Data & Knowledge Engineering 68 (2009) 1206–1223

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Formal anonymity models for efficient privacy-preserving joins

Murat Kantarcioglu [a,*], Ali Inan [a], Wei Jiang [b], Bradley Malin [c]

[a] University of Texas at Dallas, Department of Computer Science, USA
[b] Missouri University of Science and Technology, Department of Computer Science, USA
[c] Vanderbilt University, Department of Biomedical Informatics, USA

ARTICLE INFO

Article history:
Available online 22 July 2009

Keywords:
Privacy
Security
Anonymity
Data integration

ABSTRACT

Organizations, such as federally-funded medical research centers, must share de-identified data on their consumers to publicly accessible repositories to adhere to regulatory requirements. Many repositories are managed by third-parties and it is often unknown if records received from disparate organizations correspond to the same individual. Failure to resolve this issue can lead to biased (e.g., double counting of identical records) and underpowered (e.g., unlinked records of different data types) investigations. In this paper, we present a secure multiparty computation protocol that enables record joins via consumers' encrypted identifiers. Our solution is more practical than prior secure join models in that data holders need to interact with the third party one time per data submission. Though technically feasible, the speed of the basic protocol scales quadratically with the number of records. Thus, we introduce an extended version of our protocol in which data holders append *k*-anonymous features of their consumers to their encrypted submissions. These features facilitate a more efficient join computation, while providing a formal guarantee that each record is linkable to no less than *k* individuals in the union of all organizations' consumers. Beyond a theoretical treatment of the problem, we provide an extensive experimental investigation with data derived from the US Census to illustrate the significant gains in efficiency such an approach can achieve.

1. Introduction

Organizations are increasingly collecting detailed sensitive information on individuals in a wide range of environments. Until recently, the collection and analysis of person-specific data was performed locally, but scientific need and emerging policy frameworks are pushing organizations to share data beyond their borders. This trend may be best exemplified by the biomedical community and the growing trend in clinical genomics research in the US. In 2003, the US National Institutes of Health (NIH) formalized a data sharing policy for all sponsored investigators to facilitate the timely sharing of final research data generated through public funding. The policy was finalized such that investigators receiving at least $500,000 in any year of an NIH-funded project [1] must share the records that were studied to support scientific findings. In 2006, the NIH extended data sharing requirements in the context of "genome-wide" research, such that investigators receiving any funds for genome-wide association studies (GWAS) are required to share the studied person-specific records to a centralized databank managed by the NIH [2] for redistribution to other qualified researchers. The premise of these frameworks is to facilitate information reuse from costly projects. The challenge; however, is that the policies require organizations to share data in a "de-identified" manner, such that the identities of the research participants should not be inferable from the shared records.

* Corresponding author. Tel.: +1 972 883 6616; fax: +1 972 883 2399.
E-mail addresses: muratk@utdallas.edu (M. Kantarcioglu), inan@student.utdallas.edu (A. Inan), wjiang@mst.edu (W. Jiang), b.malin@vanderbilt.edu (B. Malin).

0169-023X/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2009.06.011

Various investigations have shown de-identified genomics and clinical data is often "re-identifiable" to named research participants through mechanisms that require nothing more than publicly-available resources [3–7]. These findings severely jeopardize the wide-scale availability of such databanks. In earlier work, we began to address this challenge by introducing a privacy-enhancing framework that permits data holders to submit and query biomedical data housed in a centralized repository managed by a third party [8]. The framework enables data holders to store person-specific data, such as DNA sequences, on a third party's server in an encrypted format. The cryptographic basis of the framework is homomorphic, which enables the third party to execute queries for researchers, such as counts, without decrypting the records. For example, a researcher can ask "How many DNA sequences contain pattern *X*?" and, while the third party will learn the result (i.e., the fraction of records that satisfy the criteria), it cannot learn which records satisfied the criteria.

The knowledge gained through count queries; however, does not support certain biomedical applications. For instance, in the health care realm, patients are mobile and their data can be collected by multiple locations, such as when a patient visits one hospital for primary care and a second hospital to participate in a clinical trial [5]. To facilitate robust biomedical investigations and prevent duplication of entries, it is beneficial to merge data that corresponds to the same patient. In traditional databases, such merges are achieved through joins on common attributes. The framework presented in [8]; however, is based on semantic security, so equivalent identifiers will appear different after encryption. Therefore, we need to develop a new protocol to achieve join queries.

In this paper, we exploit the fact that many organizations, such as health care providers, collect identifying information on their consumers. For instance, it is common for hospitals to use a patient's Social Security Number (SSN) and demographics for administrative purposes [9,10]. Such identifiers have been validated as excellent keys for data merging [11], but their disclosure is prohibited by organizations' policies and federal regulations. Despite restrictions, we can share SSNs, and other identifiers, in an encrypted manner for data merging purposes when the encryptions are semantically secure [12]. In this work, we demonstrate how to join patient-specific identifiers within the encrypted framework introduced in our earlier work [13]. Though our framework enables joins over semantically secure identifiers, the experimental analysis presented in Section 7 of this paper suggests that the protocol may be inefficient for large databases. Therefore, we propose an approach to speed-up the process by appending person-specific features according to a formal anonymity model, such as the concept of *k*-anonymity [14,15], which limits the number of potential joins that must be evaluated. The application of *k*-anonymity to protect the privacy of individuals in data shared beyond the collecting institution is not unreasonable. In fact, this and similar models have been proposed as a computational protection framework for health-related data. It has also been suggested that this approach satisfies the statistical standard of identity protection in the Privacy Rule in the Health Insurance Portability and Accountability Act (HIPAA), which regulates the dissemination of electronic patient data in the context of healthcare agencies [16]. Over the past several years, the model has been tailored to, and evaluated with, DNA sequences [3,17], the demographics of hospital patient's [18–20], personal information in pathology reports [21], as well as the geographic features inherent in data shared for public health and biosurveillance efforts [22]. The model has further been integrated into enterprise-level information management technologies, such as IBM's "Hippocratic Database" [23], which has been piloted in various healthcare environments around the world. It is on the aforementioned observations that we base our justification for applying *k*-anonymity to personal information for controlled information revelation when executing data joins.

Another motivation for using *k*-anonymity is to enable faster privacy preserving data integration without violating individual privacy. In other words, we want to make sure that during the privacy-preserving data integration process no individually identifiable information is revealed. Furthermore, we would like to provide privacy guarantees with respect to what is revealed. To achieve such formal privacy guarantees, we use the *k*-anonymity concept. Basically, we prove that all data that is revealed throughout the data integration process satisfies *k*-anonymity. Furthermore, we make sure that any sensitive attribute (e.g., disease information) is not revealed at all at any time. Since sensitive attributes are not revealed during the privacy-preserving data integration process, vulnerabilities of *k*-anonymity due to revelation of sensitive attributes do not apply in our case.[1] Therefore, there is no need to enforce stronger anonymity definitions that impose restrictions on the sensitive attribute distribution, such as *l*-diversity [25] and *t*-closeness [26].

This paper is an extension of the work presented at the 2008 International Conference on Privacy in Statistical Databases [13]. Compared to [13], in this paper, we provide many additional experimental results to show the effectiveness of our proposed techniques in various different scenarios. In addition, we provide extensive security and privacy analysis of our methods. Furthermore, we give formal proofs of our security and privacy guarantees.

The paper is organized as follows. In Section 2, we review related research in privacy-preserving data integration, with particular attention on its relation to the protocols proposed in this paper. We follow this section with a summary of the aforementioned framework within which our protocols are developed. Next, in Section 5, we present the secure join protocol and prove its protective properties. Section 6 then develops a more efficient protocol that integrates the secure protocol with a data anonymization model. The protocols are empirically evaluated in Section 7 with a real-world Census dataset. Finally, Section 8 discusses some lessons learned, and Section 9 concludes the paper.

Before commencing, we wish to stress the challenge of preserving privacy in biomedical data sharing is not unique to the United States. Across the globe, organizations are working to integrate person-specific DNA and clinical information from

[1] *k*-anonymous datasets might also reveal presence of individuals in the original dataset. δ-presence protects against such disclosures [24]. As long as anonymization involves generalization, extension of our methods to δ-presence is straightforward.

disparate facilities in the hopes of generating statistically significant research results [27–29]. We believe that the protocols offered in this paper (and our earlier framework) will demonstrate that biomedical data sharing and usage can be achieved in a privacy respective manner.

2. Related work

To date, privacy-preserving data integration (PPDI) research has spawned several frameworks [30,31]. PPDI frameworks generally address three basic privacy-preserving components: (1) schema matching, (2) data joins, and (3) query processing. Most PPDI research focuses on specific challenges associated with a particular component. However, among these components, schema matching has drawn significantly less attention. The most recent study towards private schema matching is [32].

Agrawal et al. [33] formalize a general notion of private information sharing across databases that relies on commutative encryption techniques. To avoid the high costs of secure function evaluation (SFE), specialized functions for intersection and equijoin operations were proposed. This study has opened the way to many other related protocols [34–37]. Compared to existing private information sharing work, the protocols we introduce in this paper require participants to have only single interaction, therefore resulting in less communication costs. Also, our protocols enable each data holder's records to be incrementally added to the central data repository.

Prior proposed protocols usually involve expensive cryptographic operations and do not scale for large data collections. To enable more efficient solutions, different solutions have been proposed by researchers. Pon et al. suggest hash-based noise addition techniques in [38]. In this method, the collision caused by hashing is the primary source of privacy. However, there are no guarantees on the amount of collision and therefore no trade-off between privacy and costs. Our method, on the other hand, relies on privacy guarantees of *k*-anonymity definition [14] such that less anonymity typically implies less privacy and lower costs. A similar method for performing private record linkage was proposed by Inan et al. in [39]. Unlike the protocols given in [39], our protocols enable incremental record addition to a central data repository.

In addition to the above approaches, another major area closely related to our work is private record linkage. Few methods for private record linkage have been investigated. Some initial approaches are motivated by the strict privacy requirements of e-health applications [40,41]. Most studies in the area focus primarily on efficient computation of alphanumeric distance functions. For example, [41] shows how N-grams can be built using secure hash functions. The work by Al Lawati et al. [42] aims at securely computing TFIDF scores. A major difference between record linkage and secure join operations is that, record linkage typically assumes there are no keys that uniquely identify individuals across distinct datasets. Instead, linkage relies on probabilistic methods and machine learning algorithms [43]. In secure join, however, the join condition need not be learned. So, the major concern is not accuracy but rather efficiency.

3. Secure architecture

The join protocols we propose in this paper are designed to function within the secure framework introduced in [8]. The framework was engineered to support (1) the secure transfer and centralized storage of person-specific biomedical records in a database and (2) response to user-issued queries as they would be performed on the original sequences. It incorporates four types of participants as depicted in Fig. 1: (i) data holders, (ii) data users, (iii) a data site (DS), and (iv) a key holder site (KHS). The latter two participants distribute the role of a third party and are crucial to the security components of the architecture. Specifically, the encrypted DNA and patient data is stored and processed at DS. In contrast, KHS manages the keys that encrypt patient information and queries, as well as the keys to decrypt the query results.

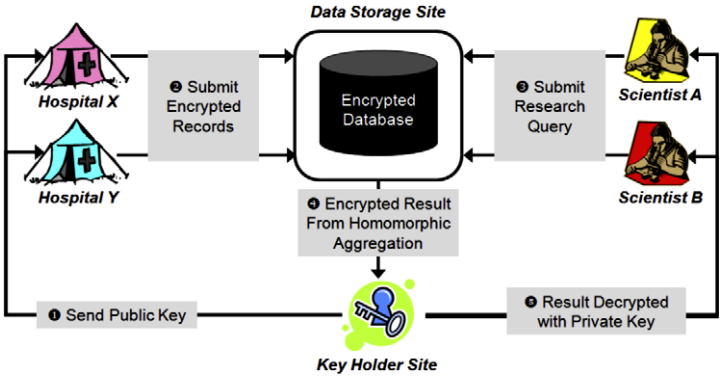


Fig. 1. General architecture.

As a running example, imagine that the set of data holders are hospitals and that the set of data users are biomedical researchers. We assume each hospital maintains patient-specific demographic information (e.g., sex, age), clinical information (e.g., medical diagnosis), and DNA records. The work-flow of the framework proceeds as follows:

Step 1 (Key Generation): First, KHS generates the cryptographic keys by which data holders will encrypt their records. Specifically KHS generates a ⟨public, private⟩ key pair, provides DS with the public key, and keeps the private key secret.

Step 2 (Data Encryption): When a hospital is ready to share its DNA sequences and other related patient information, DS sends the hospital its public key. Hospitals encrypt their records using the public key and send the results to DS. It is at DS, where the encrypted data will be queried and data mined by biomedical researchers. We assume that DS validates that the legitimacy of the data using authentication schemes described elsewhere [44–46].

Step 3 (Query Issuance): The set of queries that can be issued are known to the biomedical researchers. After the data is encrypted and stored at DS, a researcher sends a query for the database to DS.

Step 4 (Query Processing): DS executes the requested query and sends the encrypted results to KHS.

Step 5 (Result Decryption): KHS decrypts the result using the private key and sends it to the biomedical researcher.

Briefly, data stored at DS is semantically secure [12] (see the following section), so DS can learn the actual values only with the corresponding private key. However, KHS only issues DS a public key. KHS keeps the private key secret and does not share it with DS. As a result, DS is unable to discover the original patient information. Therefore, the data stored at DS are inherently secure against DS, as well as any biomedical researcher that issues queries in the framework. The same principle will be applied to support the security of the data integration protocols introduced in this paper.

The framework supports aggregation, or count queries, which are crucial to biomedical data mining tasks. In [8], we proved that DS and KHS can process the count queries of biomedical researchers without learning anything more than the query result. In this paper, we focus on the data integration tasks that occur between step 2 and step 3 discussed above.

We acknowledge that the framework is secure under the "semi-honest" model [47], such that we assume the participating parties follow the prescribed protocol, but can compute additional information through what is revealed during the protocol's execution and the final result. In the context of the framework, this implies that DS only asks KHS to decrypt *encrypted query results* as prescribed by the protocol. The semi-honest model is widely accepted in many data mining environments and its feasibility in our environment is supported by both technical and legal means. Technically, the framework's protocols are embedded in software, which is tamper-resistant. Legally, DS and KHS are trusted organizations that are contractually bound from violating the terms of data use.

4. Homomorphic cryptography

To achieve a simple and flexible architecture, we utilize a semantically secure public key encryption scheme (e.g., Paillier cryptosystem [48]).

In a public key encryption scheme, each participant maintains a pair of cryptographic keys: a private key and a public key. Generally speaking, a participant keeps the private key secret and publicly publishes the public key. For example, if Bob wants to send a message, or plaintext, to Alice, Bob can encrypt the message using Alice's public key and send her the encrypted message, or ciphertext. The ciphertext can only be decrypted by the corresponding private key, so Alice is the only one who can decipher the message from Bob.

Semantic security implies that it is infeasible for an adversary, with finite computational capability, to extract information about a plaintext when in possession of the ciphertext and the corresponding public encryption key.[2] The public key encryption scheme adopted in our architecture is probabilistic and possesses a homomorphic property. Informally, the homomorphic property allows us to compute the encrypted sum of two plaintext values through the corresponding ciphertexts. More formally, let $E_{pk}(.)$ and $D_{pr}(.)$ represent the encryption function with public key pk and the decryption function with private key pr, respectively. A secure public key cryptosystem is said to be probabilistic and homomorphic if the encryption function satisfies the following requirements:

Constant Efficient: Given a constant k and a ciphertext $E_{pk}(m)$ of m, we can efficiently compute a ciphertext of km, denoted as $E_{pk}(km) := k \times_h E_{pk}(m)$.

Probabilistic Encryption: Given a message $m, c_1 = E_{pk}(m)$ and $c_2 = E_{pk}(m), D_{pr}(c_1) = D_{pr}(c_2)$ but $c_1 \neq c_2$ with high probability. In other words, if a message is encrypted twice, with very high probability, the two ciphertexts are different.

Additive Homomorphic: Given the encryptions $E_{pk}(m_1)$ and $E_{pk}(m_2)$ of m_1 and m_2, there exists an efficient algorithm to compute the public key encryption of $m_1 + m_2$, denoted as $E_{pk}(m_1 + m_2) := E_{pk}(m_1) +_h E_{pk}(m_2)$.

[2] Semantic security implies that repeated encryptions of the same message are indistinguishable.

Our framework can be applied within any additively homomorphic cryptosystem. In this paper, we situate the framework within the Paillier cryptosystem [48] because it has relatively wide-scale adoption and standardization. Paillier is common in many secure distributed applications, such as secure voting [49] and privacy-preserving data mining algorithms [50,51]. As such, we can reuse existing code and implementations for our framework. Moreover, Paillier encryption has many variants that can be used to provide additional security properties such as threshold decryption, in which a private key is broken up into pieces and distributed among many participants. In the framework, we leverage the additive and multiplicative features of Paillier cryptosystems:

Adding Two Ciphertexts $(+_h)$**:** Given the encryption of m_1 and m_2, $E_{pk}(m_1)$ and $E_{pk}(m_2)$, we calculate the $E_{pk}(m_1+m_2)$ as follows:

$$E_{pk}(m_1)+_hE_{pk}(m_2)=E_{pk}(m_1)\cdot E_{pk}(m_2)\bmod n^2=((1+n)^{m_1}\cdot r_1^n)\cdot((1+n)^{m_2}\cdot r_2^n)\bmod n^2=E_{pk}(m_1+m_2\bmod n)$$

Ciphertext addition yields $E_{pk}(m_1+m_2\bmod n)$ to as a consequence of the modular operation.

Multiplying a Ciphertext with a Constant $(k\times_hE_{pk}(m_1))$**:** Given a constant k and the encryption of m_1, $E_{pk}(m_1)$, we calculate $k\times_hE_{pk}(m_1)$ as follows:, which are defined as follows:

$$k\times_hE_{pk}(m_1)=E_{pk}(m_1)^k\bmod n^2=E_{pk}(km_1).$$

5. Secure queries and the equijoin

In this paper, we focus on how to execute equijoin queries on the encrypted data stored at DS. Such queries are necessary for adding and integrating new datasets to the database already stored at DS. Here, we present a novel protocol to perform secure equality joins, termed as *Secure-Equijoin*. Our Secure-Equijoin protocol is applicable to non-interactive environments with independently encrypted attributes.

We adopt the following notation for this paper: $\theta^h=\{\theta_1^h,\ldots,\theta_\alpha^h\}$ is a dataset of α records in relational form, where each row (e.g., θ_i^h) indicates an individual's data, and each column represents an attribute of the individual (e.g., year of birth), at hospital h. θ_{ij}^h represents the value of the jth attribute of the ith individual in θ^h. E_{pk} and D_{pr} respectively are the Paillier's encryption and decryption functions with public key pk and private key pr generated by the KHS. $\theta^{h_1}\bowtie\theta^{h_2}$ indicates the join of datasets θ^{h_1} and θ^{h_2} on common attributes (e.g., encrypted primary key). Fig. 2 illustrates the use of above notations on a simple example.

Protocols 1 and 2 depict the pseudo-code of *Secure-Equijoin* as executed by DS and KHS, respectively. We assume a patient's record in a database is associated with identifying attributes, such as Social Security Number (SSN) or various demographics. The *Secure-Equijoin* is initiated by a hospital to encrypt the tuples in its database, θ^{h_1}, which is then sent to DS. After receiving encrypted tuples from two hospitals, $E_{pk}(\theta^{h_1})$ and $E_{pk}(\theta^{h_2})$, DS calculates $\theta^{h_1}\bowtie\theta^{h_2}$.

To evaluate the equijoin, DS securely calculates if two encrypted records are equivalent. Without loss of generality, we assume the join is performed using attributes $j_1,\ldots,j_m$. DS must inspect whether two encrypted tuples $E_{pk}\left(\theta_{i_1}^{h_1}\right)$ and $E_{pk}\left(\theta_{i_2}^{h_2}\right)$ match. Using the homomorphic properties of Paillier encryption, DS checks if $\left(\theta_{i_1j_1}^{h_1}=\theta_{i_2j_1}^{h_2}\right)\wedge\cdots\wedge\left(\theta_{i_1j_m}^{h_1}=\theta_{i_2j_m}^{h_2}\right)$ is true by checking if $M_{i_1,i_2}=\sum_{v=1}^m\left(\theta_{i_1j_v}^{h_1}-\theta_{i_2j_v}^{h_2}\right)\cdot r_v=0\bmod n$, where $r_1,\ldots,r_m$ are non-zero random values. DS calculates $E_{pk}(M_{i_1,i_2})$ on encrypted data via evaluating:

$$E_{pk}(M_{i_1,i_2})=(+_h)_{v=1}^m\left[\left(E_{pk}\left(\theta_{i_1j_v}^{h_1}\right)+_h\left(E_{pk}\left(\theta_{i_2j_v}^{h_2}\right)\times_h(-1)\right)\right)\times_hr_v\right]$$

When the decrypted value of $E_{pk}(M_{i_1,i_2})$ is 0, then two records correspond to the same patient with high probability. The main reason behind this observation is the fact that if all the attributes match then for each v, $(\theta_{i_1j_v}^{h_1}-\theta_{i_2j_v}^{h_2})=0$, and then M_{i_1,i_2} is 0. As proven below, if any of the attributes fails to match then it is highly unlikely that M_{i_1,i_2} is 0.

Algorithm 1. DS-Equijoin

Require Encrypted datasets $E_{pk}(\theta^{h_1})$ and $E_{pk}(\theta^{h_2})$; $j_1,\ldots,j_m$ are join attributes

for all $E_{pk}(\theta_{i_1}^{h_1})\in E_{pk}(\theta^{h_1})$ **do**
  for all $E_{pk}(\theta_{i_2}^{h_2})\in E_{pk}(\theta^{h_2})$ **do**
    for $v=1$ to m **do**
      $E_v\leftarrow\left(E_{pk}\left(\theta_{i_1j_v}^{h_1}\right)+_h\left(E_{pk}\left(\theta_{i_2j_v}^{h_2}\right)\times_h(-1)\right)\right)\times_hr_v$
    end for
    $E_{pk}(M_{i_1,i_2})\leftarrow E_1+_hE_2+_h\cdots+_hE_m$
  end for
end for
Send all permuted $E_{pk}(M_{i_1,i_2})$ values to KHS

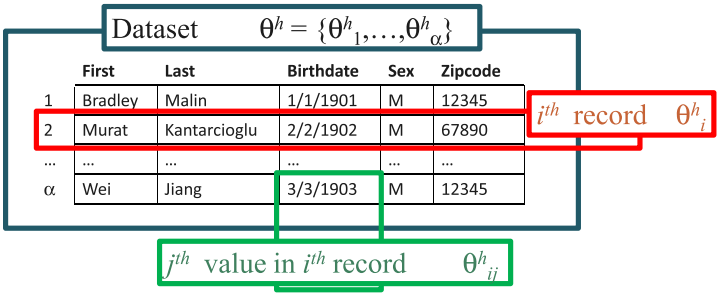


Fig. 2. Illustration of notations used in this paper on a simple example.

Algorithm 2. KHS-Equijoin

Require $E_{pk}(M_{i_1,i_2})$'s from DS
1: **for all** $E_{pk}(M_{i_1,i_2})$ **do**
2: **if** $D_{pr}(M_{i_1,i_2}) = 0$ **then**
3: (i_1, i_2) matches
4: **end if**
5: **end for**
6: Send all matching (i_1, i_2) pairs to DS

5.1. Correctness of equijoin protocol

Here, we prove that $M_{i_1,i_2} = 0$ gives the correct join result with high probability. We first derive a lemma that states the probability of computing a 0 through homomorphic addition, when there is at least one non-zero value, is very low.

Lemma 1. *Given fixed* $a_1, \ldots, a_m \in \{0, \ldots, n-1\}$ *with at least one non-zero* a_j *value and uniformly randomly chosen* $r_1, \ldots, r_m \in \{1, \ldots, n-1\}$. *Let* $S_m = \sum_{i=1}^{m} a_i \cdot r_i \bmod n$, *then* $Pr[S_m = 0] \leqslant \frac{1}{n-1}$.

Proof. Since all operations are modulo a large prime n,[3] given any $x \in \{0, \ldots, n-1\}$, we can easily state the following inequality:

$$Pr[a_j \cdot r_j = -x] = Pr[r_j = -x \cdot (a_j)^{-1}] = \begin{cases} 0, & x = 0 \\ \frac{1}{n-1}, & \text{else} \end{cases} \leqslant \frac{1}{n-1}$$

Using the above inequality, we have: $Pr[S_m = 0] = \sum_{x=0}^{n-1} (Pr[a_j \cdot r_j = -x | S_m - a_j \cdot r_j = x] \cdot Pr[S_m - a_j \cdot r_j = x])$. Thus, for any $x, Pr[a_j \cdot r_j = -x] \leqslant \frac{1}{n-1}$,

$$Pr[S_m = 0] \leqslant \frac{1}{n-1} \left(\sum_{x=0}^{n-1} Pr[S_m - a_j \cdot r_j = x] \right) \tag{1}$$

Since all operations are modulo n, $S_m - a_j \cdot r_j$ will only take values between $\{0, \ldots, n-1\}$, this implies that:

$$\left(\sum_{x=0}^{n-1} Pr[S_m - a_j \cdot r_j = x] \right) = 1 \tag{2}$$

Eqs. (1) and (2) concludes our proof. □

Lemma 1 provides intuition regarding the general properties of homomorphic addition. In the context of our protocol, this lemma can be used to prove Theorem 1. Basically, assume that we use homomorphic encryption to subtract the two patients' values in the same attribute (e.g., date of birth). Then, if the values match, the homomorphic subtraction will be an encryption of 0, or a false non-match with very low probability. Fig. 3 summarizes the findings of Theorem 1.

Theorem 1. *Given two encrypted tuples* $E_{pk}\left(\theta^{h_1}_{i_1}\right)$ *and* $E_{pk}\left(\theta^{h_2}_{i_2}\right)$, *if* $\theta^{h_1}_{i_1}$ *and* $\theta^{h_2}_{i_2}$ *matches, then* $M_{i_1,i_2} = 0$ (M_{i_1,i_2} *is defined as above); on the other hand, if* $M_{i_1,i_2} = 0$ *then* $\theta^{h_1}_{i_1}$ *and* $\theta^{h_2}_{i_2}$ *matches with probability at least* $1 - \frac{1}{n-1}$.

[3] To uphold protocol security, we recommend choosing values of n, the modular base, on the order of 1024 bits. If, for instance, we join two tables with 10 million tuples each, the expected number of mismatches is significantly smaller than one (i.e., $\frac{10^7 \cdot 10^7}{2^{1024}-1}$). Thus, for any database with the less than 2^{512} tuples, the error introduced by our scheme can be made arbitrarily small by increasing size of n.

Proof. Due to the definition of M_{i_1,i_2}, if $\theta^{h_1}_{i_1}$ and $\theta^{h_2}_{i_2}$ matches, then for all v, $\left(\theta^{h_1}_{i_1 j_v} - \theta^{h_2}_{i_2 j_v}\right) = 0$. This implies that $M_{i_1,i_2} = 0$. Let us consider the case where $M_{i_1,i_2} = 0$ but $\theta^{h_1}_{i_1}$ and $\theta^{h_2}_{i_2}$ does not match. This implies for some non-zero $a_v = \left(\theta^{h_1}_{i_1 j_v} - \theta^{h_2}_{i_2 j_v}\right)$ values, $\sum_{v=1}^{m}(a_v \cdot r_v) = 0 \bmod n$ for non-zero uniformly randomly chosen $r_v \in \{1,\ldots,n-1\}$. According to Lemma 1, the probability of such an event is less than $\frac{1}{n-1}$. This implies that if $M_{i_1,i_2} = 0$, then two tuples match with probability bigger than $1 - \frac{1}{n-1}$. □

5.2. Security of the equijoin protocol

The protocol is secure within the framework with respect to DS because it does not have access to the private keys. In addition, due to semi-honest model, we assume that DS follows the protocol and only asks KHS for the decryption for the properly constructed $E_{pk}(M_{i_1,i_2})$ values. Thus, we consider security with respect to KHS. Specifically, KHS observes only encrypted values of either 0, which corresponds to a match, or a random value, which corresponds to a non-match. Since the encryption scheme is semantically secure, KHS cannot learn anything regarding the corresponding values of the encrypted data.

5.3. Communication and computational cost

Assume *Secure-Equijoin* is performed using m attributes, and let $|\theta^{h_a}|$ indicate the number of tuples in θ^{h_a}. According to Protocol 1, for each tuple pair, we perform $2m-1$ homomorphic additions, m modulo inverses and m homomorphic multiplications. Since each homomorphic multiplication is equivalent to an exponentiation, which is much more expensive than the other operations, we define the computational complexity in terms of the number of exponentiations. Each tuple pair requires m exponentiations and there are $|\theta^{h_1}| \cdot |\theta^{h_2}|$ such pairs; as a result, the number of exponentiations for the *Secure-Equijoin* protocol is bounded by $O(|\theta^{h_1}| \cdot |\theta^{h_2}| \cdot m)$.

For each tuple pair, DS sends the M_{i_1,i_2} value to KHS. Assuming an s-bit long n value, the communication complexity is bounded by $O(|\theta^{h_1}| \cdot |\theta^{h_2}| \cdot s)$.

6. *k*-Anonymity for Secure-Equijoin

The *Secure-Equijoin* protocol is impractical with large datasets because it requires testing each new and existing record as a potential join, such that the running time increases quadratically with the number of tuples. To overcome this limitation of the protocol, we propose a method that relaxes the semantically secure protections afforded by the homomorphic cryptosystem to anonymity sets of size k. We append non-encrypted non-sensitive patient-specific values (e.g., demographics) to encrypted data (e.g., sensitive DNA information) in a manner that satisfies a formal privacy model. Specifically, hospitals disclose non-encrypted patient-specific data in a manner that satisfies k-anonymity [14,19]. Furthermore, all sensitive attributes are kept encrypted at all times. This implies that, for our purposes, k-anonymity is as strong an anonymity definition as others that require further restrictions on the "revealed" sensitive attribute distributions. Because such anonymity definitions (e.g., l-diversity [25] and t-closeness [26]) are equivalent to k-anonymity when sensitive data is not revealed.

Let QI be a set of quasi-identifier attributes that can be used with certain external information to identify a specific individual, T be a dataset represented in a relational form and $T[QI]$ be the projection of T to the set of attributes contained in QI.

Definition 1. $T[QI]$ satisfies k-anonymity if and only if each record in it appears at least k times.

The criteria for k-anonymity can be achieved via a number of mechanisms. In this paper, we concentrate on *generalization* [19]. In generalization, values are replaced by more general ones, according to a value generalization hierarchy (VGH). Fig. 4 contains VGHs for the attributes *AGE* and *ZIP CODE*. According to the VGH of *AGE*, we say that 25 can be generalized to [23,45].

As an example, consider Fig. 4a. Here, we show a dataset T with quasi-identifier $QI = \{AGE, ZIPCODE\}$. By generalization according to the VGHs, we can derive dataset in Fig. 4b ($T[Q]$), which satisfies 2-anonymity.

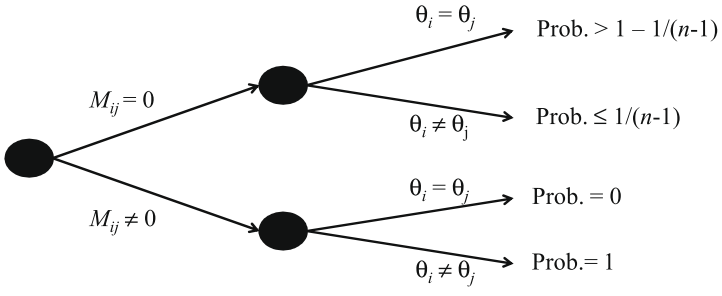


Fig. 3. Illustration of Theorem 1.

6.1. k-Anonymity as hash key

In essence, k-anonymized values serve as hash keys by which DS can partition encrypted tuples into buckets that are much smaller than the databases. In doing so, DS can perform the Secure-Equijoin procedure on the homomorphically encrypted identifiers, such as SSNs, without testing every combination of tuples in the cross-product of the submitted databases. Moreover, by k-anonymizing the data, we ensure that every tuple in a bucket is linkable to no less than k patients. Thus, after joining encrypted values, DS is unable to link any tuple to less than k patients.

To implement this model, we assume the hospitals' databases contain a common set of quasi-identifying attributes, such as a patient's residential zip code and age. Each hospital encrypts all remaining attributes via the public key of DS. The hospitals then k-anonymize the quasi-identifying values of their datasets.

Not every k-anonymization method is suitable for hashing encrypted tuples of hospital databases. Specifically, we require the generalized values output by the anonymization method to be disjoint. Otherwise, given an encrypted tuple's quasi-identifier value, the hash function will return multiple buckets. Apart from the hash operation being non-functional, there are privacy implications of the situation. With overlapping generalizations, under certain conditions, DS can infer private information about tuples through the join result. Disjoint generalizations, on the other hand do not leak private information to DS. We prove these claims next.

Claim 1. *Anonymization methods that output overlapping generalized values violate k-anonymity when used for hashing encrypted tuples.*

Proof. We prove this claim via a counter example. Suppose that some dataset anonymized based only on its *AGE* attribute (i.e. $QI = AGE$) contains overlapping generalized values [23,50) and [49,90). Some tuple r with $r \cdot AGE = 49$ can be categorized into both buckets. However, if during the join process r joins with tuples from both buckets, DS learns that $23 \leqslant r \cdot AGE < 50$ and $49 \leqslant r \cdot AGE < 90$. Obviously, $49 \leqslant r \cdot AGE < 50$ and $r \cdot AGE = 49$. The entire QI is disclosed to DS. □

Claim 2. *If the underlying anonymization method outputs disjoint generalized values DS cannot infer additional information using the join result.*

Proof. Our proof is based on the following observations:

(1) By definition, any two tuples t_1 and t_2 representing the same real-world entity should have the same QI value. Therefore, if (t_1, t_2) is in the join, then their corresponding QI values should match one another.
(2) Since generalized values used for hashing are disjoint, any encrypted tuple t is mapped to only one bucket B.
(3) Every tuple that joins with an encrypted tuple t is in its corresponding bucket B. Otherwise, a tuple from another bucket B' would join with t and since B and B' have distinct generalizations, (1) would be violated.

Based on (3), we conclude that DS cannot infer the QI values of t beyond the generalization of bucket B. This implies, DS cannot distinguish t among the tuples in B. Since $|B| \geqslant k$ by definition of k-anonymity, DS cannot use the join result to violate the anonymity of a tuple. □

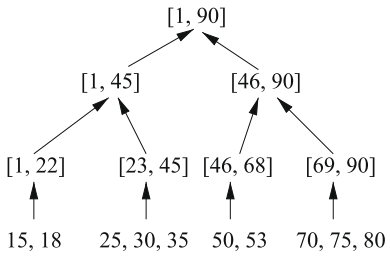
With well-defined VGHs, the most common source of overlapping generalizations is suppression. A tuple is said to be suppressed if it is generalized to the top-most level in the VGHs of all quasi-identifier attributes (i.e. [1,90] and ***** in Fig. 4c and d respectively). Therefore, algorithms that perform suppression (e.g. DataFly of [52]) are not suitable to our framework.

AGE	ZIP CODE
25	54339
75	47500
50	47535
30	54788

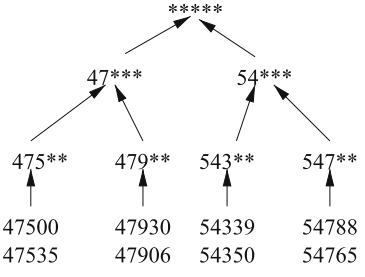
(a) Original Data

AGE	ZIP CODE
[23, 45]	54***
[46, 90]	475**
[46, 90]	475**
[23, 45]	54***

(b) 2-Anon. Data



(c) VGH of AGE



(d) VGH of ZIP CODE

Fig. 4. Data tables and value generalization hierarchies.

Even without such algorithms there still are various anonymization methods to choose from. Some of these are Mondrian multi-dimensional k-anonymity [53] and Top-down Specialization [54] methods discussed in Section 7. For the rest of the paper, we assume that the underlying anonymization method produces disjoint generalizations.

6.2. Joins with equivalent populations

First, we consider the case when all hospitals have data on the same population. In this scenario, each hospital k-anonymizes its dataset (with the same anonymization algorithm, k values and generalization schema) and submits the result to DS. When DS performs a join, it constructs buckets corresponding to each combination of k-anonymous values. For each bucket, DS executes the *Secure-Equijoin* protocol. At the completion of the protocol, every tuple from each location will be joined with data stored at DS.

Claim 3. *The joined database resulting from Secure-Equijoin at DS is k-anonymous with respect to the attributes in the quasi-identifier.*

Proof. The union of the joined quasi-identifying values is equivalent to the quasi-identifying values of any tuples involved in the join. Since there are k or more tuples in each bucket, after all tuples are joined with their corresponding partners, their quasi-identifying values do not change. Thus, the resulting database is k-anonymous. □

6.3. Joins with overlapping populations

Next, we address how to join data when hospitals collect records on overlapping populations of patients. In this case, hospitals cannot k-anonymize their databases independently, as was performed in the prior section. If this occurs, the same patient's data can be k-anonymized in different ways at different hospitals. As a consequence, data that is joined at DS could violate the k-anonymity model. For instance, consider the record {25, 47906} defined over the attributes *AGE* and *ZIP CODE* and the generalization hierarchies in Fig. 4. This tuple could be k-anonymized to {[23,45], 479**} at one hospital and {25, 47*** } at another hospital. In each submitted database, the tuples are k-anonymous, however, after joining the encrypted values, DS can infer that the corresponding demographics must be {25, 479**}, which is more specific than both of the submitted tuples. If the number of tuples with the combination of these demographic values is less than k, then the join violates k-anonymity.

Algorithm 3. k-Equijoin

Require k: anonymity threshold; $V_1, \ldots, V_m$: a set of value generalization hierarchies related to quasi-identifying attributes $A_1, \ldots, A_m$

1: DS: Send $T[Q]$ to hospital h
2: Hospital h:
  (1)Compute $C \leftarrow$ Get-Candidate$(T^h, T[Q], V_1, \ldots, V_m)$
  (2)Anonymize C based on $T[Q]$ and send C to DS
  (3)Compute $\Gamma \leftarrow (T^h - C)$
  (4)k-anonymize Γ and send it to DS
3: DS: Compute $C' \leftarrow$ Equijoin(C, T) send C' to hospital h

Algorithm 4. Get-Candidate

Require $T^h, T[Q], V_1, \ldots, V_m$
1: $C \leftarrow \emptyset$
2: $D \leftarrow T^h$
3: **for all** $t \in T[Q]$ **do**
4: $\gamma =$ GetSpec$(t[A_1, \ldots, A_m], V_1, \ldots, V_m)$
5: **for all** $\delta \in D$ **do**
6: **if** $|\delta \cap \gamma| = m$ **then**
7: $C \leftarrow C \cup \{\delta\}$
8: $D \leftarrow D - \{\delta\}$
9: **end if**
10: **end for**
11: **end for**
12: return C

To prevent this inference leak, we present a protocol that enables hospitals to coordinate and, subsequently, ensure all data stored at DS satisfies the k-anonymity framework. We call this protocol *k-Equijoin* and its key steps are presented in Protocol 3. Before delving into the details of the protocol, we provide an informal overview. Let T represent the database stored at DS. We partition T into $T[Q]$ and $T[\widehat{Q}]$. The first component, $T[Q]$, represents the projection of T on the quasi-identifier attributes. The second component, $T[\widehat{Q}]$, represents the encrypted portion of T. Similarly, data at hospital h is represented as $T^h[Q]$ and $T^h[\widehat{Q}]$. To initiate the protocol, h submits a request to DS to transfer its patient-specific records. At this point, we must consider two scenarios: (1) a base case in which h is the first hospital to submit data and (2) a general case in which h is not the first submitter. In the base case, DS has yet to receive data from any hospital, so h will k-anonymize the quasi-identifying attributes in its database and encrypt the remaining attributes. Then, h will send T^h to DS for storage. In the more general case, DS has already received and stored data from one or more hospitals. Thus, hospital h partitions his data into records that DS: (1) may have already received from other hospitals and (2) definitely has not received. Hospital h will k-anonymize the first set of records in the same schema as they were submitted to DS by other hospitals. The second set of records, which we call Γ, can be k-anonymized by h without regard to records at DS because they are the first to be submitted. Thus, h generates and sends the generalized $\Gamma[Q]$ to DS.

Now, we present the protocol more formally. To produce consistent data, the degree of k-anonymization and the generalization algorithms are fixed for the execution of the protocol. In addition, we assume there exists a fixed set of value generalization hierarchies available to the hospitals. Without loss of generality, we assume that some data are already stored at DS. Key steps of *k-Equijoin* are highlighted in Algorithm 3. In Step 1 of *k-Equijoin*, DS sends the k-anonymous portion of the centralized database to hospital h.[4] Next, in Step 2, hospital h utilizes a function Get-Candidate to compute the set C of its tuples that could potentially join to tuples in the centralized database at DS. Then, hospital h generalizes C according to $T[Q]$[5] and k-anonymizes the remaining tuples Γ. Both C and Γ, along with the corresponding encrypted portions are sent to DS. In Step 3, after DS receives C, DS will locate the tuples in C that can potentially join to its data using the *Secure-Equijoin* protocol. After this computation, DS notifies hospital h which tuples can definitely not be joined with existing records, denoted as the set C'. Finally, in Step 4, h k-anonymizes the remaining tuples with those in C', and sends the k-anonymized tuples to DS.[6]

The correctness of the *k-Equijoin* protocol relies on the Get-Candidate function (Algorithm 4) to produce a set of data tuples in T^h whose projection on quasi-identifier attributes can be anonymized to some tuples in $T[Q]$. The function works as follows: At the step 4 of Algorithm 4, GetSpec$(t[A_1,\ldots,A_m],V_1,\ldots,V_m)$ (GetSpec(t) for short) denotes a function which returns a set γ of specific values (values at the bottom of VGHs: $V_1,\ldots,V_m$) related to quasi-identifying attributes $A_1,\ldots,A_m$, and the corresponding parent values of these specific values are contained in t. For example, suppose $m=2$ and let V_1 and V_2 be VGHs presented in Fig. 4, corresponding to the attributes *AGE* and *ZIP CODE*. Using set representation, let $t=\{[46,90],475\ *\ *\}$ be one of the records in $T[Q]$. Based on the two VGHs and t, $\gamma=\{50,53,70,75,80,47500,47535\}$. Suppose $\delta=\{50,54339\}$ is one of the records in $T^h[Q]$. Since $|\delta\cap\gamma|=1\neq 2$, δ cannot be generalized to t. On the other hand, if $\delta=\{50,47500\}$, then δ can be generalized to t because $|\delta\cap\gamma|=2$. Such a δ is called a candidate for the future join process at DS. We next prove the condition $|\delta\cap\gamma|=m$, in step 6 of Algorithm 4, indeed determines whether δ can be generalized to t.

Claim 4. *Given δ and $\gamma=$ GetSpec$(t[A_1,\ldots,A_m],V_1,\ldots,V_m)$, if $|\delta\cap\gamma|=m$, δ can be generalized to t according to the quasi-identifying attributes $A_1,\ldots,A_m$ and VGHs $V_1,\ldots,V_m$.*

Proof. We prove this claim via a contrapositive argument. Assume $|\delta\cap\gamma|=m$ and $\delta[A_1,\ldots,A_m]$ cannot be generalized to t, then $\exists\delta[A_i]\in\delta$ such that $\delta[A_i]$ cannot be generalized to any value in $\{t[A_1],\ldots,t[A_m]\}$ assuming $V_1,\ldots,V_m$ are disjoint. On the other hand, $|\delta\cap\gamma|=m$ implies that $\delta\subseteq\gamma$ and consequently, $\delta[A_i]$ must match some value in γ. Based on the definition of *GetSpec*(t), we know that every value in γ can be generalized to some value in $\{t[A_1],\ldots,t[A_m]\}$. Therefore, it must be the case that $\delta[A_i]$ can be generalized to some value in $\{t[A_1],\ldots,t[A_m]\}$. This contradicts the assumption. In addition, since we assume $V_1,\ldots,V_u$ are disjoint, for any two $\delta[A_i],\delta[A_j]\in\delta$, they cannot be generalized to the same value in $\{t[A_1],\ldots,t[A_m]\}$. This guarantees that for $1\leqslant i\leqslant m$, $\delta[A_i]$ can be generalized to $t[A_i]$ as long as $|\delta\cap\gamma|=m$ holds. □

From a security perspective, the *k-Equijoin* protocol needs to guarantee its output data are still k-anonymous. Next we analyze this security issue of *k-Equijoin* based on the following claim.

Claim 5. *The database at DS after all hospitals execute k-Equijoin is k-anonymous with respect to the attributes in the quasi-identifier and its generalized values are disjoint.*

Proof. When DS holds no tuples at all, the first hospital can k-anonymize its dataset and submit directly. Using this as the basis case, we prove our claim inductively. Suppose that before hospital h engages into the *k-Equijoin* protocol, the database at DS is k-anonymous and it consists of disjoint generalizations. After the execution of *k-Equijoin*, tuples of h are either joined

[4] Before sending $T[Q]$, DS can eliminate all duplicates in $T[Q]$ to reduce communication costs by a factor of k.

[5] After generalization, tuples of C might not be k-anonymous by locally. But since $T[Q]$ satisfies k-anonymity, every tuple is guaranteed to be mapped into an equivalence class of size k or more. Therefore the view of DS is k-anonymous.

[6] Note that a small number of tuples may not be k-anonymized. When this occurs, h will not send these tuples to DS. However, these data can be combined with future collected data. If the size of combined data is greater than k, h can initiate k-Equijoin protocol again with DS.

with existing tuples, or anonymized according to $T[Q]$, or anonymized independently. We show that in all three cases, the inductive hypothesis remains true.

- Tuples that are joined successfully: Join operation over tuples $t \in T^h$ and $t' \in T$ simply appends t' with the encrypted attribute values of t. $T[Q]$ does not change and t', which by the inductive hypothesis was k-anonymous before the join, remains k-anonymous.
- Tuples anonymized according to $T[Q]$: When added to $T[Q]$, these tuples only increase the sizes of buckets. Therefore disjointness of the generalizations and k-anonymity will not be affected.
- Tuples anonymized independently: According to Algorithm 3, these are the non-candidate tuples denoted by Γ. Since QI attribute values of non-candidate tuples are distinct from every tuple in $T[Q]$, any generalized value of anonymized Γ will not overlap with existing generalized values of $T[Q]$.[7] These disjoint, k-anonymous generalizations, when added to those of $T[Q]$ do not violate k-anonymity of $T[Q]$.

It follows from the inductive hypothesis that after all hospitals execute *k-Equijoin*, the dataset at DS is still k-anonymous. □

We hypothesize that *k-Equijoin* will reduce the number of cryptographic match evaluations that DS must perform in comparison to the *Secure-Equijoin* protocol because tuples corresponding to the same patient must reside in the same bucket. We experimentally investigate this hypothesis below.

7. Experiments

To evaluate the proposed protocols, we selected the US Census income dataset, which is publicly-available on the UC Irvine Machine Learning Repository [55]. This dataset contains person-specific records that were extracted from the 1994 and 1995 Current Population Surveys. It contains 142,521 tuples without missing values. There are 40 demographic and employment-related attributes.

7.1. Secure-Equijoin

We prototyped our protocols in Java and executed the secure join query experiments using relations of 100, 200, 300, and 400 tuples extracted from the Census income dataset. Please note that for relations of size 100, we need to compare 10,000 tuple pairs. Similarly, for data set size 400, we need to compare 160,000 tuple pairs. For simplicity, we executed join queries of the form $\theta^{h_1} \bowtie \theta^{h_1}$ with different numbers of attributes in the equijoin criteria. The experimental results are summarized in Fig. 5 and indicate that join queries are computationally expensive and thus very time consuming. As expected, the running time of the join protocol increases linearly with the number of attributes in the join and quadratically with the size of the relation. For instance, it took around an hour to compute an integration of two datasets with 100 patients each (i.e., a join operation that involves 10,000 tuple pair comparisons) across four attributes.

From a practical perspective, we investigated the degree to which specialized software implementations could decrease the time necessary to complete secure equijoins. We also implemented the *Secure-Equijoin* protocol in C programming language with the GMP library (http://gmplib.org/). Our results indicate that we can achieve a five times speed-up. This implies that we can complete two million tuple pair comparisons in two days, which may be an acceptable amount of time for some biomedical research queries. Yet, as the size of the database grows, the savings afforded by specialized code is significantly outpaced by the increased time required to evaluate possible tuple pairs in the homomorphic space. Thus, scaling the basic join protocol to large datasets is not feasible for the large databases that will be employed in biomedical data mining endeavors.

7.2. k-Equijoin

To evaluate the effect of k-anonymous demographics on secure joins, we compared the number of homomorphic exponentiation operations required by *k-Equijoin* and *Secure-Equijoin* in similar settings. As discussed earlier, homomorphic exponentiation is significantly more costly in comparison to homomorphic addition and other non-cryptographic operations. Therefore, the number of exponentiations yields a good estimate of the overall costs of these two methods.

Since *k-Equijoin* never performs more exponentiations than *Secure-Equijoin*, we present our measurements as a percentage of the savings in cryptographic operations executed. For example, given two hospitals' datasets of 1000 records each, *Secure-Equijoin* would perform $10^6 \times m$ exponentiations, where m is the number of join attributes. In this scenario, a savings of 99% indicates *k-Equijoin* performs only $(1 - (99/100)) \times 10^6 \times m = 10^4 \times m$ exponentiations. Translated into execution time, one would expect *k-Equijoin* to run around 100 times faster than *Secure-Equijoin*.

The advantage of *k-Equijoin* over *Secure-Equijoin* is determined by the characteristics of the input datasets and the anonymization parameters. To assess the effect of anonymization, we performed experiments on the anonymity requirement k

[7] If not, generalizations of anonymized Γ can be refined without violating k-anonymity.

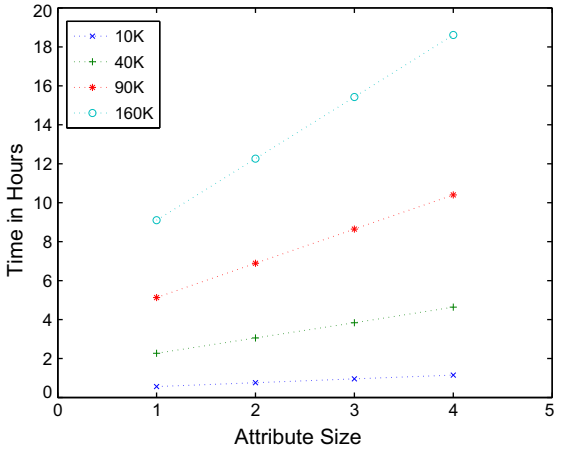


Fig. 5. Execution Time for Join Queries.

and the quasi-identifier q (i.e., the generalizable attributes). Out of 40 attributes in the dataset, we selected 4 to represent the quasi-identifier. These attributes and their generalization hierarchies are summarized in Table 2. In the experiments with varying quasi-identifier size, the first $|q|$ elements of the table constitute the quasi-identifier. For example, when $|q| = 2$, $q = \{Age, MaritalStatus\}$.

We adjusted dataset characteristics by varying hospital dataset sizes, $|h|$, and the number of such datasets $\#h$. Every input dataset is built by randomly partitioning the preprocessed Census dataset.

Finally, we investigated the effect of the join size. When the set of join attributes is pre-determined, it is impossible to adjust the selectivity of the join operation, so we opt not to use a set of join attributes. Instead, we use a parameter s that determines the ratio of the join size (i.e., $T^h \bowtie T$) to the size of the input dataset, T^h. To imitate a join of selectivity s, we randomly select $s \times |T^h|$ tuples of T^h and consider these joined. For example, $s = 0.05$ implies that roughly %5 of the input dataset T^h will be in $T^h \bowtie T$. Default values of all these parameters are provided in Table 1.

7.2.1. k-Anonymity algorithms

In our experiments, we compare three anonymization methods: (1) an algorithm of our own design, which we call MaxEntropy (with reference to its embedded heuristics), (2) the Mondrian multi-dimensional algorithm [53], and (3) the Top-Down Specialization (TDS) [54] algorithm. All three algorithms follow the "top-down" approach: they start with only one group of tuples, generalized to the highest level on all QI attributes. Then, according to some heuristics, at each step a "specialization" operation that makes some tuples' values more detailed is performed. MaxEntropy heuristically selects the attribute with the maximum entropy for specialization, as implied by the name. The premise of the Mondrian algorithm is such that at each iteration, it specializes the attribute with the maximum number of well-represented values in its domain [53]. TDS relies on information theoretic measures to maximize the classification accuracy of a decision tree classifier built on the output. An important property of TDS is that specializations that are not beneficial (i.e., does not increase classification accuracy) are not performed. Usually this implies coarser generalizations that contain more tuples in comparison to MaxEntropy and Mondrian.

Another major difference between these algorithms is related to how specializations are defined. Specializations of TDS are performed not over a group of tuples, but the domains of QI attributes. Such methods are said to apply global recoding.[8] Mondrian and MaxEntropy, on the other hand, perform local recoding that allows a broader search space of specializations. Consequently, these methods are more flexible and their output is of higher-granularity in comparison.

Mondrian and TDS builds generalization hierarchies of QI attributes on-the-fly when the corresponding attribute's domain is completely ordered.[9] Therefore, for such attributes, Mondrian and TDS can adapt themselves to the underlying distribution. In this respect, MaxEntropy is limited to user-defined generalization hierarchies.

Finally we would like emphasize that these three anonymization methods produce disjoint generalizations. Most top-down methods, including MaxEntropy, Mondrian and TDS, operate on well-defined non-overlapping generalization hierarchies and do not yield to suppression. Additionally, the generalizations usually cover the entire domain of QI attributes (i.e., $\Gamma = \emptyset$ in Algorithm 3). The only exception of the latter is highly skewed datasets: when a specialization partitions the data and some partitions contain at least k tuples while others are completely empty. Since in our experiments the datasets were partitioned randomly across hospitals, this has never occurred and Γ was always empty.

[8] See [53] for a formal definition of local versus global recoding.

[9] The categorical attributes of our quasi-identifier do not satisfy this requirement. As such, we defined user-defined generalization hierarchies.

Table 1
Default values.

Parameter	Value
Anonymity requirement, k	64
Quasi-identifier size, $\|q\|$	4
Dataset size, $\|h\|$	71,261
Number of datasets, $\#h$	2
Selectivity ratio, s	0.15

Table 2
Census dataset description.

Attribute	Values	VGH height
Age	91	5
Marital	7	4
Race	5	2
Sex	2	2

7.2.2. Anonymity requirement (k)

The first observation is that as the anonymity requirement increases, *k-Equijoin* reaps less savings over *Secure-Equijoin*. This finding is supported by Fig. 6 and is intuitive. In contrast to *k-Equijoin*, the cost of *Secure-Equijoin* is independent of k. Therefore, as k increases, so too does the amount of generalization in the anonymized dataset and the number of records per bucket that *k-Equijoin* must compare.

Among the three anonymization algorithms, Mondrian is the least affected by increasing k. TDS suffers from the limited search space of global recoding and performs the worst. MaxEntropy and Mondrian, which apply local recoding, yield much better results. Mondrian performs better than MaxEntropy because Mondrian can dynamically adjust the generalization hierarchy of the numeric *Age* attribute. MaxEntropy, on the other hand, has to comply with the user-defined hierarchies, which do not provide the best granularity of generalized values for various k (i.e., the hierarchy produced by Mondrian can be deeper than the user-defined hierarchy).

7.2.3. Quasi-identifier size (|q|)

Fig. 7 reports the savings as a function of increasing quasi-identifier size. According to our results, as the quasi-identifier set grows, the anonymized datasets of MaxEntropy and Mondrian consist of more buckets. Since the dataset size remains constant, the average number of records in each bucket reduces significantly. As a result, the *Secure-Equijoin* performed between candidate records and bucket elements requires less cryptographic operations, such that *k-Equijoin* attains greater savings in cryptographic operations.

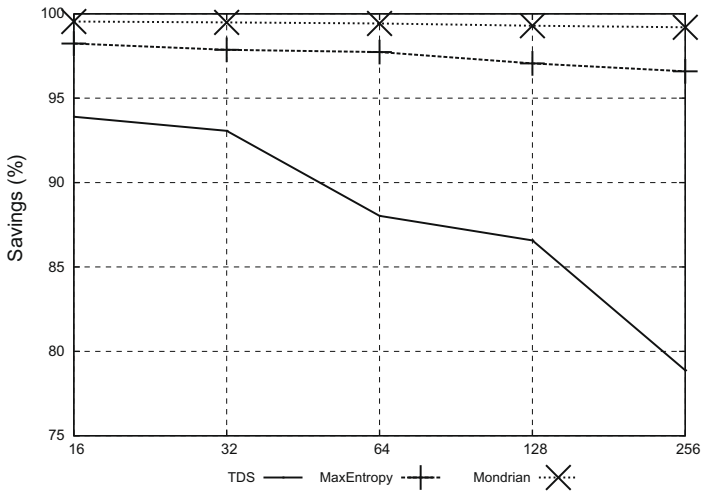


Fig. 6. Anonymity Requirement, k.

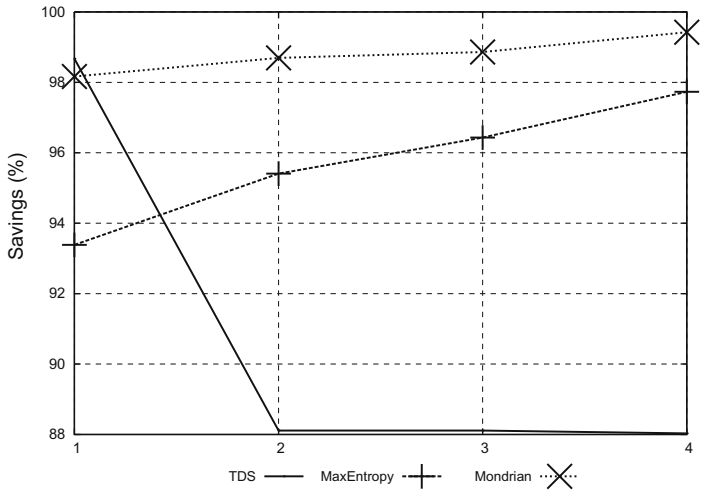


Fig. 7. Quasi-Identifier Size, $|q|$.

The situation is completely different with TDS. When $|q| = 1, q = Age$ and TDS generates the generalization hierarchy for continuous attributes like *Age* on-the-fly. Also, since there is only one QI attribute, the disadvantage of performing global recoding is minimal. So the setting is very much like MaxEntropy and Mondrian. That is why, at $|q| = 1$, TDS performs even better than Mondrian. However, as the QI grows to include a discrete attribute the savings decline quite sharply.

7.2.4. Dataset size ($|h|$)

In this experimental scenario, we incremented the sizes of hospital datasets that are integrated. For each experiment, we partitioned the Census dataset into a certain number of hospitals (indicated by the denominator of x-axis values) and integrated the first two datasets. In the results, shown in Fig. 8, $n = 142,521$ denotes the number of records in the Census dataset after preprocessing. Thus, $|h| = n/10$ implies $|h| = 142,521/10 = 14,252$.

As expected, increasing the dataset sizes, without changing the anonymity requirement k, increases the savings in cryptographic operations. This is simply because the generalization values of the buckets are of higher-granularity. In Step 3 of Algorithm 3, the number of records of h whose quasi-identifier values match a given bucket decreases. Consequently, *Secure-Equijoin* can be completed with less cryptographic operations. Notice that the savings of Mondrian is the least affected, since this algorithm yields higher-granularity buckets, even for considerably small datasets.

7.2.5. Number of datasets (#h)

Fig. 9 depicts the results of varying the number of input datasets that are integrated. In this experiment, each dataset consists of 14,252 records. Performance of the methods do not vary much with increasing number of datasets. As we discussed

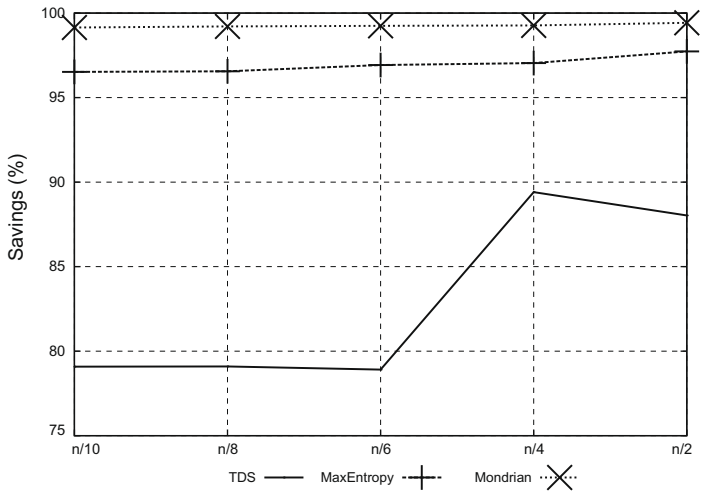


Fig. 8. Dataset Size, $|h|$.

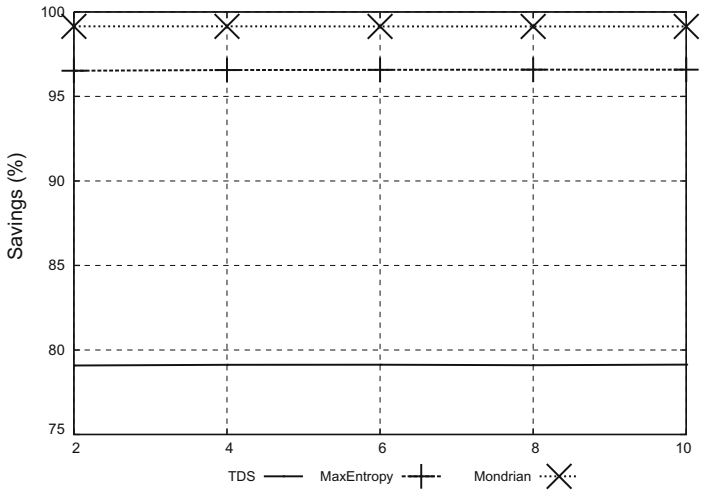


Fig. 9. Number of Datasets, $\#h$.

before, by the nature of top-down anonymization methods generalization values cover the entire domain of QI attributes. Consequently, after the first dataset is submitted to DS, in Algorithm 3, $\Gamma = \emptyset$ for every other hospital. This implies that all tuples of T^h are candidates. Due to random partitioning of tuples to hospitals, candidates of different hospital datasets match buckets of $T[Q]$ in almost the same way and bucket sizes increase linearly as new hospital datasets are added. Therefore $\#h$ has no significant effect on the savings.

7.2.6. Join selectivity (s)

Our experiments suggest that join selectivity does not affect the savings from cryptographic operations unless $\#h > 2$. This is why, in this experimental scenario, we performed the experiments with $\#h = 10$ and $|h| = 14,252$. The results are depicted in Fig. 10.

As the join selectivity decreases (or equivalently, as s increases), more records from different input datasets match each other and the integrated dataset grows at a much slower rate. With less number of records in the integrated dataset, the cost of integrating new datasets lessens. These observations suggest that increasing s decreases the costs associated with both *Secure-Equijoin* and *k-Equijoin*. However, according to Fig. 10, relative to *Secure-Equijoin*, the rate of this decrease is almost the same for *k-Equijoin*.

Given T^h and s, we build the join set by randomly selecting $\%s$ of the tuples of T^h. Notice that due to randomness, distribution of the joined tuples will be similar to the distribution of T^h. Therefore, as s increases, every bucket is expected to shrink proportionately to C' of Algorithm 3. On average, the costs decrease proportionately as well and the savings do not change with varying s.

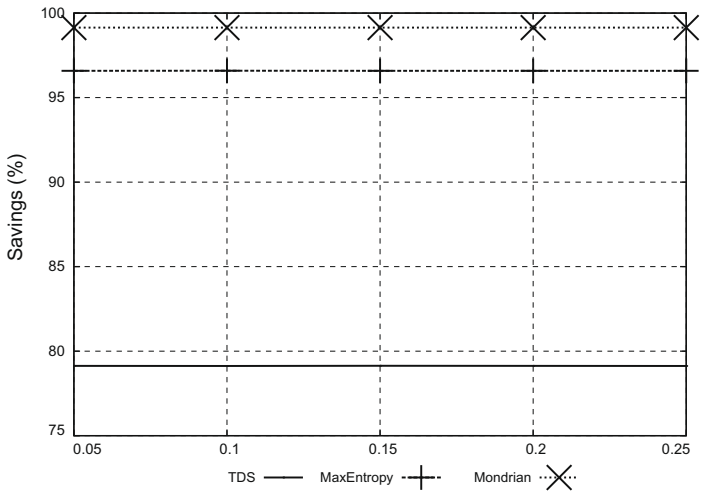


Fig. 10. Join Selectivity, s.

8. Discussion

The limiting factor in the applicability of our join protocols is the computational power needed for exponentiations and the bandwidth necessary for communication between the data site (DS) and key holder site (KHS). We believe that our protocols will be more efficient when implemented in secure computer hardware. Here, we suggest several potential hardware-based improvements.

First, significant efficiency gains for our protocols can be achieved through cryptography accelerators that are tailored to execute expensive exponentiation operations. Based on reported results with hardware accelerators, the combination of more efficient software implementations (e.g., in the GMP library of the C language) with hardware accelerators could substantially decrease the time needed to complete an exponentiation in comparison to our Java-based experiments. This implies that secure joins of relations, without the use of *k*-anonymous keys, on databases of 10,000 could be achieved in less than a day. We leave the implementation of our algorithms using crypto accelerators as a future work.

Second, we can decrease the communication cost by co-locating KHS and DS. Specifically, we envision a system in which the functions of the KHS are performed by a secure co-processor that resides on the same server as DS. A secure co-processor is a single-board computer consisting of a CPU, memory and special-purpose cryptographic hardware contained in a tamper-resistant shell; certified to level 4 under FIPS PUB 140-1 (One example of such a secure co-processor is the IBM 4758 Cryptographic co-processor [56]). When installed on the server, it is capable of performing local computations that are completely hidden from the server. If tampering is detected, the secure co-processor clears the internal memory. The implementation of KS functionality through a secure co-processor on the same machine as DS will decrease the communication cost.

9. Conclusions

In this paper, we presented a framework where the administrator of the repository can perform joins of encrypted databases without decrypting or inferring the contents of the joined records. Furthermore, we presented an efficient extension to the join protocol that reveals patient-specific demographics in a manner that satisfies a formal privacy model, i.e., *k*-anonymity. In doing so, we allow the administrator to perform efficient joins with the guarantee that each record can be linked to no less than *k* patients in the population. This research is notable in that it demonstrates how centralized biomedical data repositories can be integrated and updated with data distributed healthcare organizations without violating privacy regulations. Our extensive experimental results indicate that by combining formal anonymity requirements with cryptographic techniques, we can achieve significant efficiency gains for privacy-preserving data integration in the context of centralized biomedical data repositories. As a future work, we intend to implement our algorithms in real-world settings and conduct experiments involving real-world medical data sets.

References

[1] National Institutes of Health, Final NIH statement on sharing research data, NOT-OD-03-032.
[2] National Institutes of Health, Policy for sharing of data obtained in NIH supported or conducted genome-wide association studies, NOT-OD-07-88.
[3] Z. Lin, M. Hewitt, R. Altman, Using binning to maintain confidentiality of medical data, in: Proceedings of the American Medical Informatics Association Annual Symposium, San Antonio, TX, 2002, pp. 454–458.
[4] B. Malin, L. Sweeney, Determining the identifiability of DNA database entries, in: Proceedings of the American Medical Informatics Association Annual Symposium, Los Angeles, CA, 2000, pp. 537–541.
[5] B. Malin, L. Sweeney, How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems, Journal of Biomedical Informatics 37 (2004) 179–192.
[6] B. Malin, An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future, J. Am. Medical Informatics Assoc. 12 (1) (2005).
[7] B. Malin, Re-identification of familial database records, in: Proceedings of the American Medical Informatics Association Annual Symposium, Washington, DC, 2006, pp. 524–528.
[8] M. Kantarcioglu, W. Jiang, Y. Liu, B. Malin, A cryptographic approach to securely share and query genomic sequences, IEEE Transactions on Information Technology in Biomedicine, 12 (5) (2008) 606–617.
[9] K. Helliker, A new medical worry: identity thieves find ways to target hospital patients, Wall Street Journal (2005).
[10] C. Quantin, F. Allaert, P. Avillach, M. Fassa, B. Riandey, G. Trouessin, O. Cohen, Building application-related patient identifiers: what solution for a european country?, International Journal of Telemedicine and Applications (2008) 678302.
[11] S. Grannis, J. Overhage, C. McDonald, Analysis of identifier performance using a deterministic linkage algorithm, in: Proceedings of the 2002 American Medical Informatics Annual Fall Symposium, 2002, pp. 305–309.
[12] J. Berman, Zero-check: a zero-knowledge protocol for reconciling patient identities across institutions, Archives of Pathology and Laboratory Medicine 128 (2004) 344–346.
[13] M. Kantarcioglu, W. Jiang, B. Malin, A privacy-preserving framework for integrating person-specific databases, in: J. Domingo-Ferrer, Y. Saygin (Eds.), Proceedings of the 2008 Conference on Privacy in Statistical Databases (PSD), vol. 5262, Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 298–314.
[14] L. Sweeney, *k*-Anonymity: a model for protecting privacy, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 10 (5) (2002) 557–570.
[15] P. Samarati, Protecting respondents' identities in microdata release, IEEE Transactions on Knowledge and Data Engineering 13 (6) (2001) 1010–1027. doi: http://doi.ieeecomputersociety.org/10.1109/69.971193.
[16] Department of Health and Human Services, Standards for privacy of individually identifiable health information, Final Rule, Federal Register (2002) 160–164.
[17] B. Malin, Protecting genomic sequence anonymity with generalization lattices, Methods of Information in Medicine 44 (5) (2005) 687–692.

[18] K. El Emam, F. Dankar, Protecting privacy using *k*-anonymity, Journal of the American Medical Informatics Association 15 (2008) 627–637.
[19] L. Sweeney, Achieving *k*-anonymity privacy protection using generalization and suppression, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 10 (5) (2002) 571–588.
[20] Y. Chiang, T. Hsu, C. Liau, D. Wang, Preserving confidentiality when sharing medical database with the cellsecu system, International Journal of Medical Informatics 71 (2003) 17–23.
[21] J. Gardner, L. Xiong, Hide: an integrated system for health information de-identification, in: Proceedings of the 21st IEEE International Symposium on Computer-Based Medical Systems, Jyvskyl, Finland, 2008, pp. 254–259.
[22] C. Cassa, S. Grannis, J. Overhage, K. Mandl, A context-sensitive approach to anonymizing spatial surveillance data: impact on outbreak detection, Journal of the American Medical Informatics Association 13 (2006) 160–165.
[23] R. Agrawal, C. Johnson, Securing electronic health records without impeding the flow of information, International Journal of Medical Informatics 76 (2007) 471–479.
[24] M.E. Nergiz, M. Atzori, C. Clifton, Hiding the presence of individuals from shared databases, in: SIGMOD'07: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ACM, New York, NY, USA, 2007, pp. 665–676. doi:http://doi.acm.org/10.1145/1247480.1247554.
[25] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam, *l*-Diversity: Privacy beyond *k*-anonymity, ICDE 2006 (2006) 24.
[26] N. Li, T. Li, S. Venkatasubramanian, t-Cssloseness: Privacy beyond *k*-anonymity and *l*-diversity, in: Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on, 15–20 April 2007, pp. 106–115.
[27] S. Benkner, G. Berti, G. Engelbrecht, J. Fingberg, G. Kohring, S. Middleton, R. Schmidt, Gemss: grid-infrastructure for medical service provision, Methods of Information in Medicine 44 (2005) 177–181.
[28] Anonymous, Medicine's new central bankers, The Economist.
[29] V. Barbour, UK Biobank: a project in search of a protocol?, Lancet 361 (2003) 1734–1738.
[30] C. Clifton, M. Kantarcioglu, A. Foan, G. Schadow, J. Vaidya, A. Elmagarmid, Privacy-preserving data integration and sharing, in: Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2004.
[31] S. Bhowmick, L. Gruenwald, M. Iwaihara, S. Chatvichienchai, Private-iye: a framework for privacy preserving data integration, in: Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06), IEEE Computer Society, 2006.
[32] M. Scannapieco, I. Figotin, E. Bertino, A. Elmagarmid, Privacy preserving schema and data matching, in: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, 2007.
[33] R. Agrawal, A. Evfimievski, R.Srikant, Information sharing across private databases, in: Proceedings of ACM SIGMOD 2003, San Diego, California, 2003.
[34] L. Kissner, D. Song, Privacy preserving set operations, in: Proceedings of the 25th Annual International Cryptology Conference – CRYPTO'05, 2005, pp. 241–257.
[35] R. Agrawal, D. Asonov, M. Kantarcioglu, Y. Li, Sovereign joins, in: ICDE'06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), IEEE Computer Society, Washington, DC, USA, 2006. doi:http://dx.doi.org/10.1109/ICDE.2006.144.
[36] M.J. Freedman, K. Nissim, B. Pinkas, Efficient private matching and set intersection, in: Eurocrypt 2004, International Association for Cryptologic Research (IACR), Interlaken, Switzerland, 2004.
[37] F. Emekci, D. Agrawal, A. El Abbadi, A. Gulbeden, Privacy preserving query processing using third parties, in: Proceedings of ICDE 2006, Atlanta, GA, 2006.
[38] R. Pon, T. Critchlow, Performance-oriented privacy-preserving data integration, in: Data Integration in the Life Sciences, Springer, 2005, pp. 240–256.
[39] A. Inan, M. Kantarcioglu, M. Scannapieco, E. Bertino, A hybrid approach to private record linkage, in: Proceedings of the 24th Int'l Conference on Data Engineering – ICDE'08, 2008.
[40] C. Quantin, H. Bouzelat, F. Allaert, A. Benhamiche, J. Faivre, L. Dusserre, How to ensure data security of an epidemiological follow-up: quality assessment of an anonymous record linkage procedure, International Journal of Medical Informatics 49 (1) (1998).
[41] T. Churces, P. Christen, Some methods for blindfolded record linkage, BMC Medical Informatics and Decision Making 4 (9) (2004).
[42] A. Al-Lawati, D. Lee, P. McDaniel, Blocking-aware private record linkage, in: Proceedings of IQIS 2005, Baltimore, Maryland, 2005.
[43] M.G. Elfeky, A.K. Elmagarmid, V.S. Verykios, TAILOR: a record linkage tool box, in: Proceedings of the 18th International Conference on Data Engineering (ICDE-2002), 2002, pp. 17–28.
[44] C. Georgiadis, I. Mavridis, G. Pangalos, Healthcare teams over the internet: programming a certificate-based approach, International Journal of Medical Informatics 70 (2003) 161–171.
[45] B. Lampson, M. Abadi, M. Burrows, E. Wobber, Authentication in distributed systems: theory and practice, ACM Transactions on Computing Systems 10 (1992) 265–310.
[46] F. Wozak, T. Schabetsberger, E. Ammenwerth, End-to-end security in telemedical networks – a practical guideline, International Journal of Medical Informatics 76 (2007) 484–490.
[47] O. Goldreich, The Foundations of Cryptography, vol. 2, Cambridge University Press, 2004, Ch. General Cryptographic Protocols. URL <http://www.wisdom.weizmann.ac.il/oded/PSBookFrag/prot.ps>.
[48] P. Paillier, Public key cryptosystems based on composite degree residuosity classes, in: Advances in Cryptology – Proceedings Eurocrypt'99, Lecture Notes in Computer Science, vol. 1592, Springer-Verlag, 1999, pp. 223–238.
[49] O. Baudron, P. Fouque, D. Pointcheval, G. Poupard, J. Stern, Practical multi-candidate election system, in: Proceedings of the Twentieth ACM Symposium on Principles of Distributed Computing, 2001, pp. 274–283.
[50] Y. Lindell, B. Pinkas, Privacy preserving data mining, in: M. Bellare (Ed.), Advances in Cryptology – CRYPTO 2000: Proceedings of the Twentieth Annual International Cryptology Conference, vol. 1880, Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 36–54.
[51] S. Laur, H. Lipmaa, T. Mielikinen, Private itemset support counting, in: S. Qing, W. Mao, J. Lopez, G. Wang (Eds.), Proceedings of the Seventh International Conference Information and Communications Security (ICICS), vol. 3783, Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 97–111.
[52] L. Sweeney, Guaranteeing anonymity when sharing medical data, the datafly system, in: Proceedings of the 1997 American Medical Informatics Association Annual Fall Symposium, 1997, pp. 51–55.
[53] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Mondrian multidimensional *k*-anonymity, in: ICDE'06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), IEEE Computer Society, Washington, DC, USA, 2006, p. 25. doi:http://dx.doi.org/10.1109/ICDE.2006.101.
[54] B.C.M. Fung, K. Wang, P.S. Yu, Top-down specialization for information and privacy preservation, in: ICDE'05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05), IEEE Computer Society, Washington, DC, USA, 2005, pp. 205–216.
[55] C. Blake, C. Merz, UCI repository of machine learning databases (1998). URL <http://www.ics.uci.edu/mlearn/MLRepository.html>.
[56] IBM, IBM PCI cryptographic coprocessor, 2004. URL <http://www.ibm.com/security/cryptocards/html/pcicc.shtml>.

Murat Kantarcioglu is currently an assistant professor of computer science at University of Texas at Dallas. He had a Ph.D. degree from Purdue University in 2005. He received his Master's in Computer Science from Purdue University in 2002 and his Bachelor's degree in computer engineering from METU, Ankara, Turkey in 2000. He is also a recipient of NSF CAREER Award. His research interests lie at the intersection of privacy, security, data mining and databases: security and privacy issues raised by data mining; distributed data mining techniques; security issues in databases; and use of data mining for intrusion and fraud detection. His current research is funded by grants from NSF, AFOSR, ONR, and IARPA.

Ali Inan is currently a Ph.D. student in Computer Science department at The University of Texas at Dallas. He received his Bachelor's and Master's degrees in computer science from Sabanci University, Istanbul, Turkey, in 2004 and 2006 respectively. His research interests are in database systems, data mining and security and privacy issues related to management of data with emphasis on privacy preserving data mining and data integration.

Wei Jiang is an assistant professor at the Department of Computer Science of Missouri University of Science and Technology. He received the Bachelor's degrees in both Computer Science and Mathematics from the University of Iowa, Iowa City, Iowa, in 2002. He received the Master's degree in Computer Science and the Ph.D. degree from Purdue University, West Lafayette, IN, in 2004 and 2008 respectively. His research interests include privacy-preserving data mining, data integration, privacy issues in federated search environments, and text sanitization.

Bradley Malin is an Assistant Professor of Biomedical Informatics and Computer Science at Vanderbilt University and currently directs the Health Information Privacy Laboratory. He received a Master's (2002) and Ph.D. (2006) from the School of Computer Science at Carnegie Mellon University. He also received a Bachelor's in molecular biology (2000) and a Master's degree in public policy and management (2003) from Carnegie Mellon University. His research interests are in the design and analysis of privacy models for personal data that is collected, stored, and shared in large complex socio-technical systems. He is particularly interested in the privacy-preserving management of biological and clinical data.