure algorithms for computing the different summary statistics described in Section 4.3. The details of these algorithms are explained in the Supplementary Materials.

(iii) *Result perturbation.* Depending on the role and access level of the user, the i2b2 server can perturb the encrypted query result to satisfy the notion of differential privacy and prevent re-identification attacks. In particular, we assume i2b2 users hold a single account and do not collude. This assumption is reasonable in practice, as by definition, in order to collude, a user must involve someone else. Moreover, users are assigned a total privacy budget ϵ_{tot} whose value is decided by i2b2 administrators and may depend on the user's role and

level of trustworthiness. For each new query i from the same user, the i2b2 server draws independent noise from the Laplace distribution with mean 0 and scale $\frac{\Delta f}{\epsilon_i}$, where f is the requested aggregation function, encrypts it and homomorphically adds it to the encrypted query result in order to satisfy ϵ_{tot}-differential privacy. The user's privacy budget ϵ_{tot} is then reduced by ϵ_i and keeps decreasing every time a new query is answered; the i2b2 server will keep on providing query answers to the user until his budget runs out. The value of ϵ_i can be fixed, if set by the database administrator, or adaptive, if set by the user who may use a small value of ϵ_i and incur more noise for preliminary queries whose expected result is large and save the budget for more specific queries. What is the right value of ϵ_i is out of the scope of this paper. We note that Δf is equal to 1 for count queries and to $\frac{1}{n}$ for predicate queries (i.e., queries asking the fraction of elements in a database that satisfy a specific predicate), where n is the number of patients in the database. For consecutive queries, Δf grows linearly.

(iv) *Result partial decryption.* After computation of perturbed/unperturbed encrypted summary statistics, the i2b2 server partially decrypts them with the first part of the users' secret key s_1 (which is stored in the i2b2 database). In particular, from a ciphertext $c = (c_0, c_1)$ we obtain, after partial decryption, a new ciphertext $c' = (c'_0, c'_1)$, as described in Section 4.5. Finally, the i2b2 server sends back to the user the encrypted polynomial c'_1 and the coefficients of the encrypted polynomial c'_0 matching the variants specified by the query and user's access level. Note that, for the sake of information minimization, only the specified coefficients of the encrypted polynomial c'_0 are sent by the server to the user. In other words, we do not want the user to obtain the summary statistics of all the variants packed in the same ciphertext, but only the ones he has requested access to.

(v) *Result decryption.* At the user-side, the Web-client fetches the second part of the user's secret key s_2 from the proxy server and performs the full decryption to obtain the final results of the query, as described in Section 4.5. For performance reasons, part of the full decryption (i.e., the polynomial multiplication $c'_1 \cdot s_2$) could be run at the proxy server. Note that, by observing only s_2 and c'_1, the proxy server cannot infer anything about the plaintext.

4.5 Partial Decryption With FV Scheme

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

Definition 1 (Partial Key Generation)**.** *Let q be the ciphertext modulus, ℓ be the polynomial degree and $s \in \mathbb{Z}_q[x]/(x^\ell + 1)$ be the secret key [32]. Then, the partial key set (s_0, s_1) can be generated as $s_0 \xleftarrow{U} \mathbb{Z}_q[x]/(x^\ell + 1)$ and $s_1 = s - s_0$.*

Definition 2 (Partial Decryption)**.** *Let q be the modulus, ℓ be the polynomial degree, t be the plaintext modulus, (s_0, s_1) be the partial key set from Definition 1 and $\boldsymbol{c} = (c_0, c_1)$ be the ciphertext where $c_0, c_1 \in \mathbb{Z}_q[x]/(x^\ell + 1)$. Then, we can define the partial decryption and the full decryption as follows,*

$$\begin{aligned} Partial\_Dec(\boldsymbol{c}, s_0) &= \boldsymbol{c'} = (c'_0, c'_1) = (c_0 + c_1 \cdot s_0,\ c_1) \\ Full\_Dec(\boldsymbol{c'}, s_1) &= Dec(\boldsymbol{c'}, s_1), \end{aligned} \tag{10}$$

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

Proof. We have

$$\begin{aligned} &Full\_Dec(Partial\_Dec(\boldsymbol{c}, s_0), s_1) \\ &\quad = \left\lfloor \frac{t}{q}((c_0 + c_1 \cdot s_0 + c_1 \cdot s_1) \bmod q) \right\rceil \bmod t \\ &\quad = \left\lfloor \frac{t}{q}((c_0 + c_1 \cdot s) \bmod q) \right\rceil \bmod t \\ &\quad = Dec(\boldsymbol{c}, s) \end{aligned} \tag{11}$$

By Definition 1, $s_0 + s_1 = s$, so Eq.(11) holds. Thus, $Full\_Dec(Partial\_Dec(\boldsymbol{c}, s_0), s_1)$ is equivalent to $Dec(\boldsymbol{c}, s)$.

□

4.6 Security Analysis

In this section, we discuss about the security of our system with respect to the protection of genomic data. The protection of clinical data is not the focus of this paper. However, differently from genomic data, various anonymization techniques can be applied to protect clinical data and satisfy formal notions of privacy such as *k-anonymity* [43], *l-diversity* [44] or *t-closeness* [45]. Because using anonymization techniques could modify the original clinical data and reduce the overall utility of the system, our system can also be adapted to clinical data in case full accuracy is required.

Our system consists of four different participants: the data-warehouse (DWH), the i2b2 server (IS), the proxy server (PS) and the i2b2 user (U). As DWH is trusted, we only focus on IS, PS and U. As discussed in Section 4.2, we assume the honest-but-curious adversarial model for both IS and PS and the malicious-but-covert model for U. Moreover, we recall that s_1 represents the partial secret key stored at IS and s_2 represents the partial secret key stored at PS for a specific user. Similarly, $\mathbf{s_1}$ and $\mathbf{s_2}$ represent the sets of partial keys for all users in the system at IS and PS, respectively.

i2b2 Server. If the i2b2 server is compromised, the adversary can access the encrypted genomic data, the set of partial secret keys $\mathbf{s_1}$, the role and access level of each user, the amount of noise used for perturbing query results, accessible chromosomal ranges, and the history of queries. Moreover, from the history of queries and the accessible chromosomal ranges, the attacker can infer users' access patterns, their potential interests and medical specialties. Yet, we note that our goal is to preserve the privacy of patients, not of i2b2 users.

As such, although the adversary has encrypted genomic data and $\mathbf{s_2}$, he cannot obtain any sensitive genomic information about the patients as he still needs $\mathbf{s_1}$ to decrypt. However, because IS and PS cannot be

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

Proxy Server. If the proxy server is compromised, only the set of partial secret keys $\mathbf{s_2}$ is leaked. As before, because PS and IS cannot collude, the attacker cannot obtain any sensitive genomic information. Also in this case, new partial keys need to be generated for all users in order to avoid the decryption of leaked data with a later attack on IS.

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

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

5 Experiment and Results

In this section, we describe how we implemented and deployed our solution in the real operational setting of the Lausanne University Hospital. We evaluate its performance on real genomic and clinical data.

5.1 Plugin Implementation

We implemented our privacy-preserving solution as a plugin of the i2b2 framework. The i2b2 architecture consists of two major pieces: The first is the back-end infrastructure (the "Hive") that is responsible for the security aspects, the access rights and for managing the underlying data repository. The second piece is the user Web-client: a front-end application suite of query and mining tools that enables users to ask questions about patients' data on the i2b2 server. The native version of i2b2 does not include any support neither for privacy-preserving data processing nor for managing genomic data but it is only focused around cohort identification based mostly on clinical and demographic data. As shown in Supplementary Fig. S1, after logging in, a user can drag-and-drop search terms from the clinical ontology into the Venn diagram-like interface to construct his cohort of patients. Yet, i2b2 is easily extensible thanks to its modular design. As a consequence, we implemented our privacy-preserving plugin as a totally independent module that can be easily loaded during the setup of i2b2. Our plugin is composed of four main parts: (i) a data importation tool, (ii) a back-end module for the i2b2 server, (iii) a back-end module for the proxy server, and (iv) a front-end module for the i2b2 native Web-client.

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

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

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

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

Web-client Module. The Web-client front-end module is written in JavaScript and can be loaded from the native i2b2 Web-client. It consists of a *query builder* (Supplementary Fig. S2a), where the user can drag-and-drop a patient set (previously constructed through the native interface) and type into a search bar a set of genetic variants of interest, and a *result visualizer* (Supplementary Fig. S2b), where the user can visualize the results of his current and previous queries. The user is allowed to enter genetic variants by gene name, dbSNP identifier or chromosomal position and to select the summary statistic he is interested in.

5.2 Performance Evaluation

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

Experimental Setup. To show the practicality of our solution, we used only commodity hardware for our experiments. The i2b2 server module and the proxy server module run on two servers in the CHUV's research network. Their configurations are described in Table 3. For both servers, we limited the number of threads to 8. At the user side, we used an off-the-shelf laptop equipped with Windows 10, intel i7-3517U processor and 10 GB of memory. We ran the i2b2 Web-client on Firefox 47.0.1.

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

TABLE 3: Server Setting

We used the implementation of the FV cryptosystem provided within the NFLlib library by Aguilar-Melchor *et al.* [46]. The NFLlib is an optimized open-source C++ library dedicated to ideal lattice cryptography in the polynomial ring $\mathbb{Z}_q[x]/(x^l+1)$ for l a power of 2. We chose this library because, to the best of our knowledge, it is the most efficient one for computations over polynomials. Indeed, NFLlib uses a mixed NTT-CRT representation to reduce computational costs: Number-Theoretic Transform (NTT) for polynomials [47] and Chinese Remainder Theorem (CRT) for their coefficients.

We used the FV encryption parameters, as reported in Table 4, in order to have 128 bits of security level.

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

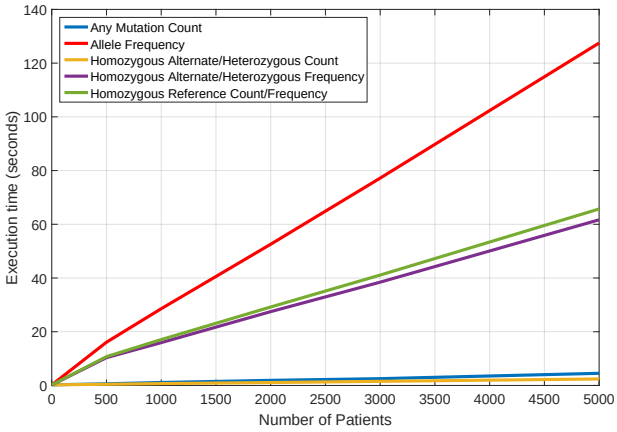
TABLE 4: Encryption Parameters

We used real genomic data coming from the exome sequencing of 392 samples giving a genotyping for 472,845 variants each. The resulting VCF file contained a total of 185,355,240 unphased genotypes. For clinical data, we used 90,454 clinical records from 134 patients available from the i2b2 demo version [48]. Patients from the i2b2 demo were duplicated in order to match the number of patients with genomic records. As a result, we had an initial cohort of 392 individuals with both clinical and genomic data. To test the scalability of our solution, this initial cohort was further extended by replicating individuals in order to obtain a cohort of 5,000 patients.

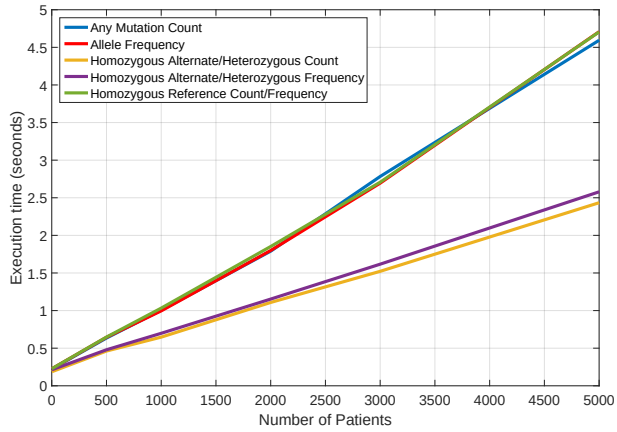
Performance Analysis. We assessed the performance of our proposed solution in terms of storage and computational overheads.

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

To measure the computational overhead, we ran experiments on all the privacy-preserving algorithms for summary statistics for different sets of variants and different cohort sizes. Experiments were run 100 times for each scenario and we report the average execution time in the following. We evaluated the different steps of the query execution phase, described in detail in Section 4.4.



(a) With *no-calls*.



(b) Without *no-calls*.

Fig. 3: Total query execution time at i2b2 server per number of patients in the cohort for a query involving 3000 genetic variants.

Fig.3a and Fig.3b show the total execution time at the i2b2 server needed to compute the different summary statis-

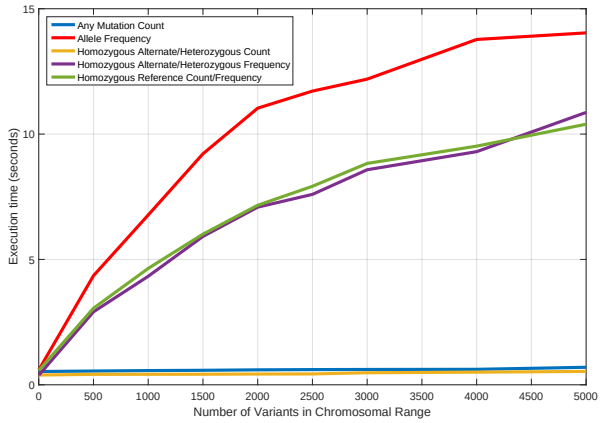
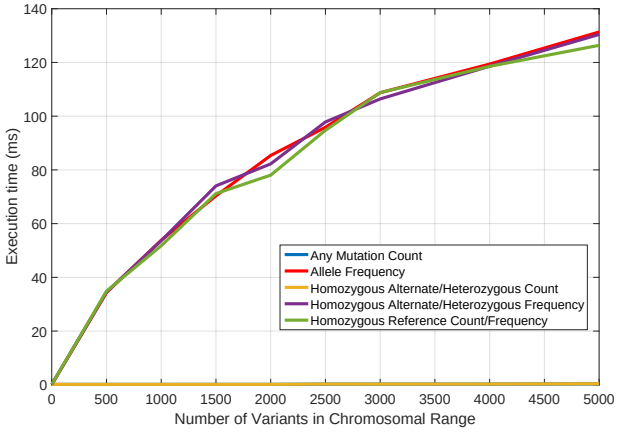


Fig. 4: Total execution time at i2b2 server per number of consecutive genetic variants.

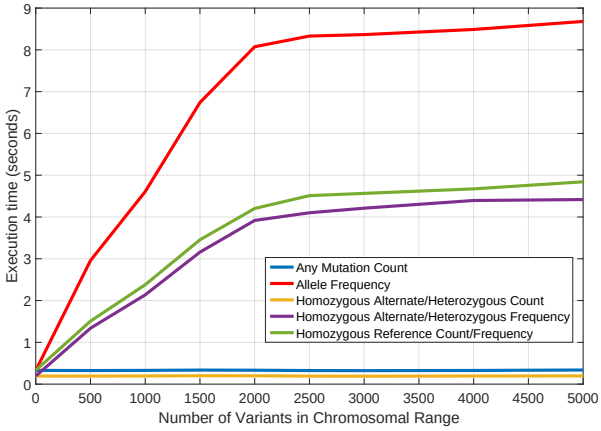
tics for increasing cohort sizes and a fixed query including 3,000 genetic variants with and without *no-calls*. It is easy to observe how the execution time increases linearly with the number of patients in the cohort and how the presence of *no-calls* in the VCF file has a significant impact on performance.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

The execution time at the i2b2 does not depend only on the number of patients in the cohort but also on the number of consecutive genetic variants specified in the query, as shown in Fig.4. The differences in execution time between the different summary statistics depend on genotypes with no-call values. In particular, the computations of the number of mutations (blue curve) and the number of homozygous alternate/heterozygous (yellow curve) are significantly influenced only by the existence of genotypes with 1 no-call, whereas the other computations are also influenced by genotypes with 2 no-calls. Note that in our data we do not have genotypes with 1 no-call. From Fig.5a and Fig.5b we can observe that most of the computational time at the i2b2 server is due to data retrieval from the database and homomorphic computations.

Finally, Fig.6a and Fig.6b show the execution time for a full decryption of the query results for an increasing number of consecutive genetic variants for the contribution of the proxy server and the client, respectively. It is easy to observe that the execution time at the proxy server is similar to the execution time at the i2b2 server for partial decryption as the number of processed ciphertexts is the same. At the client side, we can observe a different behavior for the execution time as decryption is done variant-by-variant instead of ciphertext-by-ciphertext. Note that, because of genotypes with no-call values, the number of ciphertexts including the query results can be different from the number of ciphertexts including genetic variants.



(a) Partial decryption.



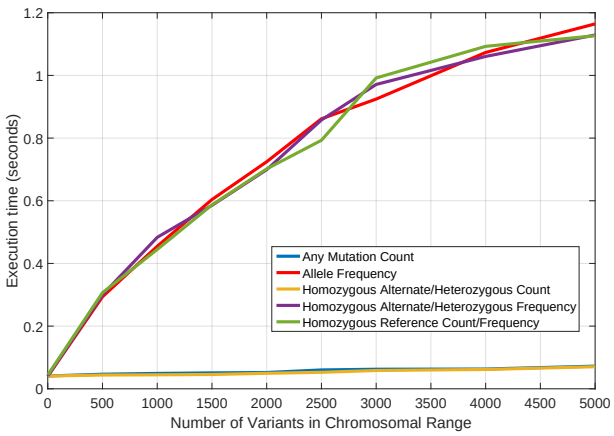
(b) Data retrieval and homomorphic computations.

Fig. 5: Breakdown of total execution time at i2b2 server per number of consecutive genetic variants.

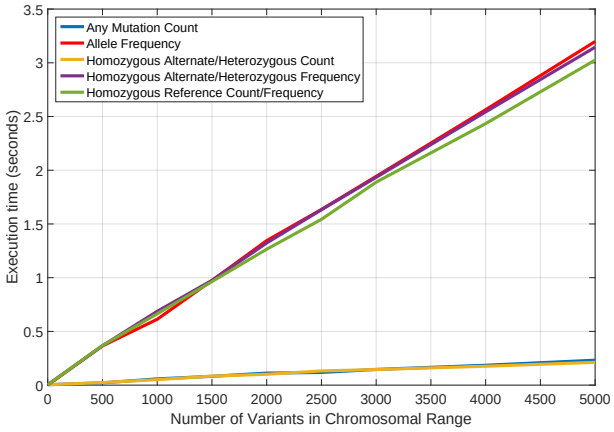
6 Discussion

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

Performance. As it can be observed from the results of Fig. 3a and Fig. 3b, the main bottleneck for the execution time of queries involving specific types of summary statistics (e.g., allele or genotype frequencies) is due to the presence of genotypes with *no-call* values. Indeed, the number of key-value pairs (i.e., set of variants that can be processed in parallel) generated by Algorithms 1, 4 and 5 at the i2b2 server can significantly grow if the distribution of *no-calls* is very different among patients. Yet, some quick alternative approaches can be used to easily address this issue. A first potential approach consists in using a different encoding for genotype values, as the one shown in Table 5, which maximizes performance at the expense of increasing the storage overhead from a factor of 8 to a factor of 20. Note that a storage overhead of 20x can be prohibitive for most



(a) At proxy server.



(b) At client.

Fig. 6: Execution time for full decryption for queries with increasing number of consecutive genetic variants.

institutions in case of large studies such as whole genome sequencing. It is definitely acceptable for studies on the exome. Another potential approach would be to perform genotype imputation before the genotype encoding in order to replace *no-calls* with imputed values at the expense of a slight decrease in accuracy. This said, it is easy to observe how the first alternative approach prioritizes high performance and high accuracy instead of low storage overhead, whereas the second approach ensures high performance and low storage overhead rather than full accuracy. Note that, by slightly sacrificing performance, our current solution assigns the highest priority to low storage cost and high accuracy as specified by CHUV's requirements (see Section 4.3). We leave for future work further investigations on how to improve our secure algorithms described in the Supplementary Materials in order to optimize both performance and storage without sacrificing accuracy.

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

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

TABLE 5: Genotype encoding optimizing performance.

Query result perturbation. As explained in Section 4.4, our solution applies the standard and well-established Laplace mechanism [37] to independently perturb query results in order to satisfy the notion of differential privacy and prevent patient re-identification. Yet, the amount of noise that the i2b2 server needs to add to new queries of a given user grows linearly with the number of queries already answered for that user. This can substantially degrade the utility of the system as the results of later queries would be useless or, in other words, the number of useful queries would be limited. For this reason more sophisticated mechanisms could be used to obtain sublinear noise. For example, the *median* mechanism [49], the *exponential* mechanism [50] or the *multiplicative weights* mechanism [51] can answer exponentially more predicate queries than the Laplace mechanism. Indeed, the algorithm proposed by Vinterbo *et al.* [52] provides the option to incorporate user preferences with regard to individual query responses, thereby increasing utility to users without compromising privacy. The authors propose as well an evaluation of the privacy/utility trade-off with i2b2 which shows the efficacy of their method. This can be easily implemented in our system.

7 Conclusion

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

To conclude, we acknowledge that the proposed solution addresses a simple use case by providing genomic summary statistics that can be securely computed through homomorphic additions. Yet, thanks to the flexibility of the FV scheme, more complex use cases such as privacy-preserving phenome-wide association studies (PheWAS) or GWAS can be envisioned and be developed on top of our solution.

References

[1] Y. Erlich and A. Narayanan, "Routes for breaching and protecting genetic privacy," *Nature Reviews Genetics*, vol. 15, no. 6, pp. 409–421, 2014.

[2] S. N. Murphy, G. Weber, M. Mendis, V. Gainer, H. C. Chueh, S. Churchill, and I. Kohane, "Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2)," *Journal of the American Medical Informatics Association*, vol. 17, no. 2, pp. 124–130, 2010.

[3] "i2b2 installations." [Online]. Available: https://www.i2b2.org/work/i2b2_installations.html

[4] S. N. Murphy, V. Gainer, M. Mendis, S. Churchill, and I. Kohane, "Strategies for maintaining patient privacy in i2b2," *Journal of the American Medical Informatics Association*, vol. 18, no. Supplement 1, pp. i103–i108, 2011.

[5] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genetics*, vol. 4, no. 8, 2008.

[6] S. S. Shringarpure and C. D. Bustamante, "Privacy risks from genomic data-sharing beacons," *The American Journal of Human Genetics*, vol. 97, no. 5, pp. 631–646, 2015.

[7] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science*, vol. 339, no. 6117, pp. 321–324, 2013.

[8] M. Humbert, K. Huguenin, J. Hugonot, E. Ayday, and J.-P. Hubaux, "De-anonymizing genomic databases using phenotypic traits," in *15th Privacy Enhancing Technologies Symposium (PETS)*, 2015.

[9] B. Malin and L. Sweeney, "How (not) to protect genomic data privacy in a distributed network: Using trail re-identification to evaluate and design anonymity protection systems," *Journal of Biomedical Informatics*, vol. 37, no. 3, pp. 179–192, 2004.

[10] R. Wang, Y. F. Li, X. Wang, H. Tang, and X. Zhou, "Learning your identity and disease from research papers: Information leaks in genome wide association study," in *CCS*, 2009, pp. 534–544.

[11] X. Zhou, B. Peng, Y. F. Li, Y. Chen, H. Tang, and X. Wang, "To release or not to release: Evaluating information leaks in aggregate human-genome data," in *ESORICS*, 2011.

[12] M. Kantarcioglu, W. Jiang, Y. Liu, and B. Malin, "A cryptographic approach to securely share and query genomic sequences," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 5, pp. 606–617, 2008.

[13] M. Canim, M. Kantarcioglu, and B. Malin, "Secure management of biomedical data with cryptographic hardware," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 1, pp. 166–175, 2012.

[14] L. Kamm, D. Bogdanov, S. Laur, and J. Vilo, "A new way to protect privacy in large-scale genome-wide association studies," *Bioinformatics*, 2013.

[15] W. Xie, M. Kantarcioglu, W. S. Bush, D. Crawford, J. C. Denny, R. Heatherly, and B. A. Malin, "SecureMA: protecting participant privacy in genetic association meta-analysis," *Bioinformatics (Oxford, England)*, 2014.

[16] X. S. Wang, Y. Huang, Y. Zhao, H. Tang, X. Wang, and D. Bu, "Efficient genome-wide, privacy-preserving similar patient query based on private edit distance," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 492–503.

[17] W. Lu, Y. Yamada, and J. Sakuma, "Efficient secure outsourcing of genome-wide association studies," in *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 2015, pp. 3–6.

[18] D. A. Duverle, S. Kawasaki, Y. Yamada, J. Sakuma, and K. Tsuda, "Privacy-preserving statistical analysis by exact logistic regression," in *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 2015, pp. 7–16.

[19] S. Wang, X. Jiang, D. Fox, and L. Ohno-Machado, "Preserving genome privacy in research studies," in *Medical Data Privacy Handbook*. Springer, 2015, pp. 425–441.

[20] S. Wang, Y. Zhang, W. Dai, K. Lauter, M. Kim, Y. Tang, H. Xiong, and X. Jiang, "Healer: Homomorphic computation of exact logistic regression for secure rare disease variants analysis in gwas," *Bioinformatics*, vol. 32, no. 2, pp. 211–218, 2016.

[21] C. Uhlerop, A. Slavković, and S. E. Fienberg, "Privacy-preserving data sharing for genome-wide association studies," *The Journal of privacy and confidentiality*, vol. 5, no. 1, p. 137, 2013.

[22] A. Johnson and V. Shmatikov, "Privacy-preserving data exploration in genome-wide association studies," in *KDD*, 2013, pp. 1079–1087.

[23] F. Yu, S. E. Fienberg, A. B. Slavkovic, and C. Uhler, "Scalable privacy-preserving data sharing methodology for genome-wide association studies," *Journal of Biomedical Informatics*, 2014.

[24] S. Simmons and B. Berger, "One size doesn't fit all: Measuring individual privacy in aggregate genomic data," in *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 2015, pp. 41–49.

[25] ——, "Realizing privacy preserving genome-wide association studies," *Bioinformatics*, vol. 32, no. 9, pp. 1293–1300, 2016.

[26] S. Simmons, C. Sahinalp, and B. Berger, "Enabling privacy-preserving gwass in heterogeneous human populations," *Cell Systems*, vol. 3, no. 1, pp. 54 – 61, 2016.

[27] F. Tramèr, Z. Huang, J.-P. Hubaux, and E. Ayday, "Differential privacy with bounded priors: Reconciling utility and privacy in genome-wide association studies," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1286–1297.

[28] P. J. McLaren, J. L. Raisaro, M. Aouri, M. Rotger, E. Ayday, I. Bartha, M. B. Delgado, Y. Vallet, H. F. Günthard, M. Cavassini, H. Furrer, T. Doco-Lecompte, C. Marzolini, P. Schmid, C. Di Benedetto, L. A. Decosterd, J. Fellay, J.-P. Hubaux, A. Telenti, and t. S. H. C. Study, "Privacy-preserving genomic testing in the clinic: a model using HIV treatment," *Genetics in Medicine*, vol. 18, no. 8, pp. 814–822, aug 2016. [Online]. Available: http://www.nature.com/doifinder/10.1038/gim.2015.167

[29] L. C. Phillips, S. Minovitsky, I. Ratnere, I. Dubchak, I. Kohane, and S. Murphy, "Use genomic variants in informatics for integrating biology and the bedside (i2b2)," *AMIA Proceedings Library: 48-52 T2011*, 2011.

[30] M. Gabetta, I. Limongelli, E. Rizzo, A. Riva, D. Segagni, and R. Bellazzi, "Bigq: a nosql based framework to handle genomic variants in i2b2," *BMC bioinformatics*, vol. 16, no. 1, p. 1, 2015.

[31] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[32] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *IACR Cryptology ePrint Archive*, vol. 2012, p. 144, 2012. [Online]. Available: http://eprint.iacr.org/2012/144

[33] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[34] J. W. Bos, K. E. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," in *Cryptography and Coding - 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings*, 2013, pp. 45–64.

[35] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in lwe-based homomorphic encryption," in *Public-Key Cryptography–PKC 2013*. Springer, 2013, pp. 1–13.

[36] C. Dwork, "Differential privacy." Venice, Italy: Springer Verlag, July 2006. [Online]. Available: https://www.microsoft.com/en-us/research/publication/differential-privacy/

[37] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*. Springer, 2006, pp. 265–284.

[38] "The variant call format specification - vcfv4.3 and bcfv2.2," http://samtools.github.io/hts-specs/VCFv4.3.pdf, accessed: 2016-08-02.

[39] J. Schlesinger, "Dark web is fertile ground for stolen medical records." [Online]. Available: http://www.cnbc.com/2016/03/10/dark-web-is-fertile-ground-for-stolen-medical-records.html

[40] G. A. for Genomics and Health, "The beacon project." [Online]. Available: https://beacon-network.org/#/

[41] "Exac browser." [Online]. Available: http://exac.broadinstitute.org

[42] M. Namazi, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Dynamic privacy-preserving genomic susceptibility testing," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2016, pp. 45–50.

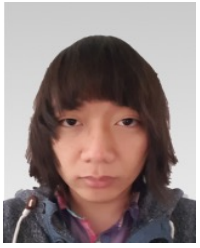
[43] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[44] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[45] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 106–115.

[46] C. Aguilar-Melchor, J. Barrier, S. Guelton, A. Guinet, M.-O. Killijian, and T. Lepoint, "Nfllib: Ntt-based fast lattice library," in *Cryptographers' Track at the RSA Conference*. Springer, 2016, pp. 341–356.
[47] N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. Huss, "On the design of hardware building blocks for modern lattice-based encryption schemes," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2012, pp. 512–529.
[48] "i2b2 software." [Online]. Available: https://www.i2b2.org/software/index.html
[49] A. Roth and T. Roughgarden, "Interactive privacy via the median mechanism," in *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM, 2010, pp. 765–774.
[50] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*. IEEE, 2007, pp. 94–103.
[51] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for privacy-preserving data analysis," in *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*. IEEE, 2010, pp. 61–70.
[52] S. A. S. Vinterbo, A. D. A. Sarwate, and A. A. A. A. Boxwala, "Protecting count queries in study design." *Journal of the American Medical Informatics Association : JAMIA*, vol. 19, no. 5, pp. 750–7, 2012.

Jean Louis Raisaro earned his PhD in Computer and Communication Sciences in 2018 from EPFL, Lausanne, Switzerland. Prior to that he obtained a MS and BS in Biomedical Informatics and Bioinformatics in 2012 and 2009 from University of Pavia, Pavia, Italy. His main interests are the design and development of new efficient privacy-enhancing technologies for the protection of medical data with a special focus on genetic data. He is an expert in applied cryptography, privacy and medical informatics.



Gwangbae Choi received M.Sc. degree in communication systems from École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland in 2016. He is currently pursuing his Ph.D. in cryptography at École Polytechnique Fédérale de Lausanne under the supervision of Prof. Vaudenay.

Sylvain Pradervand received a Ph.D. degree in molecular biology from the University of Lausanne in 1998. After a postdoc studying transcriptomics in heart disease models at the University of California San Diego, he turned his interests to bioinformatics. He is currently leading the bioinformatics team of the genomic technologies facility of the University of Lausanne and the bioinformatics team of the clinical research support platform of the Lausanne University Hospital.

Raphael Colsenet received M.Sc. at the University of Montreal (1997). He has more than 16 years of experience in Business Intelligence including analytical tools, data viz, data modeling, BI architecture, MDM and Data governance in various industries in North America and Europe.

Nathalie Jacquemont is the Operational Director of the Lausanne University Hospital (CHUV) Clinical Research Support Entity in charge of the CHUV Clinical Research IT Unit, the General Consent Unit, the Research Pre-analytical Laboratory Unit and the Institutional Biobank of Lausanne. She contributes to support CHUV's Clinical Research projects deploying processes and guidelines to obtain patients agreement.

Nicolas Rosat received a B.S. in Mathematics, a M.S. in Business Information Systems and a M.S. in Health Sciences and Organisation from the University of Lausanne in 1988, 1998, 2014 respectively. He is currently head of IT Governance at Lausanne University Hospital (CHUV). He is involved in many Swiss initiatives at the interplay of clinical research, bio-banking and personalized health with main interests in business intelligence and data sciences.

Vincent Mooser MD is board certified in internal medicine, with a particular interest in laboratory medicine, genomics and pharmaceutical sciences. Since 2011, he chairs the Lab Department and is responsible for the development of precision medicine at CHUV University Hospital. Before this appointment, he was vice-president in charge of Applied Genetics at GlaxoSmithKline, a pharmaceutical company.

Jean-Pierre Hubaux is a full professor at EPFL. Through his research, he contributes to laying the foundations and developing the tools for protecting privacy in tomorrow's hyper-connected world. He has pioneered the areas of privacy and security in mobile/wireless networks and in genomics. He is a Fellow of both IEEE (2008) and ACM (2010).