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

Jean Louis Raisaro[1], Juan Ramón Troncoso-Pastoriza[1], Mickaël Misbach[1,2], João Sá Sousa [1], Sylvain Pradervand[2,3], Edoardo Missiaglia[2], Olivier Michielin[2], Bryan Ford[1], and Jean-Pierre Hubaux[1]

[1]School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland
[2]Centre Hospitalier Universitaire Vaudois, CHUV, Lausanne, Switzerland
[3]Genomic Technologies Facility, University of Lausanne, UNIL, Lausanne, Switzerland

Abstract. Being able to share large amounts of sensitive clinical and genomic data across several institutions is crucial for precision medicine to scale up. Unfortunately, existing solutions only partially address this challenge and are still unable to provide the strong privacy and security guarantees required by regulations (e.g., HIPAA, GDPR). As a result, currently only very limited datasets of *non-sensitive* and moderately useful information can be shared. In this paper, we introduce MedCo, the first operational system that enables an investigator to explore *sensitive* medical information distributed at several sites and protected with collective homomorphic encryption. MedCo builds on top of established and widespread technology from the biomedical informatics community, such as i2b2 and SHRINE, and relies on state-of-the-art secure protocols for processing encrypted distributed data and complying with regulations. As such, MedCo can be easily adopted by clinical sites thus paving the way to new unexplored data-sharing use cases. We tested MedCo in a real network of three institutions (EPFL, UNIL and CHUV) by focusing on an oncology use-case with real somatic mutations and clinical tumor data. The relatively low overhead introduced by MedCo shows that it represents a concrete and scalable solution for sharing privacy-conscious medical data.

Keywords: homomorphic encryption, data sharing, privacy, i2b2, shrine, UnLynx, 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 to its full potential. Several initiatives, such as the Patient-Centered Clinical Research Network (PCORNet) [17] in USA, eTRIKS/TranSMART [2] in EU, the Swiss Personalized Health Network (SPHN) [18] in Switzerland, and the Global Alliance for Genomics and Health (GA4GH) [19], are laying down the foundations for new biomedical research infrastructures aimed at interconnecting (so far) siloed repositories of clinical and genomic data.

In this global ecosystem, the ability to provide strong privacy and security guarantees in order to comply with strict regulations (e.g., HIPAA [20] in USA or the new GDPR [8] in EU) is crucial, yet extremely challenging to achieve, for biomedical research to be able to scale up. The number of health-data breaches constantly increases [21] and there is significant public pressure to ensure that the privacy and security of the data can be properly protected. Yet, because of the current cultural gap between the medical and the

privacy/security communities, currently deployed technical solutions enabling the sharing of medical and genomic data still provide very limited guarantees in this sense. This constrains researchers to access very limited medical information, often of relatively low interest for research. For example, the Beacon Network of the GA4GH [7] can provide only presence/absence information of a given variant in a distributed database, and the SHRINE system [22] enables a researcher to access only aggregate information (e.g., the number of patients satisfying specific research criteria) from HIPAA-compliant "limited data sets" that exclude any sort of genetic or identifying clinical information. As a result, the development of new technologies that (i) are compliant with regulations, (ii) allow sharing data also beyond the "limited data set", and (iii) can be easily integrated on top of existing systems, is now more urgent than ever for medical research.

We address this challenge by introducing MedCo, the first operational system that enables the privacy-preserving exploration of distributed sensitive (and identifying) medical data by using strong collective encryption. Its purpose is to foster data sharing by distributing trust among different medical institutions that want to expose their data to external queries, in a way that is compliant with regulations. To achieve this, MedCo takes the best of both worlds (medical informatics and IT privacy/security) by building on top of existing and well-established open-source technologies (i) for clinical data exploration, i2b2 [13] and SHRINE [22], and (ii) for distributed and secure data processing, UnLynx [9]. In particular, MedCo enables medical institutions to federate and collectively encrypt their sensitive clinical and genetic data with homomorphic encryption in order to protect them against undesired and illegitimate access (e.g., hackers or insiders), and to enable their exploration through a set of secure distributed protocols.

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

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

- We introduce MedCo, the first operational system enabling the sharing of sensitive clinical and genomic information based on state-of-the-art open-source technologies.
- We deployed and tested MedCo in a federation of three sites (EPFL, UNIL and CHUV), focusing on a clinical-oncology case with public somatic DNA and lung cancer data.
- We propose a new generic method to add dummies in order to mitigate frequency attacks that can incur when probabilistically encrypted data are transformed to deterministically encrypted data for the sake of enabling Boolean queries.

2 MedCo Ecosystem

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

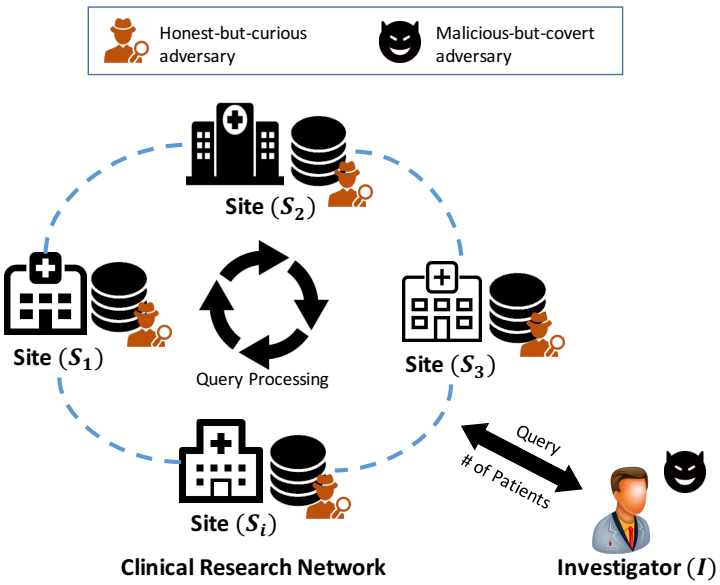


Fig. 1: **MedCo Ecosystem.** System model, including several clinical sites and an investigator; threat model, including honest-but-curious adversaries at the clinical sites and a malicious-but-covert adversary at the investigator.

2.1 System Model

We consider the system model depicted in Figure 1, where medical institutions (or sites) are organized in a decentralized federation (or network) and collaborate to share clinical and genomic data without relying on any central third party or authority. This is the typical model of existing clinical networks such as the GA4GH Beacon Network [7] and Matchmaker Exchange [15], or most of the PCORNet Clinical Data Research Networks [17]. As opposed to the centralized model, our model ensures several advantages such as, for example, the absence of a single point of failure, increased transparency and local control. Each site's data are maintained separately and data access can be monitored by each different institution. In particular, the proposed system consists of the following entities:

- Clinical sites (S_i) such as research institutions, universities or hospitals that own clinical and genomic data and are willing to share them with internal (i.e., affiliated with one of the clinical sites in the federation) and external investigators. The sites are responsible for securely storing the data and for collectively processing incoming queries in a privacy-preserving way. We assume that each clinical site is internally organized into two main departments: (i) a clinical care department where clinical and genomic data are generated by patient encounters and stored in a private electronic health record (EHR) system and (ii) a clinical research department where a subset of data are imported from the EHR system into a research data warehouse that can be exposed to external researchers or investigators for the purpose of data sharing.
- A medical investigator (I) who is interested in exploring the distributed data stored at the different sites. Her main goal is to use MedCo for generating and validating research hypotheses or identifying cohorts of interest to then ask each site for the authorization for obtaining the patients' raw individual data for further analyses.

2.2 Threat Model

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

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

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

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

2.3 Requirements

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

- Functionality. MedCo must provide at least the same functionalities as state-of-the-art systems for cohort exploration on distributed data (e.g., the SHRINE [22]) in order to enable the same use cases (e.g., feasibility studies or cohorts identification). In particular, an investigator must be able to run queries in MedCo by logically combining clinical and genomic concepts encoded by a medical ontology and obtain the number of patients per site satisfying the research criteria. Also, MedCo must enable different query breakdowns such as distribution of patient counts per age, gender, ethnicity. More formally, an investigator must be able to perform aggregate SQL queries such as "`COUNT(patients) FROM dataset WHERE * AND/OR * GROUP BY *;`" and selection

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

SQL query such as "`SELECT(patients) FROM dataset WHERE * AND/OR * GROUP BY *;`" where '`*`' represents any possible concepts/codes in the ontology.

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

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

3 MedCo Building Blocks

MedCo is the first operational system that combines established open-source technologies from both the biomedical informatics community (i2b2 [13] and SHRINE [22]) and the privacy and security community (UnLynx [9]) in order to enable privacy-preserving sharing of clinical and genomic distributed data. In this section, we provide a high-level description of these technologies and their main features that we use as MedCo building blocks.

3.1 Data Model from i2b2

Informatics for Integrating Biology and the Bedside (i2b2) [13] is the state-of-the-art clinical platform for enabling secondary use of electronic health records (EHR) [13]. It is designed to enable investigators to perform queries on an enterprise data-repository in order to find sets of patients that would be of interest for further clinical research studies.

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

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

Queries are executed as SQL statements by the data repository (or CRC) cell that returns the aggregate number of patients meeting the research criteria.

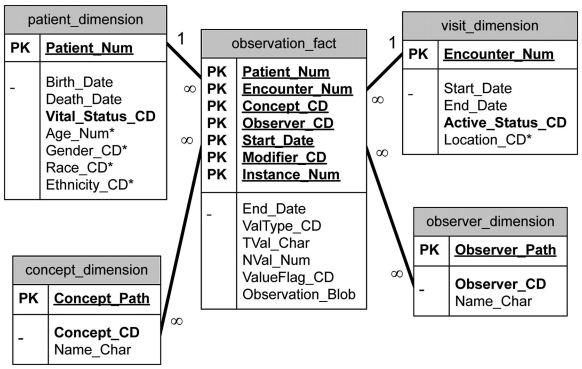


Fig. 2: **i2b2 data model [13].** It consists of a central `observation_fact` table and four dimension tables. Each row of the `observation_fact` table stores one clinical observation for a given patient. 'PK' stands for 'primary key' while 'CD' stands for 'code'.

3.2 Interoperability Layer from SHRINE

The Shared Health Research Information Network (SHRINE) [22] is the state-of-the-art framework that enables investigators to search patients' data from the 'limited data set (LDS)' across multiple independent clinical sites. SHRINE is currently deployed in at least six networks in the United States. It is built on top of i2b2 and its purpose is to connect distributed i2b2 instances from various clinical sites through an interoperability layer based on a common ontology. Such a common ontology is translated into each local site's ontology at query time, thus hiding the complexity of the local databases from the rest of the network. SHRINE comprises three main components:

- The *Adapter*: It is a Web-service designed as an i2b2 cell in order to be fully integrated within the i2b2 hive. It translates the investigator's query made through the common SHRINE ontology into a format that matches the site's local databases. In a fully decentralized network, an adapter must be deployed at each clinical site that uses SHRINE.
- The *Query Aggregator*: It is responsible for (i) broadcasting the investigator's query to each of the adapters in the network and (ii) receiving from each clinical site the count of patients satisfying the query to be sent back to the investigator. At least one clinical site in the network must deploy a query aggregator that will serve as query entry-point in the system.
- The *Web Client*: It provides a Web-based user interface through which the investigator can access the system and build i2b2-type queries by using the common SHRINE ontology. The Web client must be deployed together with the Query Aggregator.

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

3.3 Privacy-Preserving Distributed Protocols from UnLynx

UnLynx is the latest and most advanced general framework for privacy-conscious sharing of distributed sensitive data [9]. Its purpose is to enable a set of data providers to collectively protect the confidentiality of their sensitive data in the *anytrust* threat model [23], by encrypting them with a collective public key generated by a group of independent servers forming a collective authority (or "cothority"). Due to the use of additively homomorphic encryption (ElGamal on elliptic curves), users can still perform simple statistical queries directly on the encrypted data by relying on a set of secure distributed protocols run within the cothority. When a query comes to UnLynx, each data provider uploads the requested ciphertexts to the cothority, that securely processes them in order to obtain an encrypted query result. Such a result can eventually be decrypted only by the user who issued the initial query. UnLynx is designed to be modular, allowing the addition and removal of security/privacy features depending on the performance and security requirements. MedCo relies on three of the UnLynx main protocols:[3]

- **Distributed Deterministic Tag (DDT) Protocol.** The DDT protocol enables a set of cothority servers to tag (with deterministically encrypted values) data probabilistically encrypted under the cothority collective key, without ever decrypting them. The purpose of this protocol is to enable equality-matching queries on probabilistically encrypted data that otherwise would not be possible.
- **Distributed Verifiable Shuffling (DVS) Protocol.** The DVS protocol enables a set of cothority servers to sequentially shuffle probabilistically encrypted data so that the outputs cannot be linked back to the original ciphertexts.
- **Distributed Key Switching (DKS) Protocol.** The DKS protocol enables a set of cothority servers to convert a ciphertext generated with the collective public key of the cothority into a ciphertext of the same data generated under any known public key, without ever decrypting them.

4 MedCo Core Architecture & Protocol

In this section, we provide a detailed description of MedCo (Figure 3). We begin by explaining the system initialization and the data ingestion phases in which clinical sites collectively encrypt their sensitive data and store them in the i2b2 data model. We then describe the secure query workflow that enables an investigator to efficiently query the encrypted data stored in independent i2b2 databases.

In the remainder of the paper, we assume elliptic curve notation where $\mathcal{E}$ denotes an elliptic curve over the prime field $\mathbb{GF}(p)$ and G designates its base point. We denote as $E_K(m)$ the ElGamal probabilistic encryption of a message m under a public key $K = kG$, where k is the secret key, while $DT_s(m)$ denotes its deterministic tag under a secret key s.

4.1 System Initialization

Clinical sites store unencrypted patient-level clinical and genomic data in their private EHR systems and are willing to share these data by securely exposing them to internal/external investigators through an i2b2 research data warehouse. Therefore, we assume that

[3] For a more detailed description of these protocols, we refer the reader to the original paper [9].

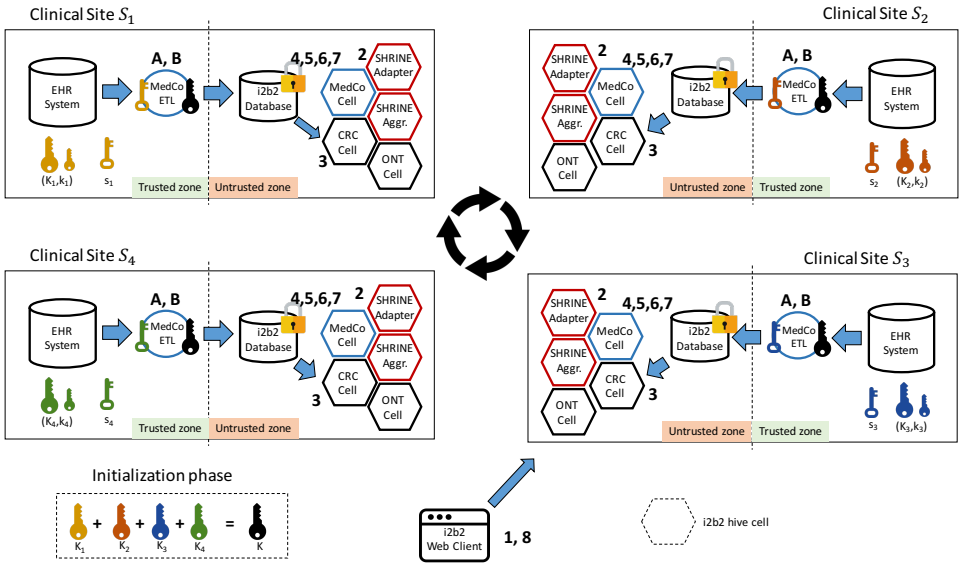


Fig. 3: **MedCo Core.** High-level representation of MedCo core phases with 4 clinical sites including *system initialization*, *ETL phase* (steps A and B) and *secure query workflow* (steps 1-8). Components in black are standard i2b2 cells; Components in red are SHRINE standard components; Components in blue are new MedCo components.

each site has already deployed an instance of the i2b2 hive on top of its own data warehouse and that SHRINE is also installed in order to provide the interoperability layer necessary for connecting data encoded with local ontologies at the different sites. During this initial phase, each clinical site (S_i) generates a pair of ElGamal cryptographic key (k_i,K_i), where $K_i = Gk_i$, along with a secret s_i. Then, all sites combine their ElGamal public keys in order to generate a single collective public key $K = \sum_i K_i$ that will be used to encrypt the data.

4.2 Data Extraction Transformation and Loading

During the data-ingestion phase, a.k.a. *extraction transformation and loading (ETL)*, each site extracts patient-level data from its private EHR system and transforms them by following the i2b2 data model representation in Figure 2. In the i2b2 data model, the private information that must be protected from an untrusted third party consists of the set of clinical observations that are considered to be sensitive or identifying for a given patient. Those are usually represented by a subset, $\mathbb{X}$, of ontology concepts encoded with ontology codes (`Concept_CD`) in the `observation_fact` table. Hence, before loading the data into the i2b2 data warehouse, each site starts an encryption phase consisting of two steps:

A. Generation of dummy patients: Each site generates a set of dummy patients with plausible clinical observations specifically chosen so that the distribution of ontology codes across patients, in the `observation_fact` table, is as close as possible to the uniform distribution. We explain the rationale behind such a step in detail in Section 5. To distinguish the real patients from the dummies, each site also generates a binary flag to be appended to the demographic information in the `patient_dimension` table. Such flag is set to 1 for real patients and to 0 for dummy patients.

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

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

4.3 Secure Query Workflow

We assume each investigator that uses MedCo has a pair of ElGamal cryptographic keys (k_u,K_u) and, optionally, an initial differential privacy budget ϵ_u. The purpose of such a budget is to limit the number of queries an investigator can run on the system so that ϵ_u-differential privacy can be guaranteed. The proposed query workflow is illustrated in Figure 3 and comprises the following steps:

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

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

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

4. Result Aggregation (optional): Once the patient numbers and the encrypted flags are fetched from the local i2b2 database, the MedCo cell homomorphically aggregates the

flags in order to obtain the encrypted local patient count $E_K(R_i)$ at each site. Because of the null contribution of their encrypted flags (i.e., $E_K(0)$), dummy patients are cancelled out from the local patient-count during the aggregation. Let $E_K(f_i^j)$ be the encrypted flag of the j-th patient in site S_i, then $E_K(R_i) = E_K(\sum_{j\in\phi} f_i^j) = \sum_{j\in\phi} E_K(f_i^j)$, where ϕ is the set of patients satisfying the query.

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

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

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

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

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

5 Dummy-Addition Strategies

For cohort-exploration queries, the deterministic encryption of the ontology codes applied during the ETL phase (see Section 4.2) avoids dictionary attacks by any subset of colluding HBC sites due to the distribution of the secrets s_i used in the DDT protocol. Nevertheless, a *Dummy-Patients Generation* step is required prior to encryption in order to avoid the unintended leakage of (i) the ontology code distribution and (ii) the query result. In this section, we analyze the optimal dummy-generation strategy to achieve this goal.

We assume, without loss of generality, that each patient has a different set of observations; if there were equal patients in the database, fake ontology codes could be added to make them different. The leakage to HBC sites can be estimated by calculating (i) the adversary's equivocation (a.k.a. conditional entropy) on the ontology codes of the `observation_fact` table given their tagged versions, as an average measure, and (ii) the smallest anonymity set of the ontology codes, as a worst case measure. The higher the equivocation

and the larger the anonymity set is, the lower the leakage is. For this exposition, we will focus only on the relation between patients and occurrences of ontology codes, leaving aside the temporal dimension, and we will follow the toy example shown in Figure 4. This figure represents the (horizontally) folded version of the (vertical) `observation_fact` table, therefore coding each patient as a row, each ontology code as a column, and each observed (resp. unobserved) code in a patient as a "1" (resp. "0") in the corresponding cell.

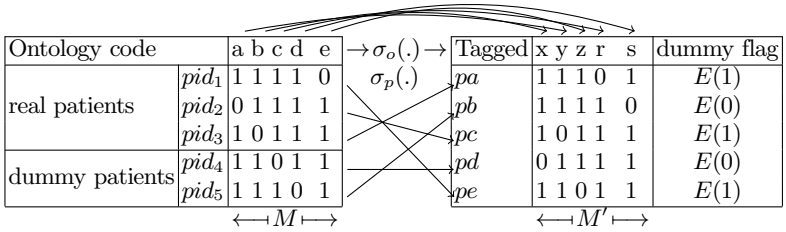


Fig. 4: **Toy example.** Ontology code mapping to real and added dummy patients with pseudo-identifiers pid_i, and ontology codes a, b, c, d, e. pa, pb, pc, pd, pe are the randomly sorted version of the patient pseudo-identifiers, and x, y, z, r, s are the shuffled and deterministically encrypted (tagged)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

More formally, let us define the matrix that associates ontology codes with patients as the tuple of a random binary matrix $\mathcal{M}$ where each row can be either a real or a dummy patient and each column represents one ontology code and two functions, σ_p and σ_o, that respectively map the patient pseudo-identifiers (pid_j in Fig. 4) to the rows (pa,pb,pc,pd,pe in Fig. 4) and the observed ontology codes (a,b,c,d,e in Fig. 4) to the columns (x,y,z,r,s in Fig. 4). These maps represent the shuffling applied to patients before they are assigned their pseudo-identifiers, and the shuffling and deterministic encrypted tag applied to ontology codes before they are loaded into the i2b2 database. In order to focus on the practical leakage of the deterministically encrypted database, let us assume that the tagging and the probabilistic encryption of the dummy flags do not leak anything about their inputs (their trapdoors cannot be broken), even if they are based on computational guarantees. Therefore, the adversary (each of the sites) observes the realization of the row- and column-permuted matrix: $\mathcal{A} \equiv [\mathcal{M}' = M']$, and her equivocation, with respect to the original information given $\mathcal{A}$, can be expressed as

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

$$\overset{(a)}{=} H(\sigma_o|\sigma_p,\mathcal{A}) + H(\sigma_p|\mathcal{A}) \overset{(b)}{\leq} H(\sigma_o|\mathcal{A}) + H(\sigma_p) \overset{(c)}{\leq} H(\sigma_o) + H(\sigma_p). \tag{2}$$

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

The last term of the equivocation can be maximized by making the dummy rows indistinguishable from the real patients; i.e., drawn from the same distribution. Empirically, this means that all the patients, real or dummy, have the same type of distribution, and

the contents of the rows are independent of the position of the dummy patients in the list. This also makes the two permutations independent of each other even when conditioned on the contents of M' (step (b) in (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 codes 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 codes through the added dummies; the geometric interpretation of this flattening is that any column permutation can be cancelled out by a row permutation, such that it is not possible to univocally map any ontology code to any column in M'. In our toy example, it can be seen that due to the two added dummies, any fixed query yields the same number of patients independently of the permutation applied to the query terms, which gives a complete indistinguishability between all the tagged ontology codes even in light of the matrix M'. It must be noted that the unobserved codes do not have to be added to the table, as the adversary does not have a priori knowledge of which is the subset of observed codes, only its cardinality. Also, this strategy fully breaks the correlation between ontology codes; for example, if the site added only one dummy with codes a,b,e to the real patients in Fig. 4 the individual appearance rate of the codes would be flattened, but it would leak that there is a correlation between the codes c and d, that could be identified in the encrypted matrix through an l_p-optimization attack [14].

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

This strategy guarantees the maximum uncertainty for the adversary for an arbitrary real distribution of codes across patients, but it generates a combinatorial number of dummies, which is not feasible in general (unless the number of observed codes is very low); but if some assumptions can be made about the code joint distribution, we can simplify the strategy. If dependencies are only found within small groups of codes, being the groups mutually independent (that is the case for genomic information and dependencies found inside subsets of localized variants), it is possible to constrain the needed number of dummies by applying the same dummy-addition strategy in a restricted block-wise fashion. In order to flatten only the histogram of group weights, we group codes in independent blocks of size $n' \ll n$ and apply the dummy-generation permutation to the blocks (inter-block), but not to the contents of each block, until the block distribution is flat, therefore reducing the needed number of dummy rows. This trade-off strategy creates an "anonymity set" of ontology codes of size n/n' in such a way that the adversary cannot distinguish between the set of codes inside different blocks. The drawback is that the equivocation is reduced, as the resulting joint distribution of the ontology code observations is only flat across blocks, but not inside each block. In the worst case in

terms of leakage (fully correlated codes within each block) the achievable adversary's equivocation becomes $H(\mathcal{M},\sigma_o,\sigma_p|\mathcal{A}) = H(\sigma_o|\sigma_p,\mathcal{A}) + H(\sigma_p|\mathcal{A}) \leq H(\sigma_{o,n/n'}) + H(\sigma_p)$, where $\sigma_{o,n/n'}$ are the permutations of the n/n' blocks of n' codes each. This bound is achieved when the blocks are mutually independent, so the best partitioning strategy consists in keeping correlated codes inside the same block. If fully independence between codes can be assumed ($n'=1$), it can be seen that flattening the observations histogram leads to the same maximum attacker equivocation as the complete permutation strategy (Eq. (2)), but with a much lower number of added dummies. In order to further reduce this number, it is possible to set a minimum anonymity set size m for the codes and add dummies to water fill the observation histogram (block-wise flat, instead of fully flat) until each code has at least other $m-1$ codes featuring the same number of observations.

Finally, it must be noted that whenever a site's i2b2 database is updated, dummies can be regenerated (and encryptions and tags re-randomized) when the ETL process (see Section 4.2) is run again for the whole updated database. The DDT protocol uses a different fresh randomness, so that the codes from the updated database cannot be linked back to the codes of the old one.

6 Privacy & Security Analysis and Extensions (MedCo+)

The main privacy and security requirements for MedCo are summarized in Section 2.3. In this section, we briefly discuss and analyze the fulfillment of these targets for MedCo, and we revisit possible extensions for more stringent requirements.

Security in MedCo is based on the cryptographic guarantees provided by the underlying decentralized data-sharing protocols (from UnLynx) and the adoption of well-established security practices when coding the interfaces with the i2b2 backend and the SHRINE interoperability layer. All input sensitive data are either deterministically (ontology codes) or probabilistically (dummy flags) encrypted with a collectively maintained key, such that they cannot be decrypted without the cooperation of all sites, thus guaranteeing confidentiality and avoiding single points of failure. For the full step-by-step security analysis of UnLynx, we refer the reader to [9]. Following this analysis, paired with the dummy strategy described in Section 5, it can be seen that MedCo covers the unlinkability requirement for the query results, thanks to the UnLynx DVS protocol; and it protects their confidentiality, as only the authorized investigator can decrypt the query results thanks to the UnLynx DKS protocol. Conversely, MedCo also enables the application of differentially private noise to the results to avoid membership inference attacks, and, thanks to the proposed dummy strategy, it guarantees confidentiality of the data also against all the clinical sites that participate in the system.

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

- Query confidentiality: In the basic MedCo system presented in Section 4, HBC sites can link the ontology codes used through different queries, as the applied deterministic tag is the same for all the queries. In the case that query confidentiality is also a requirement (e.g., investigators from pharmaceutical companies), it is possible to address it by probabilistically encrypting ontology codes during the ETL phase and by deterministically tagging the obtained ciphertexts with a fresh secret for each new query. The effective encryption key is different for each fresh run of the DDT protocol, so it is not possible

to link the query terms between different runs of the shuffling-DDT. When this modified system (which we denote MedCo+) is paired with the proposed dummy-addition strategy, the terms between queries are indistinguishable and unlinkable, at the cost of transferring and tagging the subset of the encrypted database involved in the query.

- **Malicious sites:** MedCo's threat model assumes HBC sites to be credible and plausible assumption, based on the damage to reputation that a clinical site would suffer if it misbehaves in a collective data-sharing protocol. Nevertheless, it is possible to cope with malicious clinical sites by using UnLynx's proof generation protocols [9], which produce and publish zero-knowledge proofs for all the computations performed at the clinical sites, so that the proofs can be verified by any entity in order to assess that no site deviated from the correct behavior. UnLynx features zero-knowledge proofs for the DVS, DDT and DKS protocols, and the addition of differential privacy noise. Although this solution yields a hardened and resilient query protocol, the cost of producing all proofs causes a typically unacceptable burden in regular data sharing applications, for which the basic proposed MedCo covers all fundamental privacy requirements while yielding a very competitive performance, as shown in the next Section.

7 Deployment and Evaluation

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

7.1 Oncology Use-Case

Being able to compare mutation profiles between patients across different clinics and identify those with a similar molecular profile is of critical importance for guiding treatment decision in oncology. Similarly, in clinical research, the ability to compare multiple patients with the same mutation profiles enables robust hypothesis generation and testing. The power of such an approach increases with the number of mutation profiles that can be queried. Therefore, being able to share somatic mutation information among hospitals and institutions is an absolute requirement. Yet, privacy and security concerns make such sharing extremely difficult, if not impossible. For these reasons, we tested MedCo on cancer genomic and clinical data from cBioPortal [5] by performing typical queries for oncogenomics, of which we report here two representative examples:

- **Query A:** *Number of patients with skin cutaneous melanoma AND a mutation in BRAF gene affecting the protein at position 600.* About half of melanoma patients harbor a mutation in the BRAF gene at position V600E or V600K and can be treated by the BRAF inhibitor *vemurafenib* [1]. The proportion of mutated BRAF melanoma is therefore an important benchmark for a clinic or hospital.

- **Query B:** *Number of patients skin cutaneous melanoma AND a mutation in BRAF gene AND a mutation in (PTEN OR CDKN2A OR MAP2K1 OR MAP2K2 genes).* This query is based on the fact that patients treated with *vemurafenib* develop resistance through mutations activating the *MAP kinase* pathways [24]. When facing drug

resistance, finding another patient with a similar mutation profile could bring invaluable information for clinical decisions.

We used genomic and clinical datasets obtained from a skin cutaneous melanoma study [4, 11] of 121 patients with 9 clinical attributes and an average of 1,978 genetic mutations (239,286 observations in total).

7.2 Implementation

We developed MedCo as three components that fully integrate i2b2 [13], SHRINE [22] and UnLynx [9]: specifically (i) a new i2b2 server cell, developed in Java code, responsible for the MedCo business logic described in Section 4 and for interacting with UnLynx for the execution of secure distributed protocols, (ii) a new i2b2 Web-client plugin, developed in Javascript, enabling a user to generate queries involving somatic mutations through an annotation-based search engine, and to encrypt sensitive codes in the query directly in the browser, and (iii) an ETL tool, developed in Go 1.8, responsible for extracting genomic and clinical data from a raw tab-separated file, encrypting them with the ElGamal collective key and loading them into the i2b2 data model.

Encrypted data were stored in the i2b2 data model with PostgreSQL [16]. In particular, somatic mutations were stored as deterministically encrypted observations in the `observation_fact` table encoded by the combination of their chromosomal position, reference allele and mutated allele. Genetic annotations (e.g., gene names) were stored in the `concept_dimension` table.

7.3 Experimental Setup

To avoid inconsistencies in the results, potentially caused by the heterogeneous setting of the EPFL-CHUV-UNIL network (i.e., different firewalls, servers, and access control mechanisms), and to obtain a fair comparison with the standard unprotected i2b2/SHRINE deployment, we ran our evaluation within an isolated environment. Such an environment comprises 3 servers interconnected by 10Gbps links and featuring two Intel Xeon E5-2680 v3 CPUs with a 2.5GHz frequency that support 24 threads on 12 cores, and 256GB RAM. We note, however, that bioinformaticians in the EPFL-CHUV-UNIL sites had similar user experiences. Each server hosted the SHRINE Web client, the i2b2 hive including the SHRINE adapter, query aggregator and the new MedCo cell, the i2b2 database, and the UnLynx server back-end. To set up our system and facilitate its deployment, we used Docker [12]. The default database contains the public dataset described in Section 7.1. We used UnLynx ElGamal encryption on the Ed25519 elliptic curve with 128 bit security.

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

1. Runtime for varying database size: For this setup, we ran query A (see Section 7.1) for different database sizes and measured the total runtime of MedCo, comparing it with: (i) The *insecure* i2b2/SHRINE implementation where all data are stored in clear, and (ii) the more *secure* version of MedCo, MedCo+, where we enhanced MedCo in order to protect also query confidentiality, as described in Section 6.

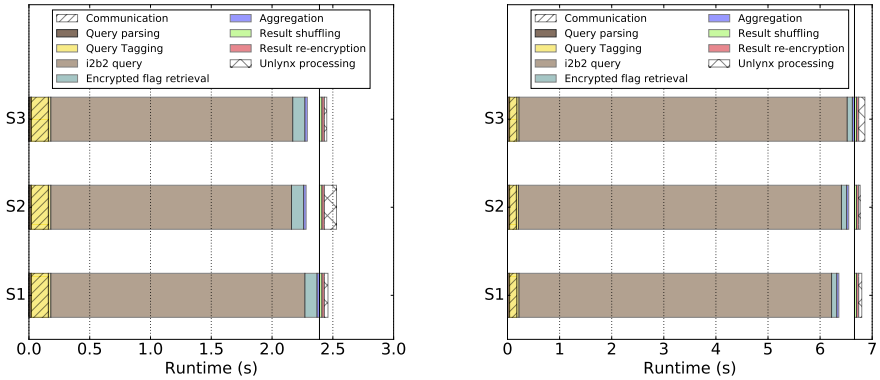
2. Runtime for varying number of sites: We studied MedCo's runtime for query A with varying number of sites in the network. We assumed that for each new site a new server is added to the system. We considered 3, 6, 9 and 10 sites with each site having a third of the original database (approximately 82.000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

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

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

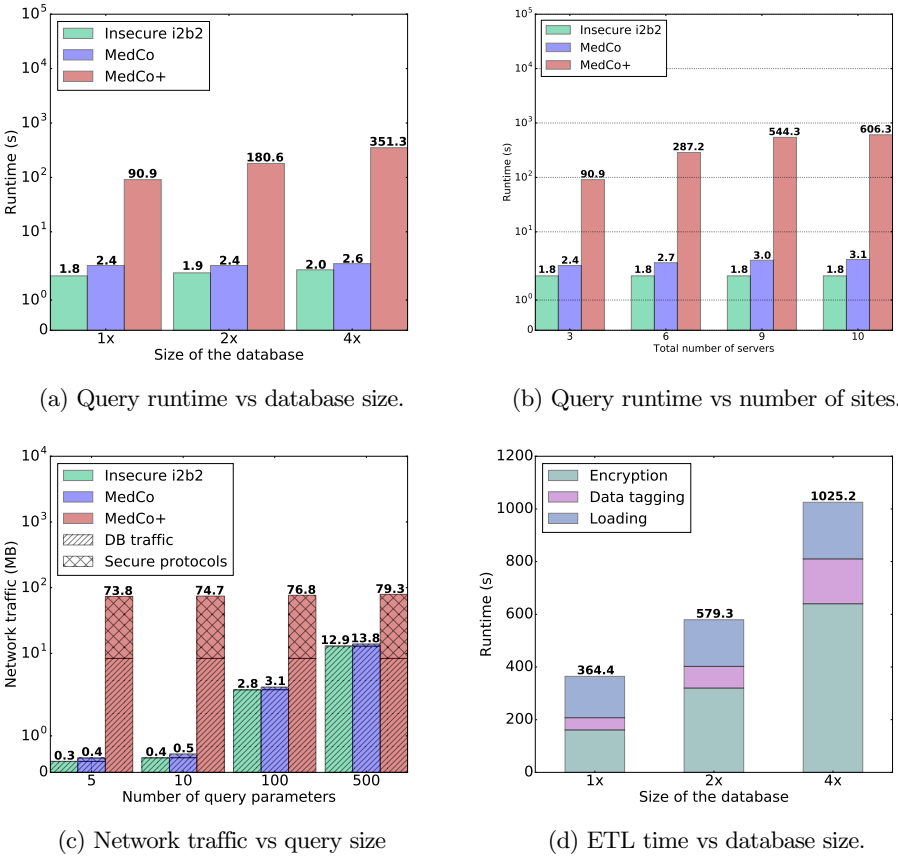
7.4 Performance Results

Here we evaluate the raw overhead without dummies of the protocols featured in MedCo and MedCo+ with respect to the insecure i2b2; we analyze the impact of dummies in Section 7.5. Figure 5 provides query-workflow breakdowns for both query A and query B. Because they are negligible, we do not account for the query parsing and encryption/decryption times in the Web client, for the time to broadcast the query from the SHRINE query aggregator to the different sites, and for the result obfuscation. Unexpectedly, results show that the i2b2 query to the `observation_fact` table is the most expensive operation in MedCo as it depends on the total number of observations. This time is also linear in the number of ontology codes in the query and it is inherent to the standard i2b2 database management for SQL-queries to the `observation_fact` table. Fetching the encrypted patients flags from the `patient_dimension` table, before homomorphic aggregation, can be also expensive as it depends on the number of patients satisfying the search criteria. The deterministic tagging of query encrypted codes is also linear in the number of ontology codes in the query, as each encrypted code has to be sequentially modified twice by each site in the network. Such a process takes more time for query B than for query A, as they consist of 79 (77 mutations and 2 clinical attributes) and 6 (4 mutations and 2 clinical attributes) query attributes, respectively. Differently, the homomorphic aggregation depends on the number of patients satisfying the query and it can be extremely fast for rare combinations of somatic mutations and clinical



(a) Query A (b) Query B

Fig. 5: **Query-workflow breakdown.** Each site is represented as S1, S2, S3. The vertical black line signals the point where each node has to wait for the others before it can proceed.



(a) Query runtime vs database size.

(b) Query runtime vs number of sites.

(c) Network traffic vs query size

(d) ETL time vs database size.

Fig. 6: **MedCo's performance results for setups 1-4**

attributes. For queries A and B it takes around 0.68 and 0.40 milliseconds as only around 16 and 9 patients per site satisfy the research criteria. The remaining secure distributed operations introduced by MedCo depend on the number of sites in the network but they are negligible as they involve only one ciphertext, i.e., the encrypted query result.

Figure 6 shows the performance results for the four above-mentioned setups (1-4). The measurements are averaged out between servers. Subfigure 6a refers to the first setup and reports the time required to execute query A with different database sizes under different scenarios. Besides the normal database ('1x'), we chose two others with twice ('2x') and four times ('4x') more observations. These two additional databases were obtained by replicating the original one. In each case, the data were evenly distributed among the three sites, thus obtaining, for the three cases '1x', '2x' and '4x', around 80k, 160k and 320k observations over 40, 80 and 160 patients per site, respectively. Results show that MedCo is extremely efficient and comparable in terms of performance to the insecure version of the i2b2/SHRINE implementation. MedCo's overhead with respect to the insecure i2b2/SHRINE is almost constant when the database size increases as the privacy-preserving protocols introduced by MedCo depend mostly on the number of queried codes and the size of the resulting patient set. We can also observe that MedCo+ has a relatively higher runtime cost as a counterpart for achieving query unlinkability,

because all the observations in the `observation_fact` table have to be deterministically tagged at runtime for each new query. However, we note that such a privacy enhancement might be necessary only under specific circumstances (e.g., when an investigator from a pharmaceutical company is using the system).

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

Subfigure 6c shows the network traffic incurred when increasing the number of queried ontology codes. The traffic was split between two main components: i2b2 database traffic, and traffic introduced by MedCo's secure protocols. As expected, network traffic increases linearly with the number of queried ontology codes for both MedCo and the insecure i2b2/SHRINE, whereas it is almost constant for MedCo+. Also, the database operations dominate the traffic in MedCo. Yet, the traffic caused by the secure protocols is not negligible, as encrypted codes in the query are broadcasted to each site in the network. In MedCo+, the traffic of secure protocols becomes predominant with respect to the database traffic, as all observations in the database are broadcasted across the whole network in order to be deterministically tagged, regardless of the number of codes in the query.

Subfigure 6d shows the results for the pre-processing or ETL phase, including the encryption of the data (deterministic tagging for ontology codes and ElGamal encryption of patients' binary flags) and the data loading into the database. Results show that the ETL phase is costly and its time increases linearly with the amount of data in the database. However, it is important to mention that this phase is only executed once at each site, offline.

Finally, Figure 7 shows MedCo's ability to scale with respect to the insecure i2b2/SHRINE for a database sizes of 100 and 1,000 times larger than the original database. Again, in both cases, patients were evenly distributed across sites thus obtaining for the two cases ('100x' and '1000x') around 8M and 80M observations over 4,000 and 40,000 patients, respectively. Results are impressive, as the total query runtime (for query A) for the '1000x' case, which can represent the standard size of a clinical site's database, is still within 10 seconds. As expected, MedCo's overhead with respect to the insecure i2b2/SHRINE becomes more significant in the 1,000x case, as around 16,000 patients' encrypted flags need to be fetched from the `patient_dimension` table. This is due the artificial carbon-copy replication of the data and it is unlikely to happen for a typical oncology database with an equivalent size but without replicated patients. Similarly, also the time required for the homomorphic aggregation increases as the number of patients satisfying the query has increased by 1,000 times. However, we note that the most expensive operation still remains the i2b2 querying time that is independent of any additional privacy-preserving feature added by our solution.

7.5 Storage Overhead

In the unprotected i2b2 data model, ontology concepts are generally represented by 64-bit integers, whereas MedCo's deterministic encryption converts each code into a 32-bytes tag. Hence, the storage overhead introduced by MedCo's encryption is in the order of 4 times. Depending on the specific distribution of ontology codes across patients, a varying number of dummy patients must also be considered. In the tested oncology use-case, we assume independent codes and follow the dummy addition strategy described

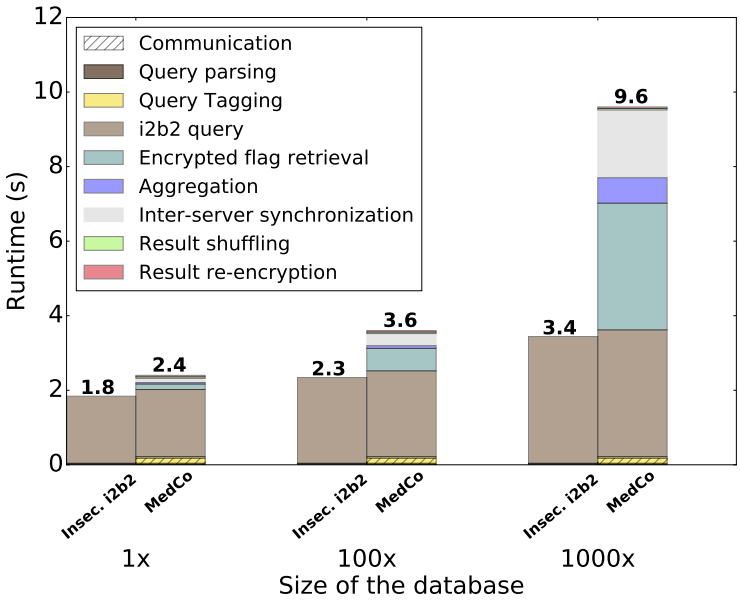


Fig. 7: MedCo's scalability test: Query runtime vs database size (for a database up to 1,000 times larger the original database).

in Section 5. Because our original dataset is very sparse, with 238,363 ontology codes most of which (around 98%) are observed only once, the adversary's equivocation is already very high without the need of dummies (around 3.76 Mbits). We focus, therefore, on the minimum anonymity set for the observed codes. The extremely low number of common mutations and their small anonymity set (some of them having a unique rate) make common mutations easily identifiable by an adversary. Consequently, for maximizing both the equivocation and the minimum anonymity set (238,363 for all codes), the number of necessary dummy patients should be increased to 7,400, with 14 million new observations in total (approx. 62x increase factor in the number of rows of the `observation_fact` and the `patient_dimension` tables). Yet, by trading off the size of the smallest anonymity set, it is possible to reduce the number of needed dummy patients to 350 with roughly 614,000 observations, yielding a table increase factor of around 3.6x, which is practical. This guarantees a minimum anonymity set of 10,000 codes, and an adversary's equivocation close to the achievable upper bound. Due to space constraints, we do not show here the performance results of MedCo with dummies, but they can be assimilated to the '4x' case in Figure 6a.

8 Related Work

Among the operational systems for sharing of clinical or genomic information, SHRINE [22] and the GA4GH Beacon Network [10] are certainly the most advanced and widespread. However, as opposed to MedCo, they provide limited privacy guarantees (only ad-hoc result obfuscation) and no protection of data confidentiality besides standard access control, thus significantly restraining the amount of sharable information.

Besides that, and to the best of our knowledge, there are mainly two recent works dealing with privacy-preserving queries in distributed medical databases. The first one, PRINCESS [6], is based on trusted hardware: the sites encrypt all their data under AES-GCM (Advanced Encryption Standard - Galois Counter Mode) and send them to an enclave running in a central server, featuring an Intel SGX processor; this server decrypts and processes them, enabling the computation of statistical models. Compared to our work, PRINCESS can be more versatile in terms of allowed computations, but it presents a single point of failure (the central server), and centralizes all trust in the enclave and in the attestation protocol provided by Intel. Moreover, the memory restrictions of the enclave limit the scalability of the scheme, requiring compression and batching techniques to enable processing of large genomic data, for which MedCo scales much better.

The other recent approach, SMCQL [3], is based on secure two-party computation; it introduces a framework for private data network queries on a federated database of mutually distrustful parties. SMCQL features a secure query executor that implements different types of queries (e.g., merge, join, distinct) on the distributed database by relying on garbled circuits and Oblivious RAM (ORAM) techniques. Whereas this work features truly decentralized trust, it does not scale well to scenarios with more than two sites, which are likely to happen in medical contexts with a high number of collaborating hospitals.

9 Conclusion

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

References

1. Ascierto, P.A., Kirkwood, J.M., Grob, J.J., Simeone, E., Grimaldi, A.M., Maio, M., Palmieri, G., Testori, A., Marincola, F.M., Mozzillo, N.: The role of braf v600 mutation in melanoma. Journal of translational medicine 10(1), 85 (2012)
2. Athey, B.D., Braxenthaler, M., Haas, M., Guo, Y.: transmart: an open source and community-driven informatics and data sharing platform for clinical and translational research. AMIA Summits on Translational Science Proceedings 2013, 6 (2013)
3. Bater, J., Elliott, G., Eggen, C., Goel, S., Kho, A., Rogers, J.: Smcql: Secure querying for federated databases. Proc. VLDB Endow. 10(6), 673–684 (Feb 2017), https://doi.org/10.14778/3055330.3055334

4. Broad Institute: Skin Cutaneous Melanoma Datasets, http://www.cbioportal.org/study?id=skcm_broad#summary
5. Cerami, E., Gao, J., Dogrusoz, U., Gross, B., Sumer, S., Aksoy, B., Jacobsen, A., Byrne, C., Heuer, M., Larsson, E., et al.: The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. cancer discov. 2012; 2: 401–4. doi: 10.1158/2159-8290. Nat Methods 7, 92–93 (2012)
6. Chen, F., Wang, S., Jiang, X., Ding, S., Lu, Y., Kim, J., Sahinalp, S.C., Shimizu, C., Burns, J.C., Wright, V.J., et al.: Princess: Privacy-protecting rare disease international network collaboration via encryption through software guard extensions. Bioinformatics p. btw758 (2017)
7. Cupak, M.: Beacon network: A system for global genomic data sharing
8. EU Parlament: The EU General Data Protection Regulation (GDPR), http://www.eugdpr.org/
9. Froelicher, D., Egger, P., Sousa, J.S., Raisaro, J.L., Huang, Z., Mouchet, C., Ford, B., Hubaux, J.P.: Unlynx: A decentralized system for privacy-conscious data sharing. In: Proceedings on Privacy Enhancing Technologies. vol. 4, pp. 152–170 (2017)
10. for Genomics, G.A., Health: The beacon project, https://beacon-network.org/#/
11. Hodis, E., Watson, I.R., Kryukov, G.V., Arold, S.T., Imielinski, M., Theurillat, J.P., Nickerson, E., Auclair, D., Li, L., Place, C., et al.: A landscape of driver mutations in melanoma. Cell 150(2), 251–263 (2012)
12. Merkel, D.: Docker: lightweight linux containers for consistent development and deployment. Linux Journal 2014(239), 2 (2014)
13. Murphy, S.N., Weber, G., Mendis, M., Gainer, V., Chueh, H.C., Churchill, S., Kohane, I.: Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). Journal of the American Medical Informatics Association 17(2), 124–130 (2010)
14. Naveed, M., Kamara, S., Wright, C.V.: Inference attacks on property-preserving encrypted databases. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security. pp. 644–655. CCS '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2810103.2813651
15. Philippakis, A.A., Azzariti, D.R., Beltran, S., Brookes, A.J., Brownstein, C.A., Brudno, M., Brunner, H.G., Buske, O.J., Carey, K., Doll, C., et al.: The matchmaker exchange: a platform for rare disease gene discovery. Human mutation 36(10), 915–921 (2015)
16. PostgreSQL Global Development Group: PostgreSQL 10, https://www.postgresql.org/
17. Selby, J.V., Beal, A.C., Frank, L.: The patient-centered outcomes research institute (pcori) national priorities for research and initial research agenda. Jama 307(15), 1583–1584 (2012)
18. Swiss Academies of Arts and Sciences: Swiss Personalized Health Network, http://www.samw.ch/en/Projects/SPHN.html
19. The Global Alliance for Genomics and Health: A federated ecosystem for sharing genomic, clinical data. Science 352(6291), 1278–1280 (2016)
20. U.S. Department of Health & Human Services: The health insurance portability and accountability act (hipaa), https://www.hhs.gov/hipaa/index.html
21. U.S. Department of Health and Human Services : Breach portal: Notice to the secretary of hhs breach of unsecured protected health information. https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf, last Accessed: September 22, 2017
22. Weber, G.M., Murphy, S.N., McMurry, A.J., MacFadden, D., Nigrin, D.J., Churchill, S., Kohane, I.S.: The shared health research information network (shrine): a prototype federated query tool for clinical data repositories. Journal of the American Medical Informatics Association 16(5), 624–630 (2009)
23. Wolinsky, D.I., Corrigan-Gibbs, H., Ford, B., Johnson, A.: Scalable anonymous group communication in the anytrust model. Tech. rep., NAVAL RESEARCH LAB WASHINGTON DC (2012)
24. Yang, H., Kircher, D., Kim, K., Grossmann, A., VanBrocklin, M., Holmen, S., Robinson, J.: Activated mek cooperates with cdkn2a and pten loss to promote the development and maintenance of melanoma. Oncogene 36(27), 3842–3851 (2017)