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

Jean Louis Raisaro, Juan Ramón Troncoso-Pastoriza, Mickaël Misbach, João Sá Sousa, Sylvain Pradervand, Edoardo Missiaglia, Olivier Michielin, Bryan Ford, and Jean-Pierre Hubaux

Abstract—The increasing number of health-data breaches is creating a complicated environment for medical-data sharing and, consequently, for medical progress. Therefore, the development of new solutions that can reassure clinical sites by enabling privacy-preserving sharing of sensitive medical data in compliance with stringent regulations (e.g., HIPAA, GDPR) is now more urgent than ever. In this work, we introduce MedCo, the first operational system that enables a group of clinical sites to federate and collectively protect their data in order to share them with external investigators without worrying about security and privacy concerns. MedCo uses (a) collective homomorphic encryption to provide trust decentralization and end-to-end confidentiality protection, and (b) obfuscation techniques to achieve formal notions of privacy, such as differential privacy. A critical feature of MedCo is that it is fully integrated within the i2b2 (Informatics for Integrating Biology and the Bedside) framework, currently used in more than 300 hospitals worldwide. Therefore, it is easily adoptable by clinical sites. We demonstrate MedCo's practicality by testing it on data from The Cancer Genome Atlas in a simulated network of three institutions. Its performance is comparable to the ones of SHRINE (networked i2b2), which, in contrast, does not provide any data protection guarantee.

Index Terms—Secure data-sharing, homomorphic encryption, differential privacy, i2b2, distributed data, decentralized trust, genomic privacy

1 Introduction

With the increasing digitalization of clinical and genomic information, data sharing is becoming the keystone for realizing the promise of personalized medicine. Several initiatives, such as the Patient-Centered Clinical Research Network (PCORNet) [1] in the USA, eTRIKS/TranSMART [2] in the EU, the Swiss Personalized Health Network (SPHN) [3] in Switzerland, and the Global Alliance for Genomics and Health (GA4GH) [4], are laying down the foundations for new biomedical research infrastructures aimed at interconnecting (so far) siloed repositories of clinical and genomic data. In this global ecosystem, the ability to provide strong privacy and security guarantees in order to comply with increasingly strict regulations (e.g., HIPAA [5] in USA or the new GDPR [6] in EU) is crucial, yet extremely challenging, to achieve.

Currently, there exist two main approaches for sharing medical data. The first is the centralized approach (see Fig. 1 A) typical of initiatives such as *All of Us* [7] and *Genomics England* [8]. With this approach, data from multiple institutions are brought together in a single and centralized repository that can be accessed by researchers willing to run analysis on a unified dataset. The second is the decentralized approach (see Fig. 1B), where the different institutions keep the data at their premises and form an interoperable peer-to-peer network accessible by researchers. PCORNet [1] and the Beacon Project of the GA4GH [9] are examples of this second approach. Unfortunately, both approaches to sharing medical data have revealed intrinsic limitations that demonstrate why neither of the two has already been fully adopted by the healthcare sector.

On the one hand, the centralized approach provides undeniable advantages in terms of availability and flexibility, although it introduces a single point of failure in the system by accumulating all the trust on a single entity (i.e., the data repository). Indeed, the security and confidentiality of all the data rely on the ability of the central repository to thwart both external (hackers) and internal (insiders) attacks. Furthermore, as the number of health-data breaches constantly increases [10], there is significant public pressure on clinical sites to ensure that the privacy and security of patients' data can be properly protected, notably when stored or processed by third parties. As a result, clinical sites are worried about adopting the centralized approach and

- *J.L. Raisaro, J.R. Troncoso-Pastoriza, M. Misbach, J. Sá Sousa, B. Ford, and J.-P. Hubaux are with the School of Computer and Communication Sciences, EPFL, Lausanne 1015, Switzerland. E-mail: jeanlouis.raisaro@gmail.com, {juan.troncoso-pastoriza, mickael.misbach, joao.gomesdesaesousa, bryan.ford, jean-pierre.hubaux}@epfl.ch.*
- *S. Pradervand is with the Lausanne University Hospital, CHUV, Lausanne 1011, Switzerlandthe Genomic Technologies Facility, University of Lausanne, UNIL, Lausanne 1015, Switzerland, and the Vital-IT, Swiss Institute of Bioinformatics, Lausanne 1015, Switzerland. E-mail: Sylvain.Pradervand@chuv.ch.*
- *E. Missiaglia and O. Michielin are with the Lausanne University Hospital, CHUV, Lausanne 1011, Switzerland. E-mail: {edoardo.missiaglia, olivier.michielin}@chuv.ch.*

Manuscript received 11 June 2018; accepted 21 June 2018. Date of publication 13 July 2018; date of current version 5 Aug. 2019.
(Corresponding author: Jean Louis Raisaro.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2018.2854776

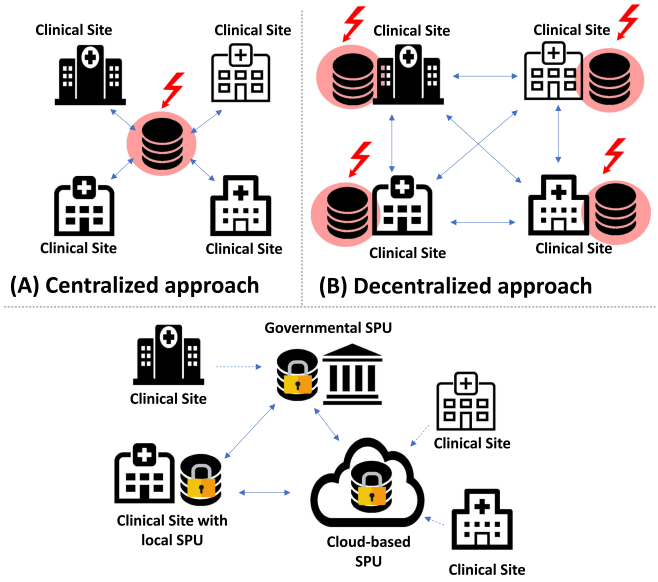


Fig. 1. Comparison of approaches for sharing medical data. (A) Centralized approach affected by the single-point-of-failure problem. (B) Decentralized approach affected by high maintenance costs (both technical and human). (C) Hybrid and secure approach enabled by MedCo, where clinical sites can securely outsource their data to the storage and processing unit (SPU) of their choice.

outsourcing their data to a single central repository (e.g., the cloud), especially when the data to be shared is highly sensitive or identifying (e.g., genomic data). On the other hand, the fully decentralized approach solves the single-point-of-failure issue: clinical sites can individually enforce local control on their own data by monitoring and managing the different accesses. However, this decentralization imposes substantial costs on the clinical sites, as they have to maintain an interoperable network, often with very limited resources (both human and technical). For this reason, the fully decentralized approach is also likely to be unsustainable in the long run, especially for large scale projects where multiple clinical sites are involved.

In this paper, to address the challenge of achieving privacy-preserving, secure and scalable data sharing we introduce MedCo. MedCo is the first operational system that enables hundreds of clinical sites to share their clinical and genomic data through a hybrid or "somewhat" decentralized approach that overcomes the limitations of the approaches described above (see Fig. 1). Instead of concentrating the trust on a single central repository as in the centralized approach, MedCo distributes the trust among a set of different "storage and processing" units to which clinical sites can securely outsource the storage of their data. Together, the storage and processing units form a secure, federated and interoperable network that investigators can query for research purposes as if it were a single unified database. MedCo enables each clinical site to choose its preferred storage and processing unit in order to offload the maintenance and availability costs that affect the fully decentralized approach. Such a storage and processing unit can be hosted either by the clinical site itself, by a governmental institution, or by a private/public cloud provider with whom the clinical site establishes a data-use agreement. For example, a clinical site with enough resources can have its own storage and processing unit hosted at its premises. Whereas, a clinical site with limited resources could use a cloud provider of its choice. Potentially, each country could have a national storage and processing unit, e.g., administered by the government or a not-profit organization, to which all clinical sites within the same country can outsource their data. The different national storage and processing units could then federate to form an international, secure and distributed clinical research network.

A critical advantage of MedCo, with respect to state-of-the-art systems for sharing medical data, is its ability to provide strong security guarantees to clinical sites willing to safely outsource the storage of their data to potentially untrusted storage and processing units. Indeed, MedCo enables each site to encrypt its data with a shared key that is collectively generated by all the storage and processing units in the federation. As the encryption scheme used by MedCo is additively homomorphic, investigators can directly query and process the encrypted data stored at different storage and processing units without the need for decrypting them. This ensures end-to-end protection of the data in the *Anytrust* adversary model. Only authorized investigators can decrypt the result of a query/analysis and none of the storage and processing units alone, even if compromised, can decrypt the data stored at its premises. Actually, in order to succeed and get access to the unencrypted data, an adversary would need to simultaneously compromise all the storage and processing units in the federation. Additionally, MedCo can also be configured to minimize the risk of re-identification stemming from the behavior of malicious or curious investigators that try to abuse the querying system; this is achieved by providing obfuscated results that provide formal and well-established notions of privacy, e.g., differential privacy.

In order to ease its adoption in operational research environments, we developed MedCo on top of existing and well-established open-source technologies for clinical data exploration, namely i2b2 [11] and SHRINE [12]. Currently, i2b2 is used at more than 300 clinical sites worldwide. We demonstrate the practicality of MedCo by testing it in a simulated federation of three clinical sites that outsource their oncology data (both clinical and genomic) to three different storage and processing units. We compare MedCo with a standard deployment, based on i2b2 and SHRINE (that does not provide any data protection guarantee) and we show that MedCo's performance overhead is practical.

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

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

- We introduce MedCo, the first operational system enabling the sharing of sensitive clinical and genomic information in a privacy-preserving, secure and scalable way.
- We developed MedCo to be fully compatible with state-of-the-art clinical research platforms such as i2b2 and SHRINE, hence it can be seamlessly deployed by clinical sites.
- We extensively tested MedCo in a simulated federation of three sites, focusing on a clinical-oncology case with tumor DNA data from The Cancer Genome Atlas, and we demonstrated its practicality.
- We propose a new generic method to add dummy data in order to mitigate frequency attacks that can target the probabilistically encrypted data after they are transformed to deterministically encrypted data for the sake of enabling equality-matching queries.

2 Related Work

Among the operational systems for sharing clinical or genomic information, SHRINE [12] (the networked version of i2b2 [11]) and the GA4GH Beacon Network [13] are certainly the most advanced and widespread. For example, SHRINE is used in several PCORNet clinical data research networks. However, as opposed to MedCo, they provide limited privacy guarantees (restricted to ad-hoc result obfuscation) and no protection of data confidentiality besides standard access control, thus significantly restraining the possibility of outsourcing the storage and of processing of the data to external parties in order to partially offload the costs of maintaining an always-available interoperable network. SHRINE provides an ad-hoc mechanism for obfuscating query results and for locking-out investigators after a certain number of queries, whereas MedCo features a privacy-budget mechanism that achieves differential privacy. Conversely, the Beacon still suffers from risk of re-identification, as none of the three practical strategies described in [14] has been implemented yet.

To the best of our knowledge, there are two recent works dealing with privacy-preserving queries in distributed medical databases; they represent the two main alternatives to the encryption-based approach followed in this work: The first one, PRINCESS [15], is based on trusted hardware: The sites encrypt all their data under Advanced Encryption Standard - Galois Counter Mode (AES-GCM) and send them to an enclave that runs in a central server, featuring an Intel SGX processor; this server decrypts and processes the sensitive data thus, enabling the secure computation of statistical models. Compared to our work, PRINCESS can be more versatile in terms of allowed computations, but it presents a single point of failure (the central server), and it centralizes all trust in the enclave and in the attestation protocol provided by Intel. Furthermore, the memory restrictions of the enclave limit the scalability of the scheme, requiring compression and batching techniques to enable processing of large genomic data, for which MedCo scales much better.

The other recent approach, SMCQL [16], is based on secure two-party computation; it introduces a framework for private data network queries on a federated database of mutually distrustful parties. SMCQL features a secure query executor that implements different types of queries (e.g., merge, join, distinct) on the distributed database by relying on garbled circuits and Oblivious RAM (ORAM) techniques. Whereas this work features truly decentralized trust, it does not scale well to scenarios with more than two sites that are typical in medical contexts with a high number of collaborating hospitals.

3 Preliminaries

In this section, we briefly introduce the main cryptographic concepts used throughout the paper.

3.1 Deterministic Encryption

Deterministic encryption (DTE) [17] is a special type of encryption that preserves the equality property of the plaintexts that, as opposed to probabilistic encryption, makes ciphertexts indistinguishable and, a priori, unusable. Yet, DTE also leaks this property; for a given plaintext and key, DTE always produces the same ciphertext. More formally, for $A, B \subseteq \mathbb{Z}$ with $|A| \leq |B|$, a function $f : A \rightarrow B$ is *equality-preserving* if for all $i, j \in A$, $f(i) = f(j)$ iff $i = j$. We say that an encryption scheme with plaintext and ciphertext spaces $\mathcal{D}$ and $\mathcal{R}$, respectively, is deterministic if $\mathrm{E}_{\mathrm{DTE}}(K, \cdot)$ is an equality-preserving function from $\mathcal{D}$ to $\mathcal{R}$ for all $K \in \mathcal{K}$ (where $\mathcal{K}$ is the key space).

DTE-based schemes have several advantages and are mainly used in the context of encrypted database systems (e.g., CryptDB [18]) as they enable relational databases to perform equality searches on encrypted data in the same way as they would operate on the plaintext data. As a counterpart, they provide less security guarantees than probabilistic encryption schemes, as they are vulnerable to inference attacks due to the amount of information they leak. Hence, their application has to be carefully assessed.

3.2 Homomorphic Encryption

Homomorphic encryption (HE) is a special type of encryption that supports computation on encrypted data. Homomorphic encryption is probabilistic and provides semantic security, meaning that no adversary without the secret key can compute any function of the plaintext from the ciphertext. In 2009, Gentry [19] introduced for the first time a special type of HE that enables *arbitrary computations* on ciphertexts, called fully homomorphic encryption (FHE).

Despite its complete functionality, FHE is currently unpractical, as it introduces huge 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 improving efficiency by sacrificing some flexibility. Such cryptosystems are called *practical* homomorphic cryptosystems, and according to their functionality, they can be classified as *additively* homomorphic if they satisfy only the addition of ciphertexts, *multiplicatively* homomorphic if they satisfy only multiplication, or *somewhat* homomorphic if they support (a limited number of) additions and multiplications.

In this paper, we use the additively homomorphic cryptosystem ElGamal on Elliptic Curves, due to its low ciphertext expansion and fast homomorphic operations.

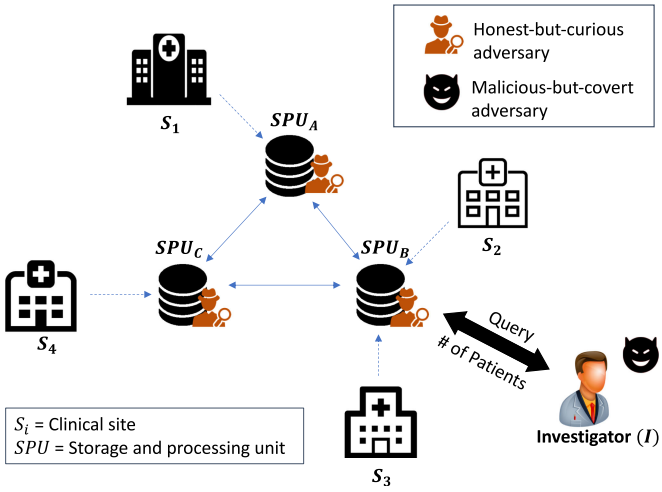


Fig. 2. MedCo's system and threat models.

3.2.1 ElGamal on Elliptic Curves

The ElGamal cryptosystem on elliptic curves (EC-ElGamal) is an asymmetric, probabilistic and additively-homomorphic encryption scheme that achieves semantic security, i.e., ciphertext indistinguishability. It enables additions and multiplications by constants in the ciphertext domain. As every asymmetric cryptosystem, EC-ElGamal features three algorithms:

- *Key generation:* Let $\mathcal{E}$ denote an elliptic curve over the prime field $\mathbb{GF}(p)$ and G its base point. Then, the secret key can be defined as an integer $k \in \mathbb{GF}(p)$, and the public key can be derived as $K = kG$.
- *Encryption:* Let m be an integer and $M = mG$ its mapping to the corresponding point on the curve $\mathcal{E}$. Then, the encryption of M with the public key K is denoted as $E_K(M) = (C_1, C_2) = (rG, M + rK)$, where r is a random nonce.
- *Decryption:* Given the ciphertext $E_K(M) = (C_1, C_2)$ and the secret key k, the decryption algorithm computes the original plaintext point as $D(E_K(M)) = -kC_1 + C_2 = M$. The original plaintext m is obtained by inverting the mapping from the elliptic curve point M.

Due to its additive homomorphism, EC-ElGamal enables combining the encryptions of any two messages in order to obtain an encrypted result that, when decrypted, equals the sum of these two messages. More formally, let M_1 and M_2 be any two messages, and α and β be two scalars; then, we have that $\alpha E_K(M_1) + \beta E_K(M_2) = E_K(\alpha M_1 + \beta M_2)$.

4 MedCo Ecosystem

In this section, we introduce the ecosystem in which MedCo operates. We begin by describing the system and threat models. We then define the goals of MedCo with respect to privacy/security and functionality.

4.1 System Model

We consider the system model depicted in Fig. 2, where several clinical sites (S_i) want to collaborate in order to share clinical and genomic data with investigators, but do not want to rely on any central third party or authority for storing or managing their data. Moreover, because of the high costs (both technical and human) for maintaining a fully interoperable decentralized network and the increasing size of the data, clinical sites want to securely outsource the storage of their data to a preferred storage and processing unit (SPU_j). Each site can have its own SPU, or multiple sites can share the same SPU. All SPUs are organized together in a peer-to-peer network and form a collective authority. SPUs are responsible for (i) securely storing the data of the clinical sites and (ii) securely processing a request of an authorized investigator that wants to explore clinical sites' data for generating and validating new research hypotheses or for identifying cohorts of interest, by finding the patients that match specific inclusion/exclusion clinical and genetic criteria across the whole network.

4.2 Threat Model

In this system model, we consider the following threats:

- *Storage and processing units*: We assume storage and processing units to be *honest-but-curious (HBC)* parties. Indeed, SPUs can be compromised by internal or external adversaries that do not tamper with the data-sharing protocol but can try to infer sensitive information about the patients from the data stored at their premises and from the data being processed during the protocol itself. As a result, SPUs cannot be trusted by clinical sites and they do not trust each other, either.
- *Investigators*: We assume investigators to be potentially *malicious-but-covert (MBC)* adversaries. Indeed, an investigator can try to legitimately use the system in order to infer sensitive information about the patients (without being discovered) by performing consecutive queries and exploiting the information leaked by the end-results. For example, a malicious investigator with some background information about a given individual can infer the presence of such individual into a sensitive cohort (e.g., patients who are HIV-positive) or even reconstruct a subset of her medical record.
- *Clinical sites*: We assume clinical sites to be *trusted* parties.

Finally, we assume that investigators cannot collude with SPUs, and that at least one SPU does not collude with the others.

4.3 MedCo's Goals

To meet end-users expectations and be compliant with regulations, MedCo has the following goals with respect to functionality and privacy/security features.

4.3.1 Functionality Goals

The purpose of MedCo is to enable investigators to securely explore the clinical and genomic data stored at all SPUs by the various clinical sites in the network. Therefore, MedCo must provide the same functionalities as those provided by state-of-the-art distributed cohort explorers such as SHRINE [12]:

- *(F1) Cohort Exploration*: An authorized investigator should be able to obtain the number of patients per

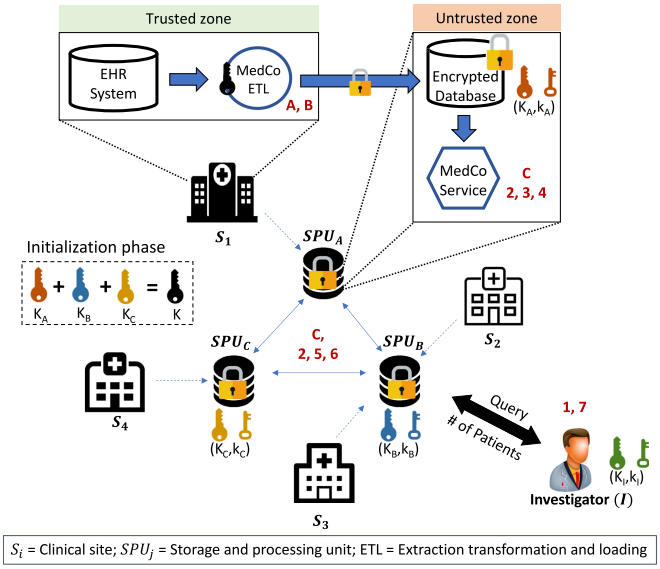


Fig. 3. MedCo core architecture and secure query protocol comprising of: ETL process (steps A, B, and C); query generation (step 1); query re-encryption (step 2); local query processing (step 3); local result obfuscation (step 4); distributed results shuffling (step 5); distributed results re-encryption (steps 6); and results decryption (step 7).

clinical site who satisfy a set of inclusion/exclusion clinical and genetic criteria, optionally grouped by age, gender or ethnicity. More formally, MedCo must support SQL queries such as

```
SELECT COUNT(patients)
FROM distributed_dataset
WHERE criteria_i AND/OR criteria_j
AND/OR ...
GROUP BY criteria_k;
```

- *(F2) Cohort Selection*: An authorized investigator should be able to obtain the pseudonyms of the patients who satisfy a set of inclusion/exclusion clinical and genetic criteria at each clinical site. More formally, MedCo must support SQL queries such as

```
SELECT patients
FROM distributed_dataset
WHERE criteria_i AND/OR criteria_j
AND/OR ...;
```

4.3.2 Security and Privacy Goals

MedCo must always provide the following privacy/security features:

- *(SP1) Trust Decentralization*: There should be no single point of failure in the system.
- *(SP2) End-to-end Data Protection*: The confidentiality of the data stored at the SPUs must be protected at rest, in transit and during computation. The data are encrypted by the clinical site and the result of the query can be decrypted only by the investigator issuing the query.

Depending on the access privileges of the investigator querying the system, MedCo should be able to also provide the following optional features (either one or both of them):

- *(SP3) Unlinkability*: The investigator must not be able to trace a query response back to its original clinical site.
- *(SP4) Result Obfuscation*: The query result is obfuscated in order to achieve formal privacy guarantees (e.g., differential privacy) and prevent re-identification.

5 MedCo Core Architecture & Protocols

In this section, we provide a detailed description of MedCo. We begin with a brief overview of the system architecture and core querying protocol. Then, we describe in detail the different steps of the system initialization and the data ingestion phases. Finally, we describe the steps of the secure querying protocol that enables an investigator to efficiently query the distributed encrypted data stored at the different storage and processing units.

5.1 General Overview

The main purpose of MedCo, whose architecture is depicted in Fig. 3, is to reassure clinical sites willing to share their clinical and genomic data with investigators, by enabling clinical sites to securely outsource the storage and processing of their data to a set of potentially untrusted storage and processing units. In order to achieve the privacy and security goals mentioned in Section 4.3, MedCo enables SPUs to collectively generate an encryption key for an additively-homomorphic encryption system,[1] used by clinical sites to encrypt their data before leaving the local trusted zone of the site. Through a set of secure distributed protocols, MedCo enables the SPUs (i) to switch the encryption of the data from probabilistic encryption to deterministic encryption in order to securely process equality-matching queries, and (ii) to re-encrypt the query result from an encryption with the collective public key to an encryption under the investigator's public key, so that (only) the investigator can eventually decrypt the result. And, depending on the access privileges of the investigator issuing the query, MedCo can securely shuffle and/or obfuscate the query results in order to achieve unlinkability and/or differential privacy, respectively (see Section 4.3.2).

5.2 System Initialization

During the initialization of MedCo, each storage and processing unit (SPU_i) generates a pair of EC-ElGamal cryptographic keys (k_i, K_i), where $K_i = Gk_i$, along with a secret s_i. Then, all SPUs combine their EC-ElGamal public keys in order to generate a single collective public key $K = \sum_i K_i$ that will be used by the different clinical sites to encrypt the data to be outsourced.

5.3 Data Extraction Transformation and Loading

During the data-ingestion phase, i.e., *extraction transformation and loading (ETL)* phase, each clinical site extracts patient-level data from its private EHR system or clinical research data warehouse, and transforms the data in order to fit the "star-schema" data model [20] used by MedCo. The star-schema data model is based on the Entity-Attribute-Value (EAV)

1. For performance reasons, in this work we use EC-ElGamal, but any other additively homomorphic scheme can be used as well.

concept also used by widespread clinical research systems such as i2b2 [11], where clinical and genetic observations (or "facts") about patients (e.g., diagnosis, medications, procedures, laboratory values and genetic variants) are stored in a narrow table called "fact" table. Observations are encoded by ontology concepts from an extensible set of medical terminologies, e.g., the International Classification of Disease (ICD) or the US National Drug Code (NDC). In this data model, four other "dimension" tables further describe the patients' data and meta-data. For example, the "patient dimension" table contains pseudonymized demographic information of the patients, and the "visit dimension" table stores information about the visit, such as its date and time and the type of provider.

In such a data model, the information that clinical sites want to protect from potential honest-but-curious adversaries at the storage and processing units is represented by the mapping between the patients in the database and the set of their clinical and genomic observations stored in the "fact" table that are considered to be sensitive or identifying. In order to protect such mapping, each site separately performs the following three steps:

A. *Generation of Dummy Patients:* Each site generates a set of dummy patients with plausible clinical observations specifically chosen so that the distribution of observations across patients in the "fact" table is as close as possible to the uniform distribution. We explain the rationale behind this step in detail in Section 6. To distinguish the real patients from the dummies, each site also generates a binary flag to be appended to the demographic information in the "patient dimension" table. Such flag is set to 1 for real patients and to 0 for dummy patients.
B. *Data Encryption:* In order to break the link between the patients and their sensitive observations in the "fact" table, each site encrypts with the collective public key K the set of ontology concepts that encode these observations along with the patients' binary flags. As EC-ElGamal is a probabilistic encryption scheme, each clinical site obtains a set of probabilistic ciphertexts that are totally indistinguishable from each other.
C. *Data Loading and Re-Encryption:* After encryption, each site uploads the encrypted data to the selected storage and processing unit that immediately starts a *Distributed Deterministic Re-Encryption (DDR)* protocol (the details of this protocol are explained in Section 5.5) in which the encrypted concepts are sent across the network of SPUs so that their encryption is switched from probabilistic to deterministic. This re-encryption is necessary for enabling the secure processing of equality-matching queries (as those defined in Section 4.3) that otherwise would be impossible with probabilistic ciphertexts. Due to the presence of dummy patients, even if the deterministic nature of the ciphertexts leaks the equality of the underlying plaintexts, an honest-but-curious adversary is not able to perform a frequency attack to distinguish ontology concepts based on their frequency distribution. Dummy patients are indistinguishable from real patients, as long as the patients' binary flags are probabilistically encrypted.

5.4 Secure Query Protocol

We assume each investigator that uses MedCo has a pair of EC-ElGamal cryptographic keys (k_I, K_I) and, optionally, is assigned an initial differential privacy budget ϵ_I during the registration phase. The purpose of such a budget is to limit the number of queries an investigator with low privileges can run on the system, hence ϵ_I-differential privacy can be guaranteed. The proposed secure query protocol is illustrated in Fig. 3 and comprises the following steps:

1. *Query Generation:* The secure query protocol starts with an authenticated and authorized investigator who wants to obtain either the number of patients or the pseudonyms of the patients who match a set of inclusion/exclusion clinical and genetic criteria across the different clinical sites. In clinical research, this procedure is called "cohort selection". For this purpose, the investigator builds a query by logically combining (i.e., through AND and OR operators) a set of "sensitive" and "non-sensitive" concepts from a common (i.e., shared across the different sites) ontology. The "sensitive" concepts in the query are encrypted with the collective public key K and the query is sent along with the investigator's public key K_I to one of the storage and processing units.
2. *Query Re-Encryption:* The SPU that receives the query starts a *Distributed Deterministic Re-Encryption* protocol (described in Section 5.5) in order to switch the encryption of the sensitive concepts in the query from probabilistic to deterministic. Once the DDR protocol is over, the initial SPU broadcasts the deterministic version of the query to the other SPUs in the network.
3. *Local Query Processing:* Each SPU locally processes the query by filtering the patients (both dummy and real) in the "patient dimension" table whose observations in the "fact" table (both the unencrypted and the deterministically encrypted ones) match the concepts in the query. If the query requests the list of matching patients' pseudonyms, each SPU returns the list of matching patients' pseudonyms along with the probabilistically encrypted binary flags. If the query requests the number of matching patients, each SPU homomorphically adds the matching-patients' dummy flags and returns the encrypted result $E_K(R_i) = E_K(\sum_{j\in\phi} f_i^j) = \sum_{j\in\phi} E_K(f_i^j)$, where $E_K(f_i^j)$ is the encrypted flag of the jth patient in site S_i and ϕ is the set of patients matching the query. In the homomorphic summation, the binary flags of the dummy patients have a null contribution (i.e., $E_K(0)$), hence the encrypted final result corresponds to the actual number of real matching patients.
4. *Result Obfuscation:* This step is optional and depends on (i) the type of query and (ii) the investigator's privileges. In order to guarantee differential privacy, each SPU can obfuscate the encrypted patient counts computed during the previous step by homomorphically adding noise sampled from a Laplacian distribution. More specifically, let ϵ_q be the privacy budget allocated for a given query q and μ be the noise value drawn from a Laplacian distribution with mean 0

and scale $\frac{\Delta f}{\epsilon_q}$, where the sensitivity Δf is equal to 1, due to R_i being a count. Then, the encrypted obfuscated query result is obtained as $E_K(\tilde{R}_i) = E_K(R_i + \mu) = E_K(R_i) + E_K(\mu)$. We note that the query result is released to the investigator only if the investigator's differential privacy budget is enough for such a query, i.e., if $\epsilon_I - \epsilon_q > 0$.

5. *Result Shuffling:* This step is also optional and depends, as the previous step, on (i) the type of query and (ii) the investigator's privileges. In order to break the link between the encrypted (potentially obfuscated) query results generated at the different SPUs and the corresponding clinical sites, the SPUs jointly run a *Distributed Verifiable Shuffling (DVS)* protocol (described in Section 5.5) on the set of encrypted patient counts. As a result, each SPU receives encrypted counts,[2] that might have been generated by another SPU.
6. *Result Re-Encryption:* The query results securely computed by each SPU are encrypted with the collective key K; to be decrypted by the investigator, each SPU runs a *Distributed Key Switching (DKS)* protocol (described in Section 5.5) that involves the other SPUs and switches the encryption of the query results from an encryption with K to an encryption with K_I, the investigator's public key. After this, the newly encrypted query results are sent back to the the SPU that initiated the protocol and then on to the investigator.
7. *Result Decryption:* As the query results are encrypted with K_I, the investigator can use the corresponding secret key k_I to decrypt them and obtain the corresponding plaintext values. If the query results are the list of patients' pseudonyms along with the patients' binary flag, the investigator can simply rule out the dummy patients by discarding those who have the flag set to zero.

5.5 Secure Sub-Protocols

The secure query protocol of MedCo is based on three secure and distributed sub-protocols re-adapted from [21]. In this section, we describe them in detail.

- *Distributed Deterministic Re-Encryption Protocol.* The DDR protocol enables a set of SPUs to deterministically re-encrypt data that are probabilistically encrypted under the collective key generated by all SPUs, without ever decrypting the data. The purpose of this protocol is to enable equality-matching queries on probabilistically encrypted data that otherwise would not be possible. More formally, let n be the number of SPUs in the network, $E_K(M) = (C_1, C_2) = (rG, M + rK)$ be the encryption of a message M under the collective public key K. The DDR protocol comprises two rounds through all SPUs. In the first round, each SPU_i sequentially uses its secret s_i and adds s_iG to C_2. After this first round, the resulting ciphertext is $(\tilde{C}_{1,0}, \tilde{C}_{2,0}) = (rG, M + rK + \sum_{i=1}^{n} s_iG)$. In the second round, each SPU partially and sequentially modifies this ciphertext. More specifically, when SPU_i receives the modified ciphertext $(\tilde{C}_{1,i-1}, \tilde{C}_{2,i-1})$ from SPU_{i-1}, it computes $(\tilde{C}_{1,i}, \tilde{C}_{2,i})$, where $\tilde{C}_{1,i} = s_i\tilde{C}_{1,i-1}$ and $\tilde{C}_{2,i} = s_i(\tilde{C}_{2,i-1} - \tilde{C}_{1,i-1}k_i)$. At the end of the second round, the deterministic re-encryption is obtained by keeping only the second component of the resulting ciphertext $DT_s(M) = C_{2,n} = sM + \sum_{i=1}^{n} s_i sG$, where $s = \prod_{i=1}^{n} s_i$ is the collective secret corresponding to the product of each SPU's secret.
- *Distributed Verifiable Shuffling Protocol.* The DVS protocol enables a set of SPUs to sequentially shuffle probabilistically encrypted data so that the outputs cannot be linked back to the original ciphertexts. More specifically, the DVS protocol uses the Neff shuffle [22]. It takes as input multiple sequences of EC-ElGamal pairs $(C_{1,i,j}, C_{2,i,j})$ forming a $a \times b$ matrix, and outputs a shuffled matrix of $(\bar{C}_{1,i,j}, \bar{C}_{2,i,j})$ pairs such that for all $1 \leq i \leq a$ and $1 \leq j \leq b$, $(\bar{C}_{1,i,j}, \bar{C}_{2,i,j}) = (C_{1,\pi(i),j} + r''_{\pi(i),j}B, C_{2,\pi(i),j} + r''_{\pi(i),j}P)$, where $r''_{i,j}$ is a re-randomization factor, π is a permutation and P is a public key.
- *Distributed Key Switching Protocol.* The DKS protocol enables a set of SPUs to convert a ciphertext generated with the collective public key K into a ciphertext of the same data generated under any known public key U, without ever decrypting them. The DKS protocol never makes use of decryption. Let $E_K(M) = (C_1, C_2) = (rG, M + rK)$ be the encryption of a message M with the collective public key K. The DKS protocol starts with a modified ciphertext tuple $(\tilde{C}_{1,0}, \tilde{C}_{2,0}) = (0, C_2)$. Then, each SPU partially and sequentially modifies this element by generating a fresh random nonce v_i and computing $(\tilde{C}_{1,i}, \tilde{C}_{2,i})$ where $\tilde{C}_{1,i} = \tilde{C}_{1,i-1} + v_iG$ and $\tilde{C}_{2,i} = \tilde{C}_{2,i-1} - k_iC_1 + v_iU$. The resulting ciphertext corresponds to the message m encrypted under the public key U, $(\tilde{C}_{1,n}, \tilde{C}_{2,n}) = (vG, M + vU)$ from the original ciphertext (C_1, C_2), where $v = v_1 + \cdots + v_n$.

6 Dummy-Addition Strategies

For cohort-exploration queries, the deterministic encryption of the ontology concepts applied during the ETL phase (see Section 5.3) avoids dictionary attacks by any subset of colluding HBC SPUs due to the distribution of the secrets s_i used in the DDR protocol. Nevertheless, a *generation-of-dummy-patients* step is required prior to encryption in order to avoid leaking to the SPUs (i) the ontology concepts distribution and (ii) the query result. In this section, we analyze the optimal dummy-generation strategy to achieve this goal.

We assume, without loss of generality, that each patient has a different set of observations; if there were equal patients in the database, fake ontology concepts could be added to make them different. The leakage to HBC SPUs can be estimated by calculating (i) the adversary's equivocation (i.e.,

2. The number of encrypted counts received by an SPU corresponds to the number of sites that have outsourced the storage of their data to that SPU.

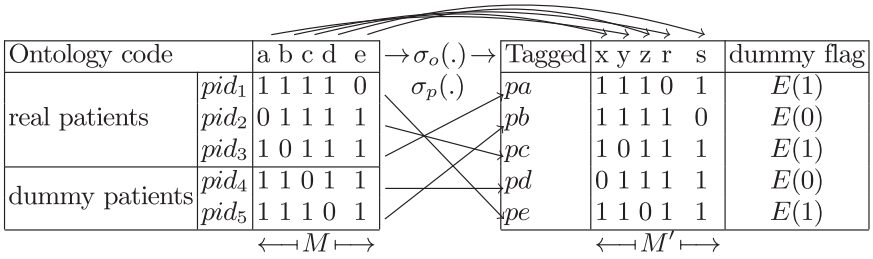


Fig. 4. *Toy example.* Ontology concepts mapping to real and added dummy patients with pseudo-identifiers pid_i, and ontology concepts a, b, c, d, e. pa, pb, pc, pd, pe are the randomly sorted version of the patient pseudo-identifiers, and x, y, z, r, s are the shuffled and deterministically re-encrypted version of the ontology concepts. The binary flag is a probabilistic encryption of 1 for real patients and 0 for dummies.

conditional entropy) on the ontology concepts of the "fact" table given their tagged versions, as an average measure, and (ii) the smallest anonymity set of the ontology concepts, as a worst-case measure. The higher the equivocation and the larger the anonymity set is, the lower the leakage is. For this exposition, we will focus only on the relation between patients and occurrences of sensitive ontology concepts, leaving aside the temporal dimension. This is a simplifying assumption, implying that (a) either there are no causality relations between concepts, or the time dimension is encrypted or not available in the database, and that (b) the non-sensitive non-encrypted concepts are independent of the encrypted ones; if this is not the case, dependent concepts should be reclassified as sensitive and be encrypted. We will follow the toy example shown in Fig. 4. This figure represents the (horizontally) folded version of the (vertical) "fact" table, therefore coding each patient as a row, each ontology concept as a column, and each observed (resp. unobserved) concept in a patient as a "1" (resp. "0") in the corresponding cell.

More formally, let us define the matrix that associates ontology concepts with patients as the tuple of a random binary matrix $\mathcal{M}$, where each row can be either a real or a dummy patient, and each column represents one ontology concept and two functions σ_p and σ_o, which map the patient pseudo-identifiers (pid_j in Fig. 4) to the rows (pa, pb, pc, pd, pe in Fig. 4), and the observed ontology concepts (a, b, c, d, e in Fig. 4) to the columns (x, y, z, r, s in Fig. 4), respectively. These maps represent the shuffling applied to patients before they are assigned their pseudo-identifiers, and the shuffling and deterministic re-encryption applied to ontology concepts before they are loaded into the SPU's database. In order to focus on the practical leakage of the deterministically encrypted database, let us assume that the deterministic re-encryption of the concepts and the probabilistic encryption of the patients' binary flags do not leak anything about their inputs (their trapdoors cannot be broken), even if they are based on computational guarantees. Therefore, the adversary (each of the SPUs) observes the realization of the row- and column-permuted matrix: $\mathcal{A} \equiv [\mathcal{M}' = M']$, and her equivocation, with respect to the original information given $\mathcal{A}$, can be expressed as

$$H(\mathcal{M}, \sigma_o, \sigma_p|\mathcal{A}) = H(\mathcal{M}|\sigma_o, \sigma_p, \mathcal{A}) + H(\sigma_o|\sigma_p, \mathcal{A}) + H(\sigma_p|\mathcal{A}) \quad (1)$$

$$\begin{aligned} &\overset{(a)}{=} H(\sigma_o|\sigma_p, \mathcal{A}) + H(\sigma_p|\mathcal{A}) \\ &\overset{(b)}{\leq} H(\sigma_o|\mathcal{A}) + H(\sigma_p) \\ &\overset{(c)}{\leq} H(\sigma_o) + H(\sigma_p). \end{aligned} \quad (2)$$

Expression (1) can be divided in three terms: the first represents the entropy of $\mathcal{M}$ conditioned to the two permutations and the observed contents of the cells, which is fully deterministic, hence zero-entropy (step (a) in (2)); the second term is the entropy of the ontology concepts permutation conditioned to the observation of the matrix cells and the patient permutation, and the third term is the entropy of the patient permutation conditioned on the observed matrix contents. We aim at maximizing these two terms.

The last term of the equivocation can be maximized by making the dummy patients indistinguishable from the real patients, i.e., drawn from the same distribution. Empirically, this means that all the patients, real or dummy, have the same type of distribution, and the contents of the rows are independent of the position of the dummy patients in the list. This also makes the two permutations independent of each other even when conditioned on the contents of M' (step (b) in (2)). In our toy example in Fig. 4, all the real patients' rows belong to the same type (weight 4); by generating two new dummy patients with the same weight, they become indistinguishable from real patients in our simplified example.

In order to maximize the entropy of the ontology concepts mapping σ_o conditioned on $\mathcal{A}$ (step (c) in (2)), all the permutations have to be equiprobable for the given M'. This is achieved by flattening the joint distribution of the observed ontology concepts through the added dummies; the geometric interpretation of this flattening is that any column permutation can be cancelled out by a row permutation, such that it is not possible to univocally map any ontology concept to any column in M'. In our toy example, it can be seen that due to the two added dummies, any fixed query yields the same number of patients independently of the permutation applied to the query terms, which gives a complete indistinguishability between all the deterministically encrypted ontology concepts even in light of the matrix M'. It must be noted that the unobserved concepts do not have to be added to the table, as the adversary does not have a priori knowledge of which is the subset of observed concepts, only its cardinality. Also, this strategy fully breaks the correlation between ontology concepts; for example, if the site added only one dummy patient with concepts a, b, e to the real patients in Fig. 4 the individual appearance rate of the concepts would be flattened, but it would leak that there is a correlation between the concepts c and d, that could be identified in the encrypted matrix through an l_p-optimization attack [23].

The last bound in (2) is the best that clinical sites can do with the dummy-patient addition strategy, knowing the matrix of real patients; it maximizes the uncertainty of the attacker about the original ontology concepts, for any real

distribution of patients and ontology concepts. The corresponding practical dummy-addition strategy can be described as follows: Real rows are grouped according to their weight (number of observations); if the whole set of observed ontology concepts has n elements, for each group of rows of weight $k < n$, dummy rows are added to complete all the k-combinations of n elements, producing $\binom{n}{k}$ rows (counting both real and dummies) per group. In our toy example, (considering independent concepts) the equivocation goes from 3.58 bits with no dummies to 10.23 bits with the two dummies, and the minimum anonymity set raises from 2 to 5.

This strategy guarantees the maximum uncertainty for the adversary for an arbitrary real distribution of concepts across patients, but it generates a combinatorial number of dummies, which is not feasible in general (unless the number of observed concepts is very low). But if some assumptions can be made about the concepts joint distribution, we can simplify the strategy. If dependencies are only found within small groups of concepts, the groups being mutually independent (this is the case for genomic information and dependencies found inside subsets of localized variants), it is possible to constrain the needed number of dummies by applying the same dummy-addition strategy in a restricted block-wise fashion. In order to flatten only the histogram of group weights, we group the concepts in independent blocks of size $n' \ll n$ and apply the dummy-generation permutation to the blocks (inter-block), but not to the contents of each block, until the block distribution is flat, therefore reducing the needed number of dummy rows. This trade-off strategy creates an "anonymity set" of ontology concepts of size n/n' in such a way that the adversary cannot distinguish between the set of concepts inside different blocks. The drawback is that the equivocation is reduced, as the resulting joint distribution of the ontology concepts is only flat across blocks, but not inside each block. In the worst case in terms of leakage (fully correlated concepts within each block), the achievable adversary's equivocation becomes $H(\mathcal{M}, \sigma_o, \sigma_p|\mathcal{A}) = H(\sigma_o|\sigma_p, \mathcal{A}) + H(\sigma_p|\mathcal{A}) \leq H(\sigma_{o,n/n'}) + H(\sigma_p)$, where $\sigma_{o,n/n'}$ are the permutations of the n/n' blocks of n' concepts each. This bound is achieved when the blocks are mutually independent, hence the best partitioning strategy consists in keeping correlated concepts inside the same block. If fully independence between concepts can be assumed ($n' = 1$), it can be seen that flattening the observations histogram leads to the same maximum attacker equivocation as the complete permutation strategy (Eq. (2)), but with a much lower number of added dummies. In order to further reduce this number, it is possible to set a minimum anonymity set size m for the concepts and add dummies to water fill the observation histogram (block-wise flat, instead of fully flat) until each concept has at least other $m-1$ concepts featuring the same number of observations.

Finally, it must be noted that whenever a site's database is updated, dummies can be regenerated (and encryptions re-randomized) when the ETL process (see Section 5.3) is run again for the whole updated database. The DDR protocol uses a different fresh randomness, so that the concepts from the updated database cannot be linked back to the concepts of the old one.

7 Privacy & Security Analysis and Extensions (MedCo+)

The main privacy and security goals for MedCo are summarized in Section 4.3. In this section, we briefly discuss and analyze the fulfillment of these targets for MedCo, and we revisit possible extensions for more stringent requirements.

Security in MedCo is based on the cryptographic guarantees provided by the underlying decentralized sub-protocols described in Section 5.5. All input sensitive data are either deterministically (ontology concepts) or probabilistically (patients' binary flags) encrypted with collectively maintained keys, such that they cannot be decrypted without the cooperation of all sites, thus guaranteeing confidentiality and avoiding single points of failure (SP1 in Section 4.3). For the full step-by-step security analysis of the distributed sub-protocols, we refer the reader to [21]. Following this analysis, paired with the dummy strategy described in Section 6, it can be seen that MedCo covers the unlinkability requirement (SP3 in Section 4.3) for the query results, thanks to the DVS protocol; and it protects their confidentiality, as only the authorized investigator can decrypt the query results thanks to the DKS protocol (SP2 in Section 4.3). Conversely, to avoid re-identification (or attribute disclosure) attacks (SP4 in Section 4.3), MedCo also enables the application of differentially private noise to the results and, due to the proposed dummy strategy, it guarantees confidentiality of the data also against all the SPUs that participate in the system (SP2 in Section 4.3).

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

- *Query confidentiality:* In the basic MedCo system presented in Section 5, HBC SPUs can link the ontology concepts used across different queries, as the deterministically encrypted values of the same concepts are the same for all the queries. In the case that query confidentiality is also a requirement (e.g., investigators from pharmaceutical companies), it is possible to address it by probabilistically encrypting ontology concepts during the ETL phase and by deterministically re-encrypting the obtained ciphertexts with a fresh secret for each new query. Then, the effective encryption key is different for each fresh run of the DDR protocol, so it is not possible to link the query terms between different runs of the shuffling-DDR. When this modified system (which we denote MedCo+) is paired with the proposed dummy-addition strategy, the terms between queries are indistinguishable and unlinkable, at the cost of transferring and re-encrypting at runtime the encrypted database of each site.
- *Malicious SPUs:* MedCo's threat model assumes HBC SPUs to be a credible and plausible assumption, based on the damage to reputation that a SPU would suffer if it misbehaves in a collective data-sharing protocol. Nevertheless, it is possible to cope with malicious SPUs by using proof generation protocols [21] that produce and publish zero-knowledge proofs for all

the computations performed at the SPUs, hence the proofs can be verified by any entity in order to assess that no SPU deviated from the correct behavior. This solution yields a hardened and resilient query protocol, but the cost of producing all proofs results in a typically unacceptable burden in common data sharing applications, for which the basic proposed MedCo covers all fundamental privacy and security requirements and yields a very competitive performance, as shown in the next Section.

8 Implementation and Evaluation

We implemented and tested MedCo on a clinical oncology use-case by simulating a network of three clinical sites, each one outsourcing the storage of their data to a different SPU.

8.1 Implementation

To ease its adoption at clinical sites, we implemented MedCo as three components that fully integrate within the i2b2 [11] framework and its networking system SHRINE [12]. i2b2 (Informatics for Integrating Biology and the Bedside) (i2b2) is the state-of-the-art clinical platform for enabling secondary use of electronic health records (EHR) [11]. It is currently used at more than 300 medical institutions, covering the data of more than 250 million patients. Its back-end consists of a set of server-side software modules implemented in Java, called "cells", that are responsible for the business logic of the platform and are organized in a "hive". The i2b2 data model is based on the "star schema" [20]. Queries are built in a dedicated JavaScript-based Web-client by logically combining ontology concepts organized in a hierarchical tree-based structure. The three components of MedCo are:

- A new i2b2 server cell, called "MedCo cell", developed in Java and Go. The MedCo cell is responsible for the execution of the secure query protocol and communicates with the other i2b2 cells through a REST API. We used the UnLynx library [21] to implement the DDR, DVS and DKS secure distributed subprotocols.
- A new i2b2 Web-client plugin developed in JavaScript. The plugin is responsible for managing the cryptographic operations in the browser.
- A data importation tool, developed in Go, that is responsible for encrypting the sensitive ontology concepts and generating the dummy patients.

These components are publicly available at [24]. We note that MedCo is not limited to i2b2/SHRINE but can also be integrated on top of other state-of-the-art platforms for clinical and translational research, such as TranSMART [2], in order to make them secure and distributed.

8.2 Oncology Use-Case

The lack of privacy and security guarantees of existing tools makes sharing sensitive oncological data outside the trusted boundaries of clinical sites extremely difficult, if not impossible. For this reason, we tested MedCo on genomic and clinical data from The Cancer Genome Atlas (TCGA) [25] by performing typical queries for oncogenomics. We report here two representative examples:

- *Query A: Number of patients with skin cutaneous melanoma AND a mutation in BRAF gene affecting the protein at position 600.* About half of melanoma patients harbor a mutation in the BRAF gene at position V600E or V600K and can be treated by the BRAF inhibitor *vemurafenib* [26]. The proportion of mutated BRAF melanoma is therefore an important benchmark for a clinic or hospital.
- *Query B: Number of patients skin cutaneous melanoma AND a mutation in BRAF gene AND a mutation in (PTEN OR CDKN2A OR MAP2K1 OR MAP2K2 genes).* This query is based on the fact that patients treated with *vemurafenib* develop resistance through mutations that activate the *MAP kinase* pathways [27]. When facing drug resistance, finding another patient with a similar mutation profile could bring invaluable information for clinical decisions.

We used genomic and clinical data of 8,000 cancer patients, 9 clinical attributes, and an average of 142 genetic mutations per patient (more than 1 million observations in total). We imported these data from the Mutation Annotation Format (MAF) into the i2b2 "star schema" data model. Each mutation is represented as a code comprising the concatenation of its chromosome, position, reference allele and tumor allele. Clinical attributes are encoded with the ICD-10 [28] and ICD-O [29] international terminologies.

8.3 Experimental Setup

The initial testing environment comprises 3 servers interconnected by 10 Gbps links and featuring two Intel Xeon E5-2680 v3 CPUs @2.5 GHz that support 24 threads on 12 cores, and 256 GB RAM. Each server represents an SPU and hosts the i2b2/SHRINE Web client with the MedCo plugin, the i2b2 hive including the SHRINE components, the new MedCo cell, and the i2b2 database implemented in PostgreSQL. In order to test MedCo's scalability, we increase the number of servers up to 9 (see setup S3 below). To set up our system and facilitate its deployment, we use Docker [30].

To evaluate MedCo's performance, we consider five different experimental setups, with each measurement averaged over 10 independent runs, and show MedCo's computational and storage overhead with respect to an unprotected i2b2/SHRINE deployment:

S1. *ETL runtime for increasing dataset size:* We analyze the amount of time needed to extract, transform and load the data (pre-processing), which includes the formatting, the initial probabilistic encryption, the deterministic re-encryption of sensitive ontology concepts, and the loading of the data in the i2b2 database.

S2. *Query runtime breakdown:* We run queries A and B (see Section 8.2) on a federation of 3 SPUs, each storing the full initial dataset (i.e., around 1 million observations on 8,000 patients at each SPU), and report the query-runtime breakdowns for each step of the secure query protocol.

S3. *Query runtime for increasing dataset size:* We run queries A and B (see Section 8.2) on a federation of 3 SPUs in order to study MedCo's scalability with respect to increasing dataset sizes.

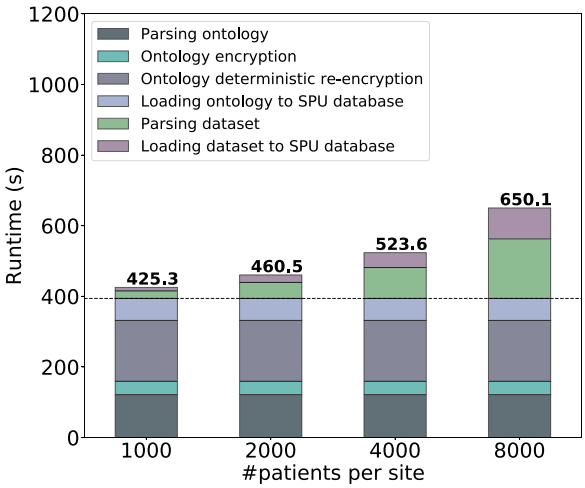


Fig. 5. ETL time versus database size for experimental setup S1.

S4. *Query runtime overhead for increasing number of SPUs:* We run queries A and B (see Section 8.2) on a federation with an increasing number of SPUs, each storing the whole initial dataset.

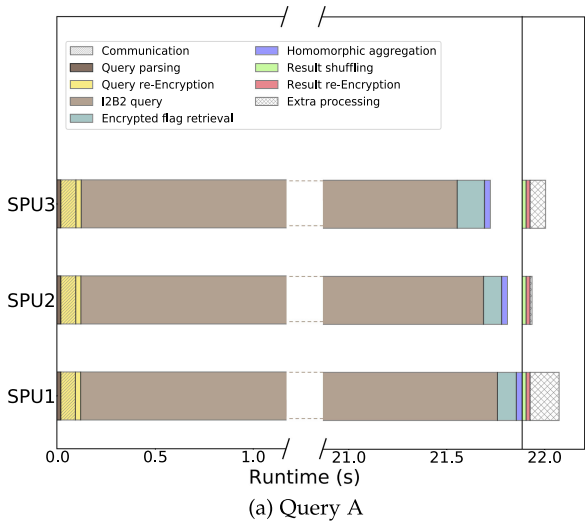
S5. *Network traffic for varying query size:* We study the amount of network traffic inter-SPU for queries with an increasing number of ontology concepts.

8.4 Performance Results

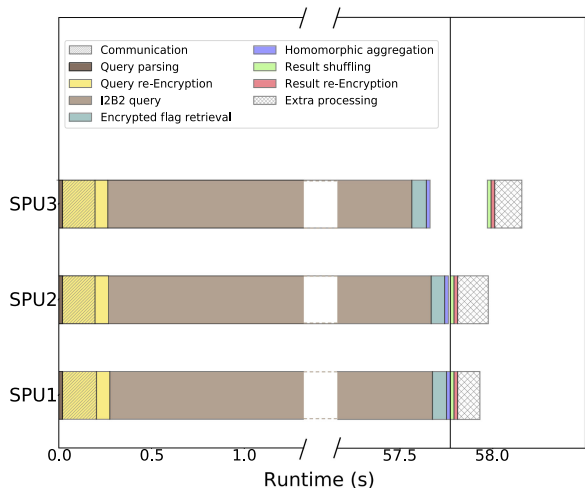
In the following, we report the performance results for the aforementioned use-cases and experimental setups. We show MedCo's computational and storage overhead with respect to an unprotected i2b2/SHRINE deployment.

As shown in Fig. 5, the ETL phase (setup S1) is a costly operation in MedCo. We can distinguish two separate subphases: (i) the processing of the ontology (including the parsing, the encryption and the distributed deterministic re-encryption), which only depends linearly on the size of the ontology and is usually constant, and (ii) the processing of patients' observations, which depends linearly on the number of observations/patients but does not involve any costly encryption operation hence it is much faster than the ontology processing. We note that the ETL phase is performed only once and can be significantly optimized through parallel computing. If new data need to be added after the first importation, there is no need to re-process the ontology again.

Fig. 6 provides query-runtime breakdowns for both query A and query B (setup S2). The times for query-parsing and encryption/decryption in the Web client, broadcasting the query across the different SPUs, and result obfuscation are all negligible, so we do not account for them. Unexpectedly, results show that the standard i2b2 query to the central "fact" table is the most expensive operation in MedCo, as it depends on the total number of observations in the database. In this case, each SPU stores approximately 1 million observations (both genomic and clinical) per affiliated clinical site (one site per SPU in our setting). This time is also linear in the number of ontology concepts used in the query (96 for query A and 281 for query B) and it is inherent to the standard i2b2 database management for SQL-queries to the "fact" table. The times for fetching the encrypted patients' binary flags from the "patient dimension" table and the homomorphic



(a) Query A

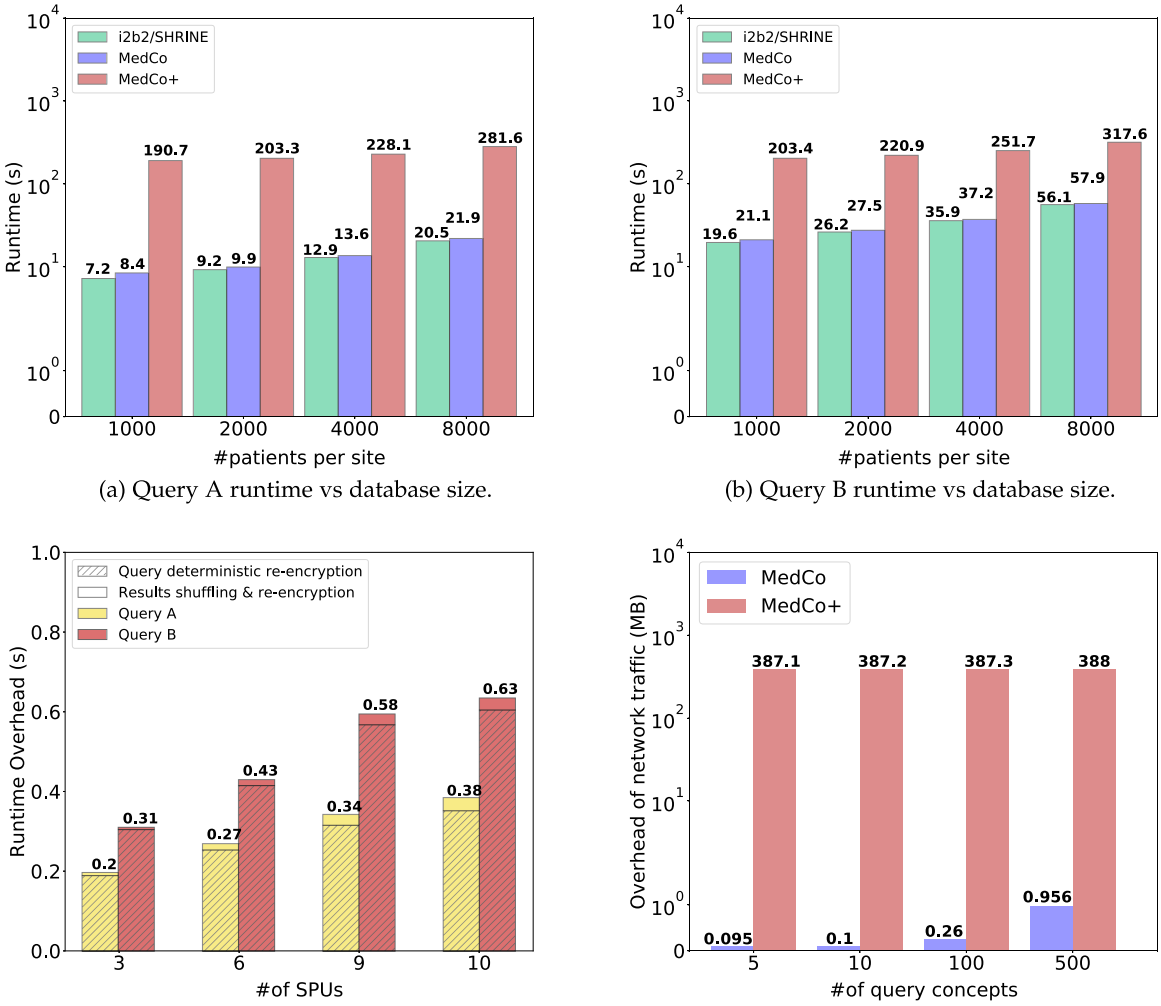


(b) Query B

Fig. 6. Query-runtime breakdown for queries A and B in a network with three sites and three SPUs for experimental setup S2. The vertical black line signals the point where each node has to wait for the others before it can proceed.

aggregation (Step 3 in the query workflow) depend linearly on the number of patients satisfying the query criteria and can be extremely fast for rare ontology concepts or rare combinations of concepts. For example, for queries A and B, homomorphic aggregation takes around 30 and 8 milliseconds respectively, as only around 32 and 7 patients per site satisfy the query criteria. Differently, the deterministic re-encryption time is linear in the number of sensitive concepts in the query and number of SPUs in the network, as each probabilistically encrypted concept has to be sequentially modified by each SPU. Such a process takes less time for query A than for query B, as they respectively comprise 96 (95 mutations and 1 clinical attribute) and 281 (280 mutations and 1 clinical attribute) query attributes. The remaining secure distributed operations introduced by MedCo depend on the number of SPUs in the network, but they are negligible, as they involve only one ciphertext, i.e., the encrypted query result.

Fig. 7 shows the performance results for setups S3-S5. The measurements are averaged out between SPUs. For setup S3 (Figs. 7a and 7b), in order to study MedCo's ability to scale with increasing database sizes, we randomly sample



(a) Query A runtime vs database size.

(b) Query B runtime vs database size.

(c) Queries A and B runtime vs number of SPUs.

(d) Network traffic vs query size

Fig. 7. MedCo's performance results for experimental setups S3-S5.

patients from the original dataset of 8k patients and create smaller datasets of 1k, 2k and 4k patients per site. For setups S4 and S5 (Figs. 7c and 7d), we use the initial dataset (8k patients). Results show that MedCo is extremely efficient and performance-wise comparable to the insecure i2b2/SHRINE deployment. MedCo's overhead only depends on the number of sensitive concepts in the query, the number of matching patients satisfying the research criteria and, marginally, on the number of SPUs in the network. As shown in Fig. 7c, the number of SPUs affects only the time needed by the distributed protocols to deterministically re-encrypt the sensitive ontology concepts in the query and to re-encrypt the query end-result under the investigator's key.

In Figs. 7a, 7b and 7c, we can also observe that MedCo+ has a relatively higher runtime cost as a counterpart for achieving query unlinkability, because all the observations in the "fact" table of each SPUs have to be deterministically re-encrypted on the fly by the whole set of SPUs for each new query. This is confirmed by Fig. 7d where the network traffic is significant and almost constant for MedCo+, whereas for MedCo it is almost negligible and it increases with the number of concepts in the query. We note, however, that the privacy enhancements brought by MedCo+ might be necessary only under specific circumstances (e.g., when an investigator from a pharmaceutical company is using the system).

Finally, the storage overhead introduced by encryption affects only the "concept dimension" table that stores the ontology, and it is in the order of 4x, as MedCo's deterministic re-encryption converts each ontology concept, represented by 64-bit integers, into a 32-bytes ciphertext. Depending on the specific distribution of ontology codes across patients, a varying number of dummy patients must also be considered. In the tested oncology use-case, we assume independent codes and follow the dummy-addition strategy described in Section 6. As a result, we obtain an increase factor of 3.6x.

9 Conclusion

In this paper, we have presented MedCo, the first operational scalable system that enables secure sharing of sensitive medical data, which so far was impossible due to the low security guarantees of existing operational systems. MedCo relies on secure distributed protocols and a new dummy-records addition strategy that enables different privacy/security versus efficiency trade-offs. With its generic architecture, MedCo is easily deployable on top of existing health information systems such as i2b2 or tranSMART. Finally, results on a clinical oncology use-case have shown

practical query-response times and good scalability with respect to the number of sites and amount of data. Therefore, we firmly believe that MedCo represents a concrete solution for fostering medical data sharing in a privacy-conscious and regulatory-compliant way.

REFERENCES

[1] J. V. Selby, A. C. Beal, and L. Frank, "The patient-centered outcomes research institute (PCORI) national priorities for research and initial research agenda," *J. Amer. Med. Assoc.*, vol. 307, no. 15, pp. 1583–1584, 2012.

[2] B. D. Athey, M. Braxenthaler, M. Haas, and Y. Guo, "tranSMART: An open source and community-driven informatics and data sharing platform for clinical and translational research," *AMIA Summits Translational Sci. Proc.*, vol. 2013, 2013, Art. no. 6.

[3] Swiss Academies of Arts and Sciences, "Swiss personalized health network." [Online]. Available: http://www.samw.ch/en/Projects/SPHN.html, Last Accessed on: Jul. 11, 2018.

[4] The Global Alliance forGenomics and Health, "A federated ecosystem for sharing genomic, clinical data," *Sci.*, vol. 352, no. 6291, pp. 1278–1280, 2016.

[5] U.S. Department of Health & Human Services, "The health insurance portability and accountability act (HIPAA)." [Online]. Available: https://www.hhs.gov/hipaa/index.html, Last Accessed on: Jul. 11, 2018.

[6] EU Parlament, "The EU general data protection regulation (GDPR)." [Online]. Available: http://www.eugdpr.org/, Last Accessed on: Jul. 11, 2018.

[7] All of us research program. [Online]. Available: https://allofus.nih.gov/, Last Accessed on: Jul. 11, 2018.

[8] The 100,000 genomes project protocol v3, genomics England. [Online]. Available: https://www.genomicsengland.co.uk/, Last Accessed on: Jul. 11, 2018.

[9] T. G. A. for Genomics and Health, "Beacon network," 2017. [Online]. Available: https://beacon-network.org/, Last Accessed on: Jul. 11, 2018.

[10] U.S. Department of Health and Human Services, "Breach portal: Notice to the secretary of HHS breach of unsecured protected health information." [Online]. Available: https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf, Last Accessed on: Jul. 11, 2018.

[11] 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)," *J. Amer. Med. Inform. Assoc.*, vol. 17, no. 2, pp. 124–130, 2010.

[12] G. M. Weber, S. N. Murphy, A. J. McMurry, D. MacFadden, D. J. Nigrin, S. Churchill, and I. S. Kohane, "The shared health research information network (SHRINE): A prototype federated query tool for clinical data repositories," *J. Amer. Med. Inform. Assoc.*, vol. 16, no. 5, pp. 624–630, 2009.

[13] G. A. for Genomics and Health, "The beacon project." [Online]. Available: https://beacon-network.org/#/, Last Accessed on: Jul. 11, 2018.

[14] J. L. Raisaro, F. Tramèr, Z. Ji, D. Bu, Y. Zhao, K. Carey, D. Lloyd, H. Sofia, D. Baker, P. Flicek, S. Shringarpure, C. Bustamante, S. Wang, X. Jiang, L. Ohno-Machado, H. Tang, X. Wang, and J.-P. Hubaux, "Addressing Beacon re-identification attacks: Quantification and mitigation of privacy risks," *J. Amer. Med. Inform. Assoc.*, vol. 24, pp. 799–805, 2017.

[15] F. Chen, S. Wang, X. Jiang, S. Ding, Y. Lu, J. Kim, S. C. Sahinalp, C. Shimizu, J. C. Burns, V. J. Wright, et al., "PRINCESS: Privacy-protecting rare disease international network collaboration via encryption through software guard extensions," *Bioinf.*, vol. 33, 2017, Art. no. btw758.

[16] J. Bater, G. Elliott, C. Eggen, S. Goel, A. Kho, and J. Rogers, "SMCQL: Secure querying for federated databases," *Proc. VLDB Endowment*, vol. 10, no. 6, pp. 673–684, Feb. 2017. [Online]. Available: https://doi.org/10.14778/3055330.3055334

[17] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Proc. Annu. Int. Cryptology Conf.*, 2007, pp. 535–552.

[18] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query processing," in *Proc. 23rd ACM Symp. Operating Syst. Principles*, 2011, pp. 85–100.

[19] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2009.

[20] P. M. Nadkarni and C. Brandt, "Data extraction and ad hoc query of an entity—attribute—value database," *J. Amer. Med. Inform. Assoc.*, vol. 5, no. 6, pp. 511–527, 1998.

[21] D. Froelicher, P. Egger, J. S. Sousa, J. L. Raisaro, Z. Huang, C. Mouchet, B. Ford, and J.-P. Hubaux, "UnLynx: A decentralized system for privacy-conscious data sharing," in *Proc. Privacy Enhancing Technol.*, 2017, pp. 152–170.

[22] C. A. Neff, "Verifiable mixing (shuffling) of ElGamal pairs, http://www.votehere.org/vhti/documentation/egshuf-2.0.3638.pdf, Apr. 2004.

[23] M. Naveed, S. Kamara, and C. V. Wright, "Inference attacks on property-preserving encrypted databases," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 644–655. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813651

[24] LCA1, EPFL, "Medco source code." [Online]. Available: https://c4science.ch/w/medco/, Last Accessed on: Jul. 11, 2018.

[25] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, "The cancer genome atlas (TCGA): An immeasurable source of knowledge," *Contemporary Oncology*, vol. 19, no. 1A, 2015, Art. no. A68.

[26] P. A. Ascierto, J. M. Kirkwood, J.-J. Grob, E. Simeone, A. M. Grimaldi, M. Maio, G. Palmieri, A. Testori, F. M. Marincola, and N. Mozzillo, "The role of BRAF V600 mutation in melanoma," *J. Transl. Med.*, vol. 10, no. 1, 2012, Art. no. 85.

[27] H. Yang, D. Kircher, K. Kim, A. Grossmann, M. VanBrocklin, S. Holmen, and J. Robinson, "Activated MEK cooperates with Cdkn2a and Pten loss to promote the development and maintenance of melanoma," *Oncogene*, vol. 36, no. 27, pp. 3842–3851, 2017.

[28] W. H. Organization, *The ICD-10 Classification of Mental and Behavioural Disorders: Clinical Descriptions and Diagnostic Guidelines*, vol. 1. Geneva, Switzerland: World Health Organization, 1992.

[29] A. G. Fritz, *International Classification of Diseases for Oncology: ICD-O*. Geneva, Switzerland: World Health Organization, 2000.

[30] D. Merkel, "Docker: Lightweight Linux containers for consistent development and deployment," *Linux J.*, vol. 2014, no. 239, 2014, Art. no. 2.

Jean Louis Raisaro received the BS degree in bioinformatics and the MS degree in biomedical informatics from the University of Pavia, Pavia, Italy, in 2009 and 2012, respectively, and the PhD degree in computer and communication sciences from EPFL, Lausanne, Switzerland, in 2018. His main interests include 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.

Juan Ramón Troncoso-Pastoriza received the PhD degree in telecom. engineering, in 2012. He is an elected member of the IEEE Information Forensics and Security TC and the IEEE Signal Processing Society Student Services Committee for the period 2017-2019, and an associate editor of four journals on information security (the *EURASIP Journal on Information Security*, *IET Information Security*, *Elsevier Digital Signal Processing*, and the *Elsevier Journal of Visual Communication and Image*). His research interests include secure signal processing, applied cryptography, and genomic privacy, areas in which he has published numerous papers in top conferences and journals and holds several international granted patents.

Mickaël Misbach received the master's degree in communication systems from EPFL in Lausanne, Switzerland, with a specialization in information security. He is expected to graduate in September 2018. During the last years of his studies, he has worked on medical data privacy, being the main developer of the privacy-conscious cohort explorer MedCo.

João Sá Sousa received the BS and MS degrees in informatics engineering from the University of Coimbra and did a 3-month internship at CMU-SV. He is currently a security/privacy software engineer with EPFL under the direction of Professor Jean-Pierre Hubaux. His main interests include wireless security, genomic privacy, cryptography, android development, web development, and business management.

Sylvain Pradervand received the PhD degree in molecular biology from the University of Lausanne in 1998. After a postdoc studying transcriptomics in heart disease models with the University of California San Diego, he turned his interests to bioinformatics. He is currently leading the bioinformatics team of the Genomic Technologies Facility, University of Lausanne, and the bioinformatics team of the clinical research support platform of the Lausanne University Hospital.

Edoardo Missiaglia received the bachelor's degree in biology from the University of Padova in 1994, the master's degree in genetics from the University of Bologna, in 1998, and the PhD degree in pathological oncology from the University of Verona, in 2003. He worked with ICRF (Cancer Research UK) (2001-2003) as a research assistant and with the University of Verona (2003-05) and ICR (2005-2010) as a post-doc and bioinformatician. He has been working as a project manager with SIB (2010-2014). He became the scientific director of the Molecular Pathology Laboratory, Institute of Pathology, CHUV, in August 2014.

Olivier Michielin received the diploma degree in physics from EPFL, in 1991, the MD degree from the University of Lausanne, in 1997, and the PhD degree under the supervision of Jean-Charles Cerottini (LICR) and Martin Karplus (Harvard and Strasbourg Universities). He is an associate professor with the University of Lausanne. He was appointed group leader of the Swiss Institute of Bioinformatics in 2002 and became an assistant professor and private docent with the Medical Faculty of Lausanne, in 2004 and 2005, respectively. In parallel, he has trained as a medical oncologist and obtained his board certification in 2007 with the Multidisciplinary Oncology Center (CePO) of Lausanne where he is currently in charge of the melanoma clinic.

Bryan Ford received the BS degree from the University of Utah and the PhD degree from MIT. He leads the Decentralized/Distributed Systems (DEDIS) Research Group, Swiss Federal Institute of Technology in Lausanne (EPFL). He focuses broadly on building secure decentralized systems, touching on topics including private and anonymous communication, scalable decentralized systems, blockchain technology, Internet architecture, and operating systems. He joined the faculty of Yale University where his work received the Jay Lepreau Best Paper Award and grants from NSF, DARPA, and ONR, including the NSF CAREER award.

Jean-Pierre Hubaux is a full professor with 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 the IEEE and ACM.