Research Article

International Journal of Advanced Computer Research, Vol 8(38)
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2018.838017

The modeling of privacy preserving and statistically analysable database (PPSADB) system

Hyun-A Park*
Assistant Professor, Medical Health Sciences, KyungDong University, 815 Gyeonhwon-ro, MunMak-eub, WonJu-city, Kangwon-do, South Korea

Received: 26-July-2018; Revised: 11-September-2018; Accepted: 14-September-2018

Abstract
As the general data protection regulation (GDPR) of the European Union (EU) became enforceable from May 25, 2018, privacy gets to hot issues again. Especially, health information includes sensitive personal information, while it is encouraged to contribute to medical research data. The modelling for privacy preserving and statistically analysable database (PPSADB) system was proposed as a solution for this bilateral feature of health information. The proposed system consists of largely two kinds of database; encrypted database (EnDB) for usual time and statistically analysable database (SADB) for publishing. The health information (HI) in SADB is permuted by pseudorandom permutation, instead of encryption. In result, our system can satisfy privacy requirements and simultaneously provide almost all kinds of SQL queries and arithmetic operations for medical research. Additionally, it solves the problem of previous researches such as inter-column operations and dynamic database on encrypt (cryptographic or privacy technical) database.

Keywords
Health information, Statistical analysis for medical research, Privacy, Security, Database.

1.Introduction
Nowadays health information (HI) is digitized and stored in health record system such as electronic health record (EHR), electronic medical record (EMR) or personal health record (PHR) systems. Some people manage their health through remote medical system. Some people want to share their symptoms or experiences (success or fail) with other people and researchers find better treatments through on-line website service systems ("PatientsLikeMe"[1] or "Curetogether"[2]). Recently, privacy has gotten into hot issues again as one of the most important and necessary problems because the general data protection regulation (GDPR) began to replace the previous data protection directive of the European Union (EU) from May 25, 2018.

The important thing is that HI has bilateral features. Health data include a lot of sensitive and private things, while they are encouraged to contribute to medical research. Hence, HI systems should satisfy both properties of privacy protection and medical research data as a publishable database (DB) systems.

*Author for correspondence

The most secure way to protect privacy may be DB encryption, but it brings about other problems such as inefficiency and malfunctioning. The secure cryptographic method requires lots of computational overhead and cannot support statistical analysis of medical research because arithmetic operation with encrypted data cannot be the same as an arithmetic result with raw data. The one bit of difference between plaintexts comes to be a number of bits on ciphertexts. It makes DB engineers much harder to design a DB engine with the functions supported by the original DB itself [3]. The research area of SQL queriable encryption has tried to solve this problem; nevertheless it is not still enough for statistical analysis.

Another method is privacy preserving in data mining (PPDM) which does not need heavy overhead in comparison to encryption. Some PPDM techniques are not suitable for medical research, while some of PPDM is able to do statistical analysis. Yet, these PPDM methods are not also still enough for statistical arithmetic.

Objectives and contribution
In this paper, we propose the optimized privacy preserving and efficient arithmetic operable database

system for medical research, which satisfies the bilateral features of HI and overcomes the weakness of the previous works in this research area.

The main method is the pseudorandom permutation (PRP) in column level for HI, and the encryption for ID and personal data. Therefore, *our scheme can meet privacy requirements*, especially including *unlinkability* between a permuted element and the data subject. *At the same time, it is possible for statistical arithmetic operation* in the statistically analysable database (SADB). That is, we can break the relation between the data and the data subject, where the data subject is the person whose personal data are collected, hold or processed.

Consequently, *our scheme in HI to be permuted has almost similar performance to general DB*, because it does not take the effects of encryption despite of cryptographic function (permutation). *It can support most kinds of queries such as in a general plaintext database*. The time and accuracy for statistical analysis over changed (permuted SADB) data in a datacenter is also the same as a general database, even inter-column operations such as "AND" query. While previous schemes based on a *column-level encryption or permutation (data swapping) may be inefficient for inter-column operation, the proposed system privacy preserving and statistically analysable database (PPSADB) solves the problem* by applying the same PRP to the columns which need inter-column operations (refer to 2.4.1).

Whereas another problem of previous researches is *the hardness for dynamic database*, our system *PPSADB also overcomes the problem* by managing two kinds of database; encrypted database (*EnDB)* and *SADB*.

Related works

As related works, two research areas are addressed; the aggregated SQL queries on encrypted data and PPDM.

In prior research, there are various active studies to solve aggregated SQL queries on encrypted data with largely three kinds of techniques;

1) Bucketization (partition) [4−6]: The numeric data domain is partitioned into a set of partitions and an identifier is randomly selected for each partition. This method is only possible for rough range queries, but impossible for precise range queries and arithmetic operations.

2) Special indexing method: Fast comparison encryption (FCE) [7] and order preserving encryption scheme (OPES) [5] fall under this method. But, some of these schemes require (partial) decryption in a server. This method also does not consider arithmetic operation.

3) Privacy homomorphism (PH) [4, 8, 9]: Decrypting the arithmetic operation on encrypted data is equal to the arithmetic operation on the plaintext data like this; $D(E(a) \circ E(b)) = a \circ b$. However, in [6], PH of [3] is insecure against a ciphertext-only attack.

In [4], [10], and [11], Hacigumus et al. proposed a method for range queries on encrypted data in database as a service (DAS) model. They regard the database service provider as an untrustworthy party. For SQL queries, they utilize the partitioning method with the concept of 'bucketization'. In [4], they use 'privacy homomorphism'. PH allows basic arithmetic (+, −, ×) over encrypted data. In [11], they study the problem of query optimization in the encrypted database system. Their schemes make it possible all kinds of the SQL queries over encrypted data. However, since PH suggested in [4] is insecure against a ciphertext-only attack [6], the server is able to know the corresponding plaintext.

In Ozsoyoglu et al. [12] have been considered an attribute (field)-level encryption for relational databases under an untrusted server. They propose a family of order preserving open-form and closed-form homomorphic encryption/ decryption function. SQL queries, which are expressible in relational algebra on encrypted database, have no extra query processing cost except for the decryption of final output. [13] is also the following work of [12]. In that paper, they quantify the additional costs incurred when executing aggregate nested SQL queries and they show detailed experimental results. However, they did not consider SQL queries with arithmetic expressions and aggregate functions. Order preserving encryption in [13] has a potential of leaking information, which would lead to compromise.

In Mykletun and Tsudik [6] have been proposed an alternative for handling aggregation queries on the server without homomorphic encryption functions. They explore a variant of DAS and mixed DAS model, where some attributes are sensitive and thus they are encrypted while others are not sensitive and thus left in the clear. However, their alternative cannot implement perfect aggregated queries because they do not consider the arithmetic over encrypted

data but concentrate on bucketization (partition). In [14] and [15], in order to enable a database intruder not to determine which attributes have been tested randomly permute attributes in an encrypted table. They add some meta-data to eliminate the need for testing an equation and replace the sequential search with a binary search so that they can improve the efficiency. However, these schemes also do not consider the arithmetic and allow limited SQL queries over encrypted data.

In Boneh and Waters [14] propose the public key systems supporting queries on encrypted data with tokens. The tokens are produced by a secret key to test any supported query predicate. They construct the systems for comparisons, subset queries, and conjunctive versions of these predicates. For these new constructions, they introduce a primitive, hidden vector encryption (HVE). Shi and Waters [16] and Katz et al. [17] developed methods in public key predicate encryption system, where Shi et al.'s scheme supports the delegation of capabilities and conjunctive queries. Katz et al.'s scheme supports "inner product" queries and it is known as the most expressive scheme to date. They showed that any predicate encryption scheme supporting "inner product" predicates can be used as a building block to construct predicates of more general types such as disjunctions, polynomials, and so on.

As another related research area to our proposed scheme, we address the publishing PPDM. PPDM techniques include all of the things that can be used to induce some knowledge's from data with preserving privacy. PPDM has been worked in various perspectives by many researchers, so that this paper deals with only publishing PPDM techniques which are more closed to our conditions and methods. The representative privacy models of publishing PPDM can be largely classified into five categories; *k*-anonymity, *l*-diversity, *t*-closeness, personalized privacy, ε-differential privacy.

In [18] and [19], Samarati and Sweeney has been proposed *k*-anonymity model, whose main idea is that if one record of the table has some value Quasi identifier (Qid, non-sensitive data) [*Quasi-identifiers can become personally identifying information, when combined. This process is called re-identification*], *k-1* other records have the same value Qid. It means that the minimum size of Qid attributes is at least *k* and is called "k-anonymouse". The *k*-anonymity model is one of the most known models and the first developed privacy model. Aggarwal [20] called the set of k records as "equivalence class", recently, He et al. [21] proposed semi-homogenous generalization method in cloud computing based on *k*-anonymity technique.

The important problem of *k*-anonymity is that no consideration is taken into sensitive attributes when forming the k-anonymized dataset. Hence, it may happen that the sensitive attributes in equivalent classes are equal for all k records. That makes it possible to know the same private information (sensitive attribute) for k owners of k records. To solve this problem, Machanavajjhala et al. [22] suggested a new technique of *l*-diversity. Based on *k*-anonymity, they increased the diversity of sensitive values within the equivalence classes. There should be at least *l* distinct values for the sensitive attribute in each Qid group (equivalence class). Therefore, *l*-diversity privacy model satisfies *k*-anonymity, where $k = l$ [23]. As for health data research, Kim et al. [24] worked with *l*-diversity in.

The weakness of *l*-diversity is that any probable information about record owners can be inferred when the overall distribution of a sensitive attribute is skewed. Li et al. [25] solved this problem with *t-closeness* privacy model, where the distribution of the sensitive values in each equivalence class should be close to the corresponding distribution in the original table and the closeness should be within *t. t-closeness* uses the earth mover distance (EMD) function to measure the closeness between two distributions (original table and the same attribute in any equivalence class) of sensitive values and [20] uses variational Kullback-Leibler (KL) distance. The disadvantage of *t-closeness* is that correlative information between quasi-identifier, and sensitive attributes is lost as *t- closeness* decreases and privacy increases.

Xiao and Tao [26] addressed a new concept of privacy model "personalized privacy", where record owners can define their privacy level. Each sensitive attribute has a taxonomy tree and each record owner defines a guarding node under the tree. When an attacker infers any domain sensitive value within the subtree of his guarding node with a probability, the record owners' privacy can be violated (breach). [27, 28] are in the same line of research. This model has the weakness that it is hard to implement in practice.

The previous privacy models have focused on protecting the record owner's identity, or preventing the inference of sensitive values from anonymized

records, which have not yet measured how the presence of a record impacts owner's privacy. Dwork [29] proposed ε-differential privacy model which measures how the risk of individual privacy disclosure is different between the presence and the absence of the individual's record in the published data. In this paper, Dwork formally proved that ε-differential privacy can guarantee the privacy against attackers with arbitrary background knowledge. Thereafter, Dwork had showed other proofs in [30]. As for the related works to E-health with ε-differential privacy, there are [31] and [32].

From the summarized related research, we find that anything of both researches cannot accurately provide all types of SQL queries and the arithmetic operation for statistical analysis. Although [4] can support the most various types of SQL queries, it is insecure against a ciphertext-only attack. Additionally, the hardness of inter-column operation and dynamic database has been an unsolved problem until now. Therefore, we propose the efficiently arithmetic-operable database system for sensitive medical information.

The rest of this paper is organized as follows. Section 2 describes basic contents of the paper such as application environments including entities and attribute classifications, notations, algorithms, and system processes. Section 3 presents security setting and section 4 construct our system *PPSADB* based on the algorithms. In section 5, we show the formal security proofs. We discuss the privacy of *PPSADB* and compare our proposed system with other previous works in section 6.1 and 6.2. Finally, we conclude the paper in section 7.

2.Preliminaries

2.1Application environments and system details

2.1.1Application

Our application environments include all health record systems from the database management system (DBMS) for EHR/EMR of hospitals to the internet scale of database systems for remote medical system/website services. Even the data stored in the DBMS of hospital server should be published when medical research is needed. The data stored in the remote server, such as datacenter of cloud services may be requested for research resources, or be shared with other organizations or people. Hence, our application can be summarized as largely two categories;

Application 1. Internal storage systems of medical organizations such as hospital or health authorities and so on.

Application 2. Network based storage systems of medical organizations such as remote medical system or web services such as "PatientsLikeMe".

Therefore, we construct the trustworthy database system for sensitive HI to keep the data's security and privacy and at the same time to do statistical analysis.

2.1.2Entities

In our scheme, there are four entities; user U, organization server Os, datacenter server Ds, requestor R.

A user is a member or a customer of an organization, and he/she is a subject or owner of the data. He/she registers at the organization.

An organization server (Os) plays a similar role of a system administrator and it is a trusted party. Os should manage all users' data with safety and keep key-related information, schema information such as attribute names, and all keys only, except for users' private keys, to encrypt and decrypt the data. In application 1, Os stores users' all registered data in their server, while in application 2 Os does not store them in their server but in datacenter server Ds. One of the most important tasks is that Os generates the encrypted database and statistical analysable database with the registered users' information and cryptographic methods.

A datacenter server (Ds) is a remote server, such as datacenter of cloud service. It is vulnerable to a compromise and an outside attacker who can access to the data. All data in Ds are encrypted, and there is no decryption process by Ds.

A requestor is the person who needs statistical medical research. He queries Os or Ds manager for SADB.

2.1.3The types of attributes

The database table consists of three types of attributes.

Identifier (ID) is a user name or an identification which can identify a person. ID is encrypted with pseudorandom function (PRF) and placed in the first column every table.

Personal information (PI) includes the user's address, phone number, job, and so on.

HI includes all information for medical research. It can be classified two types of data; text data (TD) and numeric data (ND).

Text Data is expressible in characters; sex, diagnosis, symptom, past history, smoking habit,

liquor, and so on.
Numeric Data is expressible in numeric; age, measure of diagnostic or blood test result, vital sign, and so on

ID and PI are kept encrypted from registration time; HI is encrypted with Os secret key and stored. When receiving the request for medical research, the encrypted HI is decrypted and permuted in plaintext to make *SADB*.

2.2Notations

The notations of PPSADB system are addressed in *Table 1*.

The following *Figure 1* shows the logic structure of the proposed system PPSADB.

Table 1 The notations of PPSADB

S	Security parameter of PPSADB system
$F_{K'}$	PRF family
Π_K	PRP family
$f(\cdot)$	PRF
$\pi(\cdot)$	PRP
k, k'	Secret keys for $\pi(\cdot)$, $f(\cdot)$
T_i	Table identifier
A_i	Attribute identifier of i-th attribute
k_{ui}	User's private key
Ko	Secret key of an organization server
e_{ij} $(e_{i\text{-}j})$	Element of the i-th column, j-th row
C_{ij}	Encryption value of e_{ij}
D_{ij}	Decryption value of e_{ij}
$P_{ij'}$	The value for the permuted row e_{ij}
t	Total number of PI attributes
p	Total number of TD attributes of HI
q	total number of Numeric Data (ND) attributes of HI
n	total number of users (tuples or rows)

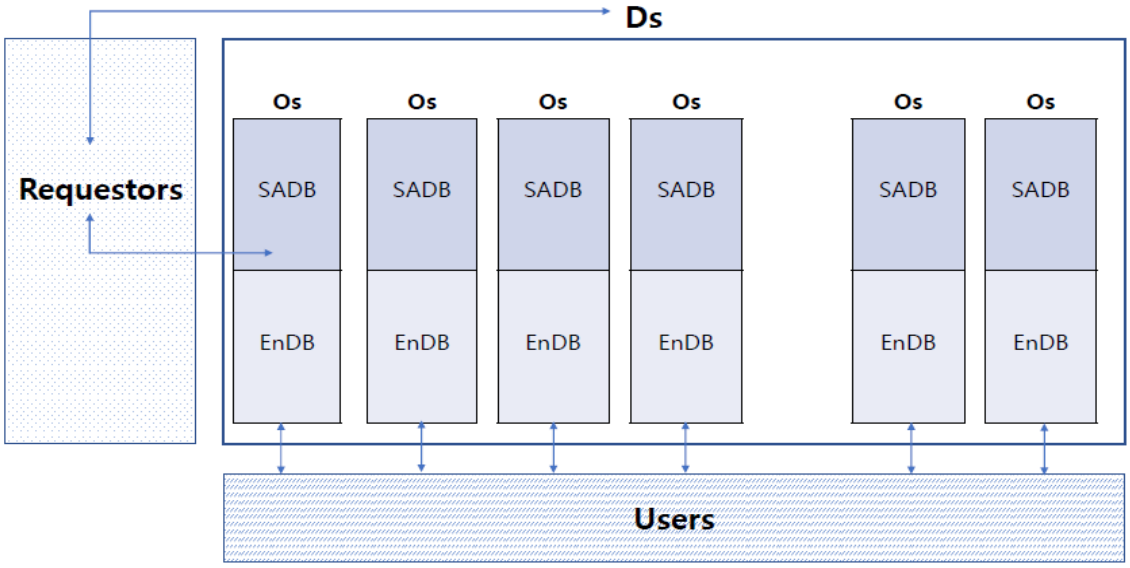

Figure 1 Logic structure of PPSADB

2.3Algorithm for PPSADB

Our scheme SADB has five algorithms as follows.
▪ ***SetUpSys*(1^S, $1^{S'}$)** - This algorithm takes an input as security parameters 1^S, $1^{S'}$ and produces λ which determines system parameters such as the size of database or encryption/decryption algorithms.
▪ ***Registration*(*ID, PI, HI*)** - Taking input as identifier *ID*, personal information *PI*, health information *HI*, this algorithm outputs original database *OgDB*.

▪ ***GenEnDB(λ, OgDB)*** - Given system parameter λ and original database *OgDB*, it outputs encrypted database *EnDB*.
▪ ***Req(Q_Rct, EnDB)*** - It takes an input as query for research content *Q_Rct*, *EnDB*, and outputs statistical analyzable database *SADB*.
▪ ***DnAL(Rct, SADB)*** - Given *Rct and SADB*, this algorithm outputs the result *Rt*.

2.4System processes

The proposed system consists of three processes as follows.

A. System setting

With the algorithm *SetUpSys(1S, 1S′)*, the system is set for privacy preserving *SADB*.

B. Data management

Users' information is registered and organization server Os produces and manages encrypted database *EnDB*.

C. Publishing

If a requestor asks database publishing for medical research, Os changes the *EnDB* to *SADB*. Then, Os or Ds publishes the *SADB* and the requestor downloads and analyses it.

2.4.1Request processes

A. Os and Ds normally keeps encrypted database *EnDB*.

B. If a requestor queries Os or Ds with *Q-Rct* (query for research contents), Os produces *SADB* and it is published by Os (application 1) or Ds (application 2). In the case of application 2, Ds usually keeps *EnDB*, and asks Os to produce *SADB* when *Q-Rct* is invoked. The produced *SADB* by Os manager is published in Ds.

C. The point to note is that the attributes for conjunctive ("*AND*", ∧) query should be informed to Os in requesting *Q-Rct*, so that Os runs the same permutation function for the corresponding attributes. This method makes *inter-column operation* and almost all kinds of queries possible with keeping permuted status.

(An example of "AND" query: Count the number of users (patients) from "age">50 AND "BP (blood pressure)">140 from the table T_1). In this case, the requestor should inform Os that two attributes of "age" and "BP" are used for "AND" query.

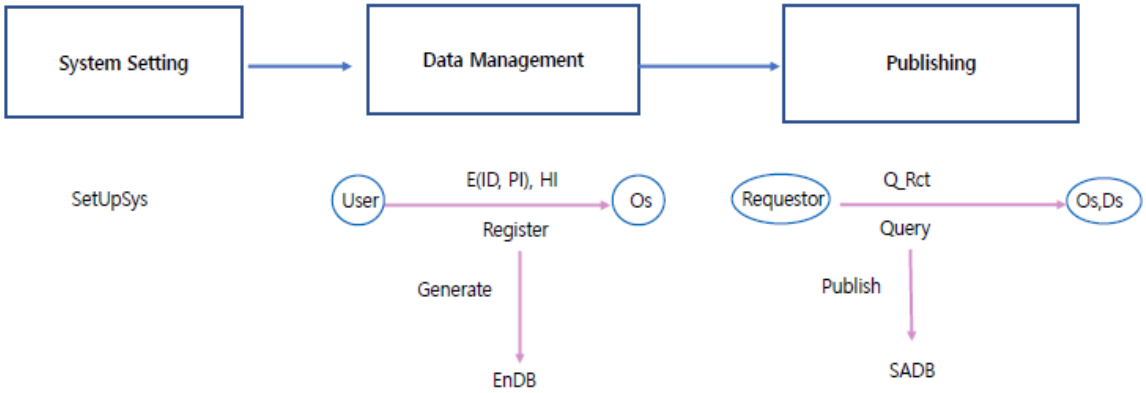

Figure 2 System process of PPSADB

3.Security model

3.1Security building blocks

Definition 1. Pseudo random permutation (PRP). We say a permutation family $\{\Pi_K: \{0,1\}^n \rightarrow \{0,1\}^n \mid K \in \{0,1\}^S\}$ is pseudo-random if it satisfies the following;

- Given $x \in \{0,1\}^n$ and $k \in \{0,1\}^S$, there is a probabilistic polynomial time (PPT) algorithm to compute $\Pi_K(x)$.
- For any PPT oracle algorithm A, the following value is negligible in S;

  $|Pr_{k \leftarrow \{0,1\}s}[A^{\Pi k}(1^S)=1] - Pr_{\phi \leftarrow U\phi}[A^{\phi}(1^{S'})=1]|$,

  Where U_ϕ is the set of all the permutations on $\{0,1\}^n$ [33].

Definition 2. Pseudo random function (PRF). We say a function family $\{F_K:\{0,1\}^l \rightarrow \{0,1\}^m \mid K' \in \{0,1\}^{S'}\}$ is pseudo-random if it satisfies the following;

- Given $x \in \{0,1\}^l$ and $k' \in \{0,1\}^{S'}$, there is a PPT algorithm to compute $F_{K'}(x)$.
- For any PPT oracle algorithm A, the following value is negligible in S';

  $|Pr_{k \leftarrow \{0,1\}s'}[A^{Fk'}(1^{S'})=1] - Pr_{g \leftarrow Ug}[A^{g}(1^{S'})=1]|$,

  Where U_g is the set of all the functions mapping $\{0,1\}^l$ to $\{0,1\}^m$ [33].

3.2Chosen plaintext attack on database (CPA-DB)

Based on the security model in [34], which is a kind of chosen plaintext attack (CPA) model, we modify it in accordance with our scheme and environment. It is called CPA-DB.

Definition 3. CPA-DB. We consider a security game between an attacker A and a challenger C.

Setup. A requests the *SADB* publishing, which is composed of n records $R_i = (ID_i, PI_i, HI_i (TD_i;ND_i))$, $1 \leq i \leq n$. C runs algorithm $SetUpSys(1^S, 1^{S'})$, *Registration(ID, PI, HI)*, *GenEnDB(λ, OgDB)*, and *Req(Q_Rct, EnDB)*. Then, C produces statistical analysable database *SADB* and sends it to A.

Queries. A is allowed to query C with *Q_Rct (plaintext)* and receives *SADB* (permuted and encrypted mode). With this *SADB*, A invokes algorithm *DnAL(Rct, SADB).*

Challenge. After making some queries, A decides on a challenge by picking two elements E_0, E_1 ($\in A_i$) and sends them to C. A must not have queried the query *Q_Rct* belonged to E_0 and E_1($\in A_i$). C chooses $b \leftarrow \{0, 1\}$ and gives A the encrypted and permuted E_b (*SADB* mode). The challenge for A is to determine b. After the challenge is issued, A is not allowed to query C with the *Q_Rct* belonged to E_0 and E_1($\in A_i$).

Response. A eventually outputs a bit b', representing its guess for b. The advantage of A in winning this game is defined as $Adv_A = | Pr[b = b'] - 1/2 |$. Adversary $A(t, \varepsilon, q)$ is said to have an ε-advantage if $Adv_A > \varepsilon$ after A takes at most t times and makes q queries to the challenger.

4.Construction of PPSADB

We construct our scheme privacy preserving *SADB* according to five algorithms.

4.1System setting

4.1.1SetUpSys(1^S, $1^{S'}$) construction

- **Input;** 1^S and $1^{S'}$
- **Output;** $\lambda=(\pi(\cdot), f(\cdot), k, k')$

Given security parameters 1^S and $1^{S'}$, the algorithm ***SetUpSys*** outputs system parameter $\lambda=(\pi(\cdot), f(\cdot), k, k')$. Where $k \leftarrow K \in \{0,1\}^s$ and $k' \leftarrow K' \in \{0,1\}^{s'}$ are secret keys. $f(\cdot) \leftarrow F_{K'}$ and $\pi(\cdot) \leftarrow \Pi_K$ are PRF and PRP respectively. $F_{K'}$ and Π_K are called pseudorandom function family and PRP family.

A database table T consists of *ID* attribute, t *PI* attributes, m *HI* attributes including p text-data attributes, and q numeric-data attributes ($m=p+q$) like this; $T = (ID, A_1, A_2, \ldots A_t, A_{p1}, A_{p2}, \ldots A_{pp}, A_{q1}, A_{q2}, \ldots A_{qq})$. Let A_i be an attribute identifier of i-th attribute. ID attribute is always placed in the first column A_1 every table and plays a role of a primary key in relational DB query.

4.2Data management

The data management process consists of two alghorithms; *Registration(ID, PI, HI)* and *GenEnDB(λ, OgDB).*

4.2.1Registration(ID, PI, HI) construction

- **Input;** *ID, PI, HI*
- **Output;** *OgDB*

At the registration process, every user submits his identifier *ID*, personal information *PI*, and health information *HI* to organization server *Os* manager.

With PRF f and each user's private key k_{uj}, *ID and PI* should be encrypted automatically and simultaneously with input. *Os* produces original database *OgDB* in cyphertexts for *ID / PI, and* in plaintexts for *HI* .

- $C_{0j} = f_{Kuj}(ID_j)$
- $C_{ij} = f_{Kuj}(PI_{ij})$, $1 \leq i \leq t$, $1 \leq j \leq n$

4.2.2GenEnDB(λ, OgDB) construction

- **Input;** *λ, OgDB*
- **Output;** *EnDB*

Using λ, an organization server encrypts an original DB *OgDB* in column(attribute) level.

- ID and PI (Personal Information). At registration time, these attributes are already encrypted with the user's private key.
- HI (Heath Information). These data should be encrypted with Os's secret key.

- $k_{ij} = f_{Ko}(i, j)$
- $C_{ij} = f_{kij}(e_{ij})$

Ko is the secret key of an organization server and e_{ij} is an element of the i-th column, j-th row. k_{ij} plays a role of the secret key for an i-th attribute and j-th row. C_{ij} is the encryption value of e_{ij}.

- Attribute Name (AN). Each table identifier T_i is selected randomly. For each attribute A_i, $f_{Ko}(A_i \| T_i)$ is the encryption of the i-th attribute identifier.

4.3Publishing

4.3.1Req(Q_Rct, EnDB)

- **Input;** *Q_Rct, EnDB*
- **Output;** *SADB*

If a requestor queries Os or Ds with the research content *Q_Rct*, Os produces statistical analysable database *SADB* from *EnDB*.

- ID and PI (Personal Information). These attributes are encrypted with the user's private key from the registration time.
- HI (Heath Information). Os decrypts HI attributes with k_{ij}; $D_{ij} = f^{-1}_{kij}(C_{ij}) = f^{-1}_{kij}(f_{kij}(e_{ij})) = e_{ij}$. Then, HI attributes are separately permuted column by column. A PRP π_i is randomly selected for each A_i.

- $\pi_i(j) = j'$
- $P_{ij'} = e_{ij}$

j' is the permuted row of e_{ij} and $P_{ij'}$ is the value for the permuted row e_{ij}.

4.3.2DnAL(Rct, SADB)

- **Input;** *Rct, SADB*
- **Output;** *Rt*

The requestor, who queried with *Q-Rct*, downloads the published *SADB*. He analyses *SADB* statistically and outputs the result *Rt*.

5.Security analysis

In this section, we show that the proposed system PPSADB is secure.

Theorem1. If *f* is PRF, *π* is PRP, and the key material is chosen as described above, then privacy preserving *SADB* system is secure according to the security game CPA-DB.

In *SADB* construction, we use PRF, which is generally known and proved as a secure algorithm, for encryption of ID and PI without any additional technique. Hence, it can be said that the parts using PRF is secure. However, although we use a secure PRP, we do not apply the element value itself to PRP but apply the row position of the element to PRP. In PPSADB, since PRF and PRP are used independently without any inter-operation, we only have to prove the security for the parts using PRP for the proof of Theorem 1.

Lemma1. If *π* is PRP, then PPSADB for HI is secure according to the security game CPA-DB.

Proof. We prove it with contraposition. We assume that PPSADB for HI is not secure according to the security game CPA-DB. Let A be an attack algorithm that wins the game CPA-DB with advantage ε. We construct an algorithm β which can solve the problem about π is PRP or random permutation.

β can access an oracle Ω_π for the unknown permutation π. β substitutes the values of π through the queries to the oracle Ω_π. β uses an algorithm A as a subroutine and simulates algorithm A using Security Game CPA-DB.

Setup. A requests the *SADB* publishing with *n* tuples R_j, $(1 \le j \le n)$. The tuple is composed of ID, PI(personal information), and HI(health information) attributes like this; $R_j = \{(ID_j, PI_j, HI_j)|PI_j = (e_{1\text{-}j}, e_{2\text{-}j}, \ldots e_{t\text{-}j}), HI_j = (TD_j||ND_j) = (e_{t+1\text{-}j}, e_{t+2\text{-}j}, \ldots e_{t+p\text{-}j}, e_{t+p+1\text{-}j}, e_{t+p+2\text{-}j}, \ldots e_{t+p+q\text{-}j})$. β produces *SADB* and sends it to A.

Queries. A makes some queries to β.

Challenge. After making some queries, A selects two elements $E_0 = e_{i\text{-}j0}$ and $E_1 = e_{i\text{-}j1}$ in *i*-th attribute A_i, where $t+1 \le i \le t+p+q$, j_0 *and* $j_1 \in [1, n]$, $j_0 \neq j_1$. A must not have queried the queries belonged to A_i. C randomly chooses $b \leftarrow \{0, 1\}$ and gives A the permuted row number $j'_b = \pi_i(j_b)$ of E_b. The challenge for A is to determine *b*. Thereafter, A is not allowed to make a query belonged to A_i.

Response. Finally, A outputs a guessing bit b'. If $b = b'$, β outputs 1 indicating that β guesses *π* is a PRP, otherwise 0. β can know the PRP challenge with the same advantage as A in winning game CPA-DB.

We show that β can solve the problem about whether *π* is pseudorandom or random permutation with non-negligible probability. Accordingly, the advantage of β in winning this experiment is;

$$Adv_\beta = Pr[Exp_\beta^{PR} = 1] = Pr[b' = b]$$

$$= \Pr[b' = b|b=1] \cdot \Pr[b = 1] + \Pr[b' = b|b=0] \cdot \Pr[b = 0]$$

$$= \Pr[b' = b|b=1] \cdot \frac{1}{2} + \Pr[b' = b|b=0] \cdot \frac{1}{2}$$

$$= \Pr[b' = 1|b=1] \cdot \frac{1}{2} + \Pr[b' = 0|b=0] \cdot \frac{1}{2}$$

$$= \Pr[b' = 1|b=1] \cdot \frac{1}{2} + (1 - \Pr[b' = 1|b=0]) \cdot \frac{1}{2}$$

$$= \frac{1}{2} + \frac{1}{2}(\Pr[b' = 1|b=1] - \Pr[b' = 1|b=0])$$

$$= \frac{1}{2} + \frac{1}{2}(\Pr[Exp_A^{CPA-DB-1} = 1] - \Pr[Exp_A^{CPA-DB-0} = 1])$$

$$= \frac{1}{2} + \frac{1}{2} Adv_A^{CPA-DB} = \frac{1}{2} + \frac{1}{2}\varepsilon$$

We showed the existence of β which can solve whether *π* is pseudorandom or random with the probability more than 1/2. Accordingly, by contradiction, it can be said that PPSADB is secure.

6.Discussion

6.1Privacy preserving SADB

6.1.1Privacy requirements

Privacy can be defined as fair information practices (FIPs) meaning the ability of individuals to control the disclosure and use of their personal data. Privacy technologies safeguard personal privacy by minimising or eliminating the collection of identifiable data (quasi-identifier). Hence, extended security criteria for systems with high privacy requirements should cover a diversity of privacy enhancing security aspects such as the followings [35];

- **Anonymity.** Anonymity ensures that a user may use a resource or service, without disclosing the user's identity.
- **Pseudonymity.** Pseudonymity can protect the user's identity in cases where anonymity cannot be provided, e.g. if the user has to be held accountable for his activities.
- **Unobservability.** It ensures that a user may use a resource or a service without others being able to observe that the resource or service is being used. Furthermore, it has to be prevented that an attacker can link various information about a user to the profile that could finally be used to re-identify the user.
- **Unlinkability.** It ensures that a user may make use of resources and services without others being able to link these uses together.

6.1.2Privacy and security of PPSADB

The way of encryption is different on the type and status of information. ID and PI are automatically encrypted with users' private key at the same time of data input for registration and kept with the encrypted status all the time. All users' private keys are not stored and only users know their private key. It makes even Os impossible to know ID and PI of the user. In *EnDB* status, all the elements of HI (For medical research) attributes are encrypted with each different secret key which is induced from Os' secret key and each cell's row number and column number. These encryptions provide anonymity, pseudonymity and unobservability.

In SADB status changed from *EnDB*, Os cannot know ID and PI because of the encryption with each user's private key. HI attributes are permuted within a column in a plaintext status. The randomly selected permutation function for each attribute can provide unlinkability, which has the important meaning in publishing research database, in that this method is not to encrypt data itself, but to break the matching between data and the owner of the data in plaintext.

6.2Comparative analyses

We showed the security and privacy protection of PPSADB in section 5 and 6.1. In section 6.2, we analyse the difference from the previous works.

6.2.1Two kinds of DB

The proposed system manages two kinds of DB; *EnDB* and *SADB*. At usual time, the system keeps *EnDB*, which is updated whenever a new user joins the system or the registered user leaves in real time. Therefore, our proposed system PPSADB falls under the type of dynamic database. With this dynamic *EnDB*, Os produces *SADB* for publishing when it receives the request *Q_Rct*. *SADB* is the same concept such as the published datasets of US department of human and healthcare services (HHS) for researchers. As for the resource dataset, *SADB* does not need to satisfy the dynamic database.

The improvements of *SADB* are resulted from applying PRP function to HI attributes which are statistically analysed for medical research. The permutation function can lead to the possibility for almost all kinds of arithmetic calculations with keeping security. This is because *SADB* can implement the queried analysis directly on the values of HI attributes which are not encrypted and modified, while the values of previous researches need some additional modifications or noises insertion. Hence, *SADB* can guarantee the accuracy of results and the similar level of performance to other general databases. The only thing to be considered is the corresponding attributes for conjunctive queries should be informed to Os when requesting *Q_Rct*. Os applies a same permutation function to the attributes for the conjunctive query, which makes "*AND*(∧)" query possible, and then Os publishes the produced *SADB*. Nevertheless, it has no problem because *SADB* does not need to be dynamic database.

In addition, the registered users may query Os or Ds with their health status such as medical test results because PPSADB is also database system for storage. When users ask their data, the first their IDs are encrypted with their private key and some HI is queried. Then, Os searches the encrypted ID and the corresponding HI attributes on *EnDB*. Finally, the corresponding cells are decrypted with Os' secret key and the results are sent to the user.

6.2.3The comparison with the previous privacy models

Table 2 shows the protective coverages against privacy attacks for the previous publishing privacy models and PPSADB.

Table 2 The comparison with the previous privacy models

	Attack Model			
Privacy Model	**Record linkage**	**Attribute linkage**	**Table linkage**	**Probabilistic attack**
k-anonymity	√			
l-diversity	√	√		
t-closeness		√		√
personalized privacy		√		
ε-differential privacy			√	√
PPSADB	√	√	√	√

Record linkage is said that an attacker links a record owner to the owner's record in a published data table. Linking a record owner to the sensitive attribute in a published data table, or to the published data table itself, we say them as attribute linkage, or table linkage. The attacker of table linkage tries to determine the presence or absence of the victim's record in the published table. From a different perspective than these attack models, the probabilistic attack means that the attacker tries to change the probabilistic belief on the sensitive victim's information after accessing the published data.

PPSADB guarantees 'Unlinkability' as mentioned in the section 6.1.2. Therefore, PPSADB can preserve privacy from the record, attribute, and table linkage. PPSADB can also preserve the privacy from probabilistic attack because our method for HI attributes is PRPs without any noise insertions or modifications of the values themselves.

7.Conclusion

Differently from other encrypted database schemes, the performance in our scheme is similar to the general plaintext database because of cryptographic permutation function. It enables query operations to be implemented directly on permuted data. PPSADB manages two kinds of database; EnDB and SADB. Consequently, our proposed scheme can keep all data securely with EnDB and support the arithmetic operations and almost all kinds of SQL queries over permuted databases with SADB. Moreover, it can satisfy privacy requirements and solve the problems of the previous researches including inter-column operations and dynamic DB.

However, our system still has the weakness, in that the attributes for AND query should be informed to Os manager in advance. Therefore, the researches about more improved inter-column operations on encrypted database need to be worked.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (Ministry of Education, NRF-2017R1D1A1B03029488).

Conflicts of interest

The author has no conflicts of interest to declare.

References

[1] https://www.patientslikeme.com. Accessed 20 June 2018.
[2] https://curetogether.com. Accessed 26 June 2018.
[3] Park HA. Encrypted similarity search feasible of keyword index search schemes. International Journal of Internet Technology and Secured Transactions. 2016; 6(3):231-57.
[4] Hacıgümüş H, Iyer B, Mehrotra S. Efficient execution of aggregation queries over encrypted relational databases. In international conference on database systems for advanced applications 2004 (pp. 125-36). Springer, Berlin, Heidelberg.
[5] Agrawal R, Kiernan J, Srikant R, Xu Y. Order preserving encryption for numeric data. In proceedings of the international conference on management of data 2004 (pp. 563-74). ACM.
[6] Mykletun E, Tsudik G. Aggregation queries in the database-as-a-service model. In IFIP annual conference on data and applications security and privacy 2006 (pp. 89-103). Springer, Berlin, Heidelberg.
[7] Ge T, Zdonik S. Fast, secure encryption for indexing in a column-oriented DBMS. In international conference on data engineering 2007 (pp. 676-85). IEEE.
[8] Ferrer JD. A new privacy homomorphism and applications. Information Processing Letters. 1996; 60(5):277-82.
[9] Domingo-Ferrer J. A provably secure additive and multiplicative privacy homomorphism. In international conference on information security 2002 (pp. 471-83). Springer, Berlin, Heidelberg.
[10] Hacigümüş H, Iyer B, Li C, Mehrotra S. Executing SQL over encrypted data in the database-service-provider model. In proceedings of the international conference on management of data 2002 (pp. 216-27). ACM.
[11] Hacıgümüş H, Iyer B, Mehrotra S. Query optimization in encrypted database systems. In international conference on database systems for advanced applications 2005 (pp. 43-55). Springer, Berlin, Heidelberg.

[12] Özsoyoglu G, Singer DA, Chung SS. Anti-tamper databases: querying encrypted databases. In DBSec 2003 (pp. 133-46).
[13] Chung SS, Ozsoyoglu G. Processing aggregate queries over encrypted relational databases. The Technical Report. 2004.
[14] Boneh D, Waters B. Conjunctive, subset, and range queries on encrypted data. In theory of cryptography conference 2007 (pp. 535-54). Springer, Berlin, Heidelberg.
[15] Yang Z, Zhong S, Wright RN. Privacy-preserving queries on encrypted data. In European symposium on research in computer security 2006 (pp. 479-95). Springer, Berlin, Heidelberg.
[16] Shi E, Waters B. Delegating capabilities in predicate encryption systems. In international colloquium on automata, languages, and programming 2008 (pp. 560-78). Springer, Berlin, Heidelberg.
[17] Katz J, Sahai A, Waters B. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In annual international conference on the theory and applications of cryptographic techniques 2008 (pp. 146-62). Springer, Berlin, Heidelberg.
[18] Samarati P, Sweeney L. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report, SRI International; 1998.
[19] Samarati P, Sweeney L. Generalizing data to provide anonymity when disclosing information. In PODS 1998 (p. 188).
[20] Aggarwal CC. Data mining: the textbook. Springer; 2015.
[21] He XM, Wang XS, Li D, Hao YN. Semi-homogenous generalization: improving homogenous generalization for privacy preservation in cloud computing. Journal of Computer Science and Technology. 2016; 31(6):1124-35.
[22] Machanavajjhala A, Gehrke J, Kifer D. L-diversity: privacy beyond k-anonymity. In proceedings of the international conference on data engineering 2006.
[23] Truta TM, Vinay B. Privacy protection: p-sensitive k-anonymity property. International conference on data engineering workshops 2006 (pp. 94-104). IEEE.
[24] Kim S, Sung MK, Chung YD. A framework to preserve the privacy of electronic health data streams. Journal of Biomedical Informatics. 2014; 50:95-106.
[25] Li N, Li T, Venkatasubramanian S. T-closeness: privacy beyond k-anonymity and l-diversity. In international conference on data engineering 2007 (pp. 106-15). IEEE.
[26] Xiao X, Tao Y. Personalized privacy preservation. In proceedings of the international conference on management of data 2006 (pp. 229-40). ACM.
[27] Yuan M, Chen L, Yu PS. Personalized privacy protection in social networks. Proceedings of the VLDB Endowment. 2010; 4(2):141-50.
[28] Komishani EG, Abadi M, Deldar F. PPTD: preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression. Knowledge-Based Systems. 2016; 94:43-59.
[29] Dwork, C. Differential privacy. In proceedings of the international conference on automata, languages and programming 2006 (pp. 1-12). ACM.
[30] Dwork C. Differential privacy: a survey of results. In international conference on theory and applications of models of computation 2008 (pp. 1-19). Springer, Berlin, Heidelberg.
[31] Dankar FK, El Emam K. Practicing differential privacy in health care: a review. Transactions Data Privacy. 2013; 6(1):35-67.
[32] Lin C, Song Z, Song H, Zhou Y, Wang Y, Wu G. Differential privacy preserving in big data analytics for connected health. Journal of Medical Systems. 2016; 40(4):1-9.
[33] Chang YC, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data. In international conference on applied cryptography and network security 2005 (pp. 442-55). Springer, Berlin, Heidelberg.
[34] Goh EJ. Secure indexes. IACR Cryptology ePrint Archive. 2003:1-19.
[35] http://govdocs.ourontario.ca/node/14782. Accessed 26 June 2018.

Hyun-A Park received the BS degree from the Department of Mathematics at Korea University, Seoul, in 2003, and the MS and PhD degrees in Information Security from Korea University, Seoul, in 2005 and 2010, respectively. Currently, she is an Assistant Professor with KyungDong University. Her main research interests include Medical (Health) Information Security, Practical Retrieval System on Encrypted Database Systems. She is interested in Database Security, Access Control, Privacy Preserving in Data Mining (PPDM), Anonymous Communication Channel, Privacy Enhancing Technology (PET), and Cryptographic Protocols.
Email: kokokzi@kduniv.ac.kr