Request-Based Comparable Encryption

Jun Furukawa

NEC Corporation, Kanagawa 211-8666, Japan
j-furukawa@ay.jp.nec.co.jp

Abstract. An order-preserving encryption (OPE) scheme preserves the numerical order of numbers under encryption while hiding their original values in a some extent. However, if all the numbers in a certain domain are encrypted by an OPE, the original numbers can be restored from their order. We introduce a notion of novel encryption scheme "request-based comparable encryption" that provides a certain level of security even when OPEs cannot. A request-based comparable encryption hides original values, but it enables any pair of encrypted values to be compared each other when and only when one of them is accompanied by a "token". We also consider its weaker notion and a concrete construction satisfying it. We consider a request-based comparable encryption complements OPEs and can be an essential security primitive.

Keywords: order-preserving encryption, request-based, database encryption, range query.

1 Introduction

1.1 Background and Motivation

A database (DB) is a system that stores a large amount of data and passes its portions when requested. It has been an indispensable platform for variety of services through the network. Since many DBs store sensitive information, they are potentially vulnerable to abuse, leakage, and theft. Hence, it is important to unfailingly protect confidentiality of their data. An access control is a fairly effective approach for it, but it is helpless if the DB is compromised. Hence, it is desirable to enforce DBs by such an encrypting mechanism that the keys for decryption are kept by only data owners (not DB). This strategy is considered to be especially effective for the database-as-service, and can indeed be found in [12,20,21].

Although encrypting data in a DB can be effective in protecting data, it tends to spoil the availability of the DB since the DB can handle data only in limited manner. This may require users to retrieve all data in the DB, decrypt them, find necessary data among them, and process them all by himself. This imposes a large amount of computation, communication, and the memory on the user.

A searchable encryption [2,5,17,19] enables DBs to search necessary data without decrypting them, and an order-preserving encryption (OPE) [1,8,9] enables DBs to recognize the numerical order of data without decrypting them.

J. Crampton, S. Jajodia, and K. Mayes (Eds.): ESORICS 2013, LNCS 8134, pp. 129–146, 2013.

These ability recovers the availability of DBs by enabling them to return only the ciphertexts of data that are required by the users.

A relational database (RDB) [16], which is the most widely-used database nowadays, frequently selects data in a certain range from a table. This task can be done by an OPE even if data are encrypted. Since such selection of data drastically reduces the amount of computation and communication of the users, OPE is considered to be one of pivotal primitives for RDB with encrypted data. This is why the proposal of an OPE [8] immediately received attention from the applied community [18,25,26,29,31,34]. An OPE as well as a searchable encryption plays an important role for CryptDB [29], an encrypted RDB, to mark practical efficiency in TPC-C [32] measure.

Boldyreva et al. proposed an OPE [8] and studied the security of OPEs [9] for its practical use. Their positive result shows that OPEs enjoy reasonable security as long as the number of ciphertexts is sufficiently small compared to the square root of size of the domain of relevant numbers. But nothing is guaranteed in the case the number of ciphertexts is larger than that. Indeed, it is clear, as in the following example, that OPEs fail to hide anything about encrypted numbers in some cases. Consider a set of numbers that includes the all numbers in a domain D and every elements of this set are encrypted by an OPE. If all of these encrypted numbers are given to an adversary, the adversary is able to decrypt all the ciphertexts simply by sorting all of them.

That an OPE has a limitation in its secure use causes a serious concern for encrypted DBs since the OPE is a pivotal primitive for them. Several stronger primitives such as the committed efficiently-orderable encryption (CEOE) [9] that exploits a monotone minimal perfect hash function [3], range query methods in a public key setting [30,11], and searchable encryptions in a public key setting [5,6,10] have been proposed, but these are not sufficient for salvaging the benefit of DBs in the case described above. An order-preserving encryption with additional interactions [28] can enhance the security, but most applications assume that an RDB handles a thread of instructions without such additional interactions. It is now clear that we definitely need a novel cryptographic primitive so as an encrypted DB to function with practical efficiency and security.

1.2 Request-Based Comparable Encryption

In this paper, we propose a novel notion of cryptographic primitive called "request-based comparable encryption (comparable encryption for short)" that complements OPEs. The comparable encryption overcomes the limitation of OPEs just as the searchable encryption in [17,13,22] does the limitation of deterministic encryptions. It is a symmetric key encryption with such an additional mechanism that enables one to compare an encrypted number to other encrypted numbers if and only if the one is given a token associated to this number. Searches in [11] are also triggered by tokens.

Let us consider applying a comparable encryption to an encrypted DB. The DB stores encrypted numbers only and, upon a range query, it receives tokens for

the edges of the range. Then, the DB is able to compare these stored encrypted values with the edge values without interacting with the user[1]. Thus, the DB is able to select out the data which the user required via the query. We emphasize that encrypted values themselves cannot be compared each other unless either of them is an edge unlike the case of OPE. Although the token does leak some numerical orders of the data to the DB, what is leaked to the DB is what the DB needs for processing data with practical efficiency. A protocol such as"private information retrieval" introduced in [14,15,23] leaks less data to DB, but such an approach inevitably requires heavy computational and communicational cost for DBs. This is not practical for realistic DBs and we thus dismiss such an approach.

If a user makes a huge number of range queries to a database and this database accumulates all tokens in these range queries, the database may acquire enough knowledge to decrypt all ciphertexts in some cases. Our approach is no longer effective in such an extreme case as OPE is no longer so. However, real users rarely deposit their data to totally untrusted DBs. The real concerns are that DBs leak their data because of careless system managers, viruses, via unpatched vulnerability of the system, design error, or configuration fault. As long as an intrusion of an adversary is temporal, it succeeds to seize only those tokens that are in insertion or selection queries which are made at the time of the intrusion. An example of temporal intrusion is a leakage of the memory data with respect a query. Such a temporal intrusion only enables the adversary to compare the each element in the stored data with the encrypted numbers in the query. Since such a comparison is already delegated to the DB in the query corrupted, leakage of this result can be considered as the minimum, unavoidable, and acceptable as long as efficiency is required.

1.3 A Weaker Property and Our Comparable Encryption

The introduced comparable encryption is a very promising primitive for practical encrypted DBs. However, we have not completely succeeded to propose an ideal comparable encryption with practical efficiency. We find no definite reason that it is inherently impossible but we have not. As the DB cannot be practical unless with practical efficiency literally, we propose a comparable encryption that has a weaker property than ideal one, but has a stronger security property than OPE and has practical efficiency. In particular, our comparable encryption is such that its tokens leak knowledge more than ideally allowed.

To evaluate the difference of security properties between the ideal one and ours, we first formalized the ideal security requirement and its weaker variant as well. Then, as a measure of the security level of this weaker variant, we evaluate the expected ratio between the number of occasions when a token of an ideal scheme leaks and the number of occasions when a token of a weaker scheme

[1] Since DB receives a sequence of requests at one time to avoid heavy communication and incoherent transaction, DB needs to process requests without interacting with the user.

leaks, which we show to be only at most "2.8". Suppose a temporal intrusion leaked a token as well as encrypted numbers. Then, the probability that this token helps to distinguish any of two encrypted numbers is 2.8 times larger in our scheme than in an ideal scheme.

Our comparable encryption is proved to satisfies this weaker property in the standard model but is sufficiently fast. The length of ciphertext is proportional to the bit length of the maximum number. The size of database shall increase severely if all data are encrypted with our comparable encryption. However, if the encryption is limitedly applied to only highly confidential data that require comparison, the database can remain in moderate size. Such limitation is common when a current product for database encryption such as [27] is used. Hence, although to reduce ciphertext length is highly desirable, our comparable encryption as it is still has practical value. The dominant cost for encryption, generation of token, and comparison are the cost for computing hash values in these processes, whose number of computation is again proportional to the bit length of the maximum number [2], which cost is very light. Considering the merit of efficiency that our scheme enjoys, we consider the weakness of our scheme is not so serious.

1.4 Organization:

The paper is organized as follows: Section 2 introduces the model of comparable encryption and describes its basic functionality. Section 3 presents a concrete scheme of comparable encryption and compares complexity of our scheme with that of OPE. Section 4 introduces the security requirement of ideal comparable encryptions and its weaker variant. Then it evaluates the difference between the two security requirements. Section 5 concludes the paper and poses an open problem.

2 Model

We introduce the model of comparable encryption and a basic property. Comparable encryption is composed of four algorithms, $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Der}$, and Cmp.

Gen: A probabilistic algorithm that, given a security parameter $\kappa \in \mathbb{N}$ and a range parameter $n \in \mathbb{N}$, outputs a parameter *param* and a master key *mkey*. n is included in *param*.

$$(param, mkey) = \mathsf{Gen}(\kappa, n)$$

Enc: A probabilistic algorithm that, given a parameter *param*, a master key *mkey*, and a number $0 \leq num < 2^n$, outputs a ciphertext *ciph*.

$$ciph = \mathsf{Enc}(param, mkey, num)$$

[2] The cost for the decryption is constant if we provide this functionality.

Der: A possibly probabilistic algorithm that, given a parameter *param*, a master key *mkey*, and a number $0 \leq num < 2^n$, outputs a token *token*.

$$token = \mathsf{Der}(param, mkey, num)$$

Cmp: An algorithm that, given a parameter *param*, two ciphertexts *ciph* and $ciph'$, and a token *token*, outputs $-1, 1$, or 0.

$$\mathsf{Cmp}(param, ciph, ciph', token) \in \{-1, 1, 0\}$$

Although we call the scheme *encryption*, it provides no **decryption** algorithm. But such a functionality can be easily provided by appending an ordinary ciphertext $\widetilde{ciph}$ to each comparable encryption ciphertext *ciph* as $ciph|\widetilde{ciph}$ and preparing an ordinary decryption algorithm for it. Then, decryption is straightforward. Although we consider the decryption algorithm is necessary in practice, we omit it in our model for the simplicity of the presentation.

We assume *ciph* and *token* input to Cmp are related so that they satisfy $ciph = \mathsf{Enc}(param, mkey, num)$ and $token = \mathsf{Der}(param, mkey, num)$ for the same *param*, *mkey*, and *num*. The output of Cmp is $-1, 1$, or 0, respectively, when $num < num'$, $num > num'$, or $num = num'$. This requirement is formalized in the following property of completeness.

Definition 1. *We say a comparable encryption is* **complete** *if, for every* $\kappa \in \mathbb{N}$, $n \in \mathbb{N}$, *and* $0 \leq num, num' < 2^n$, *there exist* $param, mkey, token, ciph$, *and* $ciph'$ *such that*

$$(param, mkey) = \mathsf{Gen}(\kappa, n) \ , \ token = \mathsf{Der}(param, mkey, num)$$
$$ciph = \mathsf{Enc}(param, mkey, num) \ , \ ciph' = \mathsf{Enc}(param, mkey, num')$$
$$\mathsf{Cmp}(param, ciph, ciph', token) = \begin{cases} -1 & \text{if} \quad num < num' \\ 1 & \text{if} \quad num > num' \\ 0 & \text{if} \quad num = num' \end{cases}$$

hold with overwhelming probability. Where probability is taken over the distribution of random tapes input to Gen, Enc, *and* Der.

3 Proposed Scheme

3.1 Preliminaries and Overview of Our Scheme

Our construction of comparable encryption exploits prefix-preserving encryption (PPE) [35,4,24]. PPE considers each message as a sequence of blocks. If two messages have the same sequence of n blocks as their prefixes, the encryptions of these messages also have the same sequence of n blocks as their prefixes. But the rest of blocks are different. Thus, a PPE preserves the equivalence of prefix blocks. A PPE as-is does not meet the purpose of our comparable encryption since it enables neither to hide the similarity of two numbers nor to recognize the numerical order of two numbers from their ciphertexts. Our comparable

encryption is similar to PPE in that it also considers numbers as a sequence of blocks, where each block is a bit".

We list here some of the terms necessary in the rest of the paper. Suppose that n is a given fixed number such that $num = \sum_{i=0}^{n-1} b_i 2^i$ and $num' = \sum_{i=0}^{n-1} b'_i 2^i$ with $b_i, b'_i \in \{0,1\}$ for all $0 \leq i \leq n-1$. We let $(b_0, \ldots, b_{n-1})$ and $(b'_0, \ldots, b'_{n-1})$, respectively, represent num and num'. We say the most significant prefix $(n - \ell - 1)$ bits of num is $(b_{\ell+1}, \ldots, b_{n-1})$. We let $\mathsf{MSPBs}(num, \ell) = (b_{\ell+1}, \ldots, b_{n-1})$ denotes this relation.

Our comparable encryption uses PPE ciphertext of a number num as the token of num ($token = \mathsf{Der}(param, mkey, num)$). Note that, if tokens of num and num' are, respectively, $token = \mathsf{Der}(param, mkey, num)$ and $token' = \mathsf{Der}(param, mkey, num')$ and if $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$, then $\mathsf{MSPBs}(token, \ell) = \mathsf{MSPBs}(token', \ell)$ holds. Let $token = (d_0, \ldots, d_{n-1})$. If each ℓ'-th bit of num, i.e. $b_{\ell'}$, is probabilistically encrypted by $d_{\ell'}$, then one can check whether or not $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$ holds for given ℓ (e.g., by decrypting them) using either $token$ or $token'$. But, whether $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$ or not is hidden if the both $token$ and $token'$ are kept hidden. This mechanism enables to compare the similarity of encrypted two numbers only when either of their tokens is given.

When $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$ but $\mathsf{MSPBs}(num, \ell - 1) \neq \mathsf{MSPBs}(num', \ell - 1)$, Cmp compares num and num' by comparing ℓ-th bits of num and num'(b_ℓ and b'_ℓ respectively). For this comparison, $e_\ell = b_\ell + mask_\ell \bmod 3$ is generated with a random looking mask $mask_\ell$, and encryption of e_ℓ is included in the ciphertext of num. Let $mask'_\ell$ and e'_ℓ be also generated in the same manner for num' here. Suppose that $mask_\ell$ and $mask'_\ell$ depend on only on $\mathsf{MSPBs}(num, \ell)$ and $\mathsf{MSPBs}(num', \ell)$ respectively (as well as on the master key), then $mask_\ell = mask'_\ell$ if $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$. Then b_ℓ and b'_ℓ are revealed from e_ℓ and e'_ℓ if b_ℓ and b'_ℓ are different (i.e.,$\mathsf{MSPBs}(num, \ell - 1) \neq \mathsf{MSPBs}(num', \ell - 1)$), since $e_\ell - e'_\ell = b_\ell - b'_\ell = 1 \bmod 3$ if $b_\ell = 1$ but $e_\ell - e'_\ell = 2 \bmod 3$ if $b_\ell = 0$. But b_ℓ and b'_ℓ are hidden if b_ℓ and b'_ℓ are the same (i.e., $\mathsf{MSPBs}(num, \ell - 1) = \mathsf{MSPBs}(num', \ell - 1)$), since $e_\ell - e'_\ell = b_\ell - b'_\ell = 0 \bmod 3$ does not depend on b_ℓ. b_i and b'_i for $i < \ell$ are hidden if $\mathsf{MSPBs}(num, \ell - 1) \neq \mathsf{MSPBs}(num', \ell - 1)$, since $e_i - e'_i \bmod 3$ depends on $mask_i - mask'_i \bmod 3$ which is pseudo-random. If $token$ is designed to reveals e_ℓ and e'_ℓ, one can decide which number (num or num') is greater from their ciphertexts. Note that b_i and b'_i for none of $i \neq \ell$ is revealed.

The above construction of comparable encryption from PPE provides satisfactory functionality of comparable encryption. However, its tokens leak knowledge more than the numerical order of numbers. Suppose that $ciph$ and $ciph'$ are, respectively ciphertexts of two numbers num and num'. From $ciph$, $ciph'$, and the token $token$ of num, one can recognize not only the numerical order of num and num' but also the most significant bit at which num and num' differ. This is not a scheme with an ideal security property, but this is the best we can provide at this moment. And we analyze the negative impact of this leakage later.

3.2 Construction

Now we present the specific construction of our comparable encryption below.

Gen: Suppose a security parameter $\kappa \in \mathbb{N}$ and the number of digit n. Gen first randomly chooses a hash function $\mathsf{Hash} : \{0,1\}^{\kappa} \times \{0,1\}^{4+\kappa+1} \to \{0,1\}^{\kappa}$ and assigns $param = (n, \mathsf{Hash})$. Next, Gen uniformly and randomly chooses a master key $mkey \in \{0,1\}^{\kappa}$. Gen outputs $param = (n, \mathsf{Hash})$ and $mkey$.

Der: Suppose that $param = (n, \mathsf{Hash})$, $mkey$, and a number $num = (b_0, b_1, \ldots, b_{n-1}) := \sum_{0 \le i \le n-1} b_i 2^i$ are given. Der generates

$$\begin{aligned} d_n &= \mathsf{Hash}(mkey, (0, 0^{\kappa}, 0)) \\ d_i &= \mathsf{Hash}(mkey, (1, d_{i+1}, b_i)) \qquad \text{for} \quad i = n-1, \ldots, 0 \end{aligned}$$

Der outputs the token $token = (d_0, d_1, \ldots, d_n)$.

Enc: Suppose that $param = (n, \mathsf{Hash})$, $mkey$, and a number $num = (b_0, b_1, \ldots, b_{n-1})$ are given. Enc first generates $(d_0, d_1, \ldots, d_n) = \mathsf{Der}(param, mkey, num)$ and then randomly chooses random number $I \in \{0,1\}^{\kappa}$. Next, Enc generates

$$\begin{aligned} c_i &= \mathsf{Hash}(d_i, (2, I, 0)) \\ e_i &= \mathsf{Hash}(mkey, (4, d_{i+1}, 0)) + b_i \bmod 3 \\ f_i &= \mathsf{Hash}(d_{i+1}, (5, I, 0)) + e_i \bmod 3 \end{aligned}$$

for $i = n-1, \ldots, 0$. Enc finally outputs ciphertext $ciph = (I, (c_0, \ldots, c_{n-1}), (f_0, \ldots, f_{n-1}))$.

Cmp: Suppose that $param = (n, \mathsf{Hash})$, a pair of ciphertexts $ciph = (I, (c_0, \ldots, c_{n-1}), (f_0, \ldots, f_{n-1}))$ and $ciph' = (I', (c'_0, \ldots, c'_{n-1}), (f'_0, \ldots, f'_{n-1}))$, and a token $token = (d_0, d_1, \ldots, d_n)$ are given.

1. Cmp searches and find j such that

$$\begin{aligned} &(0 \le j \le n-1) \quad \wedge \\ &\big(\forall k \text{ s.t. } j < k < n,\ c'_k = \mathsf{Hash}(d_k, (2, I', 0))\big) \ \wedge \ \big(c'_j \neq \mathsf{Hash}(d_j, (2, I', 0))\big) \end{aligned}$$

   In case

$$\forall k \ \text{ s.t. } 0 \le k < n, \quad c'_k = \mathsf{Hash}(d_k, (2, I', 0))$$

   hold, Cmp outputs 0 and stops.

2. Cmp generates

$$\begin{aligned} e_j &= f_j - \mathsf{Hash}(d_{j+1}, (5, I, 0)) \bmod 3 \\ e'_j &= f'_j - \mathsf{Hash}(d_{j+1}, (5, I', 0)) \bmod 3 \end{aligned}$$

3. Cmp outputs

$$\begin{aligned} 1 \quad &\text{if} \quad e_j - e'_j = 1 \bmod 3 \\ -1 \quad &\text{if} \quad e_j - e'_j = 2 \bmod 3 \end{aligned}$$

Here, input $(c_1, \ldots, c_n)$ are unnecessary. But we include them in the input only for the simplicity of the description.

3.3 Completeness of Our Comparable Encryption

The theorem 1 below guarantees that our scheme successfully compares encrypted numbers.

Definition 2. *We say a function* $\mathsf{Hash} : \{0,1\}^\kappa \times \{0,1\}^\ell \to \{0,1\}^L$ *is a* pseudorandom function *if every poly-time distinguisher D has an advantage in distinguishing whether it is accessing* $\mathsf{Hash}(K,\cdot)$ *with randomly chosen key* $K \in \{0,1\}^\kappa$ *or it is accessing a random function* $R : \{0,1\}^\ell \to \{0,1\}^L$ *with at most negligible probability in* κ.

Theorem 1. *The proposed comparable encryption is complete as long as* Hash *is a pseudorandom function.*

Proof. Let $num = \sum_{i=0}^{n-1} b_i 2^i$, $num' = \sum_{i=0}^{n-1} b'_i 2^i$, ℓ be the largest ℓ' such that $\mathsf{MSPBs}(num, \ell') = \mathsf{MSPBs}(num', \ell')$ holds, $(d_0, \ldots, d_n) = \mathsf{Der}(param, mkey, num)$, $(d'_0, \ldots, d'_n) = \mathsf{Der}(param, mkey, num')$, $(I, (c_0, \ldots, c_{n-1}), (f_0, \ldots, f_{n-1})) = \mathsf{Enc}(param, mkey, num)$, and $(I', (c'_0, \ldots, c'_{n-1}), (f'_0, \ldots, f'_{n-1})) = \mathsf{Enc}(param, mkey, num')$. Since d_i and d'_i depend only on $\{b_j\}_{j=i+1,\ldots,n-1}$ and $\{b'_j\}_{j=i+1,\ldots,n-1}$ respectively and on $mkey$, that $b_i = b'_i$ holds for $i = \ell+1, \ldots, n-1$ implies that $d_i = d'_i$ holds for $i = \ell+1, \ldots, n-1$. Hence, $\mathsf{Hash}(d'_k, (2, I', 0)) = c'_k = \mathsf{Hash}(d_k, (2, I', 0))$ for $i = \ell+1, \ldots, n-1$.

If $num = num'$, $\mathsf{Hash}(d'_k, (2, I', 0)) = c'_k = \mathsf{Hash}(d_k, (2, I', 0))$ holds for $i = 0, \ldots, n-1$. Hence, the output of Cmp is 0 if $num = num'$. If $num \neq num'$, then $d_\ell = d'_\ell$ holds with negligible probability. This is because, if collision occurs with non-negligible probability for a function whose output length is κ, such a function can be distinguished from the random function by using collisions. Hence, $\mathsf{Hash}(d'_\ell, (2, I', 0)) = c'_\ell \neq \mathsf{Hash}(d_\ell, (2, I', 0))$ with overwhelming probability. For this ℓ,

$$\begin{aligned} e_\ell - e'_\ell &:= (f_\ell - \mathsf{Hash}(d_{\ell+1}, (5, I, 0))) - (f'_\ell - \mathsf{Hash}(d_{\ell+1}, (5, I', 0))) \bmod 3 \\ &= (\mathsf{Hash}(mkey, (4, d_{\ell+1}, 0)) + b_\ell) - \left(\mathsf{Hash}(mkey, (4, d'_{\ell+1}, 0)) + b'_\ell\right) \bmod 3 \\ &= (\mathsf{Hash}(mkey, (4, d_{\ell+1}, 0)) + b_\ell) - (\mathsf{Hash}(mkey, (4, d_{\ell+1}, 0)) + b'_\ell) \bmod 3 \\ &= b_\ell - b'_\ell \bmod 3 \end{aligned}$$

Since that $num > num'$ if $b_\ell = 1 > 0 = b'_\ell$ and that $num < num'$ if $b_\ell = 0 < 1 = b'_\ell$, the output of Cmp is 1 if $num > num'$ and is -1 if $num < num'$.

3.4 Efficiency

We compare complexity measures of our scheme with those of OPE. We list them when numbers num are chosen as $0 \leq num < 2^n$ in the Table 1. The dominant cost of computation is computation of hash functions in our scheme. Hence, we evaluate the computational complexity of our scheme by the number of hash function Hash. Encryption in OPE [8] requires sampling from negative hypergeometric distribution, which cost is denoted by "sampling". This requires rather high cost.

Table 1. Comparison

	Our Scheme	OPE[8]
ciphertext(text) length (bits)	$(n+1)\kappa + 2n$	$n +$ constant
token length (bits)	$(n+1)\kappa$	-
encryption cost	$(4n+1) \cdot \mathsf{Hash}$	$n \cdot$ sampling
token generation cost	$(n+1) \cdot \mathsf{Hash}$	-
comparison cost	$(n-B+2) \cdot \mathsf{Hash}$	$(n-B) \cdot$ bit-comparison

"bit-comparison" is very light computation and n bit-comparison operations is usually executed in one operation. B is the largest ℓ such that $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$ holds.

From the table, we see that OPE is more efficient except for generating ciphertexts. However, we consider that the cost our comparable encryption requires is still acceptable for most applications, and a comparable encryption is essential for data to which OPE cannot be applied securely.

4 Security Analysis

We analyze the security of our scheme. As our scheme is not ideal comparable encryption, we introduce a weaker security requirement of comparable encryption as well as the ideal one.

We require comparable encryption to be semantically secure under chosen plaintext attacks as long as no token is generated. When a token *token* is generated with respect to a number *num*, it is best if *token* only enables to compare this *num* with other encrypted numbers. To capture such a requirement, we start from defining a distinguishing game of comparable encryption. In this game, the adversary may send the challenger either of two types of test query, that is, type I and type II. This type indicates whether or not ciphertext in the test query is accompanied with the corresponding token. Then we define two notions of resolved games followed by two related definitions of indistinguishability of comparable encryption. The first notion captures ideal comparable encryption but the latter captures comparable encryption with an extra leakage of knowledge.

We chose game-based definition rather than simulation-based definitions (in [17,13,22]) because what each token leaks depends on all issued ciphertexts, which bothers ideal functionality to check all of them every time a token is issued. However, game-based definition requires to check if issued tokens have leaked something crucial only once at the end of the game.

4.1 Ideal Indistinguishability

Definition 3. *The* **distinguishing game** *is played between challenger C and adversary A^* as in the following. It begins when C receives a security parameter $\kappa \in \mathbb{N}$ and a range parameter $n \in \mathbb{N}$, runs $(\mathrm{param}, \mathrm{mkey}) \leftarrow \mathsf{KeyGen}(\kappa, n)$, and gives* param *to A^*. C responds to queries from A^* in the game as follows;*

- *Whenever C receives* $(\mathsf{encrypt}, num)$ *for any* $0 \leq num < 2^n$*, it returns* $ciph = \mathsf{Enc}(param, mkey, num)$.
- *Whenever C receives* $(\mathsf{cmprkey}, num)$ *for any* $0 \leq num < 2^n$*, it returns* $token = \mathsf{Der}(param, mkey, num)$.
- *C receives* $(\mathsf{test}, type, num_0^*, num_1^*)$ *such that* $0 \leq num_0^*, num_1^* < 2^n$*,* $num_0^* \leq num_1^*$*, and* $type \in \{I, II\}$ *only once in the game. On receiving this message, C randomly chooses* $b \in \{0, 1\}$ *and generates* $ciph^* = \mathsf{Enc}(param, mkey, num_b^*)$ *and* $token^* = \mathsf{Der}(param, mkey, num_b^*)$*. Then C returns*

$$\begin{array}{ll} ciph^* & \text{if} \quad type = I \\ token^*, ciph^* & \text{if} \quad type = II. \end{array}$$

At the end of the game, A sends $b' \in \{0, 1\}$ *to C. The result of the game* $\mathsf{Exp}_{C,A}^{\kappa}$ *is* 1 *if* $b = b'$*; otherwise* 0.

Type I tests indistinguishability of the encryption of num_b^*. Type II tests indistinguishability of the token with respect to num_b^*. We do not consider chosen-ciphertext attacks here since encrypt-then-MAC [7] generic construction can easily make the scheme resistant for them when an ordinary ciphertext is concatenated to each ciphertext so as to be decryptable.

The distinguishing game challenges the adversary's ability to distinguish ciphertexts. However, if a certain set of queries is sent to the challenger, it is inevitable to prevent rational adversaries from distinguishing these ciphertexts. This is because that tokens enable to compare encrypted numbers inevitably leaks their orders. Hence, the cases and only the cases when such a leakage trivially helps distinguishing ciphertexts/tokens need to be excluded from the games to measures the strength of the scheme. For this purpose we introduce the notion of resolved games.

Definition 4. *We say a distinguishing game is* **resolved** *if A^* queries such* $(\mathsf{command}, num)$ *that the following relation holds during the game, where* command *is* cmprkey *if* $type = I$ *but* command *is either* cmprkey *or* encrypt *if* $type = II$.

$$(num_0^* \leq num \leq num_1^*) \wedge (num_0^* \neq num_1^*)\,, \tag{1}$$

which relation can be equivalently expressed as

$$((num_0^* < num) \wedge (num_1^* \not< num)) \vee ((num \not< num_0^*) \wedge (num < num_1^*))\,.$$

The first form of the relation in Def. 4 represents that num is between num_0^* and num_1^* but the case $num_0^* = num = num_1^*$ is excluded. It is crystal clear that two test messages can be distinguishable if a token that can distinguish them is queried (type I). And it is also clear that two test tokens can be distinguishable if an message that these tokens decide in different way is encrypted (type II).

The second form of the relation in Def. 4 represents that num_0^* and num_1^* are related to num in different way via the relation "$<$". The first and the second forms are equivalent but the second form has more affinity with distinguishability, and we use the second type of form for Def. 6.

Definition 5. *We say that a comparable encryption is* **indistinguishable (Ind)** *if, for every polynomial time adversary* A^*, $\mathsf{Adv}^{\kappa}_{C,A^*} := |\Pr[\mathsf{Exp}^{\kappa}_{C,A^*} = 0] - \Pr[\mathsf{Exp}^{\kappa}_{C,A^*} = 1]|$ *is negligible with respect to* κ *in the game which is not resolved.*

We emphasize that num_0^* and num_1^* are always distinguishable in resolved games as long as the comparable encryption is complete. In other words, adversaries are not considered to be successful in distinguishing ciphertexts if and only if distinguishing them is trivially possible due to the functionality of the scheme.

4.2 Weak Indistinguishability

The indistinguishability in Def. 5 is ideal but the scheme we devised does not satisfy this property. However, the scheme partially achieves this property. Hence, we need to estimate what and how much it achieves. A token for *num* in our scheme leaks one bit for each ciphertext addition to that in an ideal scheme leaks. As we want estimate the relative impact of this leakage compared to the impact of what an ideal scheme leaks, we introduce a security notion that include this leakage in term of indistinguishability. For this purpose, we introduce weak indistinguishability.

We say $num <_\ell num'$ if $num < num'$, $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$, and $b_\ell \neq b'_\ell$ all hold. Note that "$num \not<_\ell num'$" (the negation of $num <_\ell num'$) holds for some ℓ even if $num < num'$. We will see how this notion works.

Suppose that $num < num' < num^\ddagger$ and $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$ and $\mathsf{MSPBs}(num', \ell') = \mathsf{MSPBs}(num^\ddagger, \ell')$ for $\ell < \ell'$. It is trivial that $token^\dagger = \mathsf{Der}(param, mkey, num^\dagger)$ and $ciph^\dagger = \mathsf{Enc}(param, mkey, num^\dagger)$ enable to distinguish $ciph = \mathsf{Enc}(param, mkey, num)$ and $ciph' = \mathsf{Enc}(param, mkey, num')$ if $num < num^\dagger < num'$. In our scheme, $token^\ddagger = \mathsf{Der}(param, mkey, num^\ddagger)$ and $ciph^\ddagger = \mathsf{Enc}(param, mkey, num^\ddagger)$ also enable to distinguish $ciph$ and $ciph'$. This is because as follows. $ciph$, $ciph^\ddagger$, and $token^d dagger$ reveal that ℓ-th bit of num and $num^\ddagger$ are different. $ciph'$, $ciph^\ddagger$, and $token^d dagger$ reveal that ℓ-th bit of num' and $num^\ddagger$ are the same. The notion "$<_\ell$" captures this property by $num <_\ell num^\ddagger$ and $num' \not<_\ell num^\ddagger$.

Definition 6. *We say a distinguishing game is* **weakly resolved** *if* A^* *queries such* $(\mathsf{command}, num)$ *that the following relation holds during the game, where* command *is* cmprkey *if* $type = I$ *but* command *is either* cmprkey *or* encrypt *if* $type = II$.

$$\exists \ell (0 \leq \ell < n) \text{ s.t.}$$
$$((num_0^* <_\ell num) \wedge (num_1^* \not<_\ell num)) \vee ((num \not<_\ell num_0^*) \wedge (num <_\ell num_1^*)) \quad (2)$$

Here, n *is the range parameter given to* C *at the beginning of the game.*

Note that Def. 4 and Def. 6 are different only in that "$\exists \ell$" is added and that $<$ is replaced with $<_\ell$. The Fig. 1 illustrates this difference between Def. 4 and Def. 6 in the case $num_0^* = 9$ and $num_1^* = 13$. The figure consists of nodes of a tree expressed by dots. The leftmost dot is the root and rightmost dots are

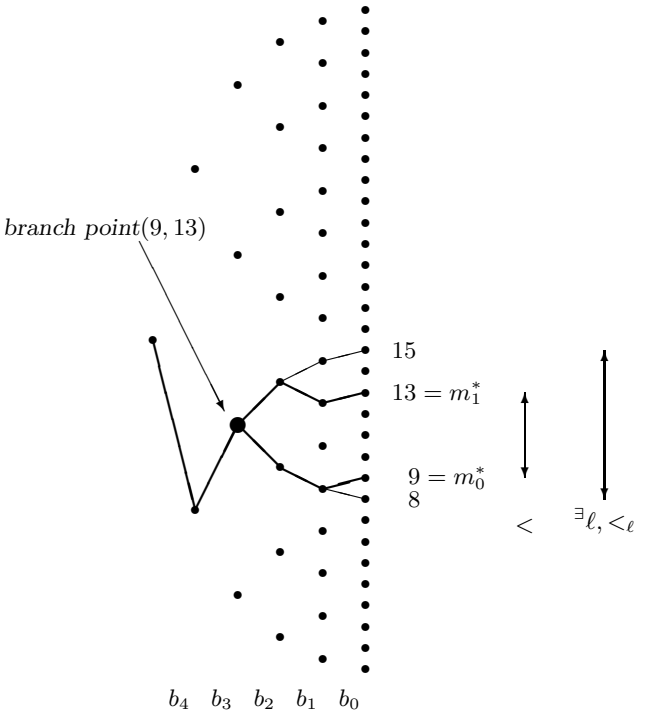

Fig. 1. Tree Representations of 9 and 13, and the ranges specified by "$<$" and "$\exists \ell, <_\ell$"

leaves. Other dots are internal nodes. Each path from the root to a leaf expresses a number in $[0, 2^5)$. Each path consists of five edge and each edge represents a bit. An upward edge represents 1 and downward one represents 0. Hence 13, which is $(b_4, b_3, b_2, b_1, b_0) = (0, 1, 1, 0, 1)$, is expressed as a path that advances from the root to a leaf by choosing directions (down,up,up,down,up) at nodes on the path.

In the case of Fig. 1, the game is resolved if $(\mathsf{command}, num)$ for $m_0^* = 9 \leq num \leq 13 = m_1^*$ is queried but the game is weakly resolved if $(\mathsf{command}, num)$ for $8 \leq num \leq 15$ is queried. Note that these numbers $8, 9, 13, 15$ share the same node pointed indicated by "*branch point*$(9, 13)$" in the figure. Here, 8 and 15 are the minimum and the maximum number that share the node where 9 and 13 branch away. Def. 6 forbids numbers in wider range to be queried so as the game to be not resolved than Def. 4 forbids. We consider how much this range is widened is how much schemes get weaker. In this example, the range $13 - 9 + 1 = 5$ is widened to $15 - 8 + 1 = 8$ by the ratio of $8/5 = 1.6$. We later argue that the expected value of this ratio is 2.8.

Definition 7. *We say that a comparable encryption is* **weakly indistinguishable (wInd-secure)** *if, for every polynomial time adversary* A^*, $\mathsf{Adv}^{\kappa}_{C,A^*} :=$

$|\Pr[\mathsf{Exp}^{\kappa}_{C,A^*} = 0] - \Pr[\mathsf{Exp}^{\kappa}_{C,A^*} = 1]|$ *is negligible with respect to* κ *in the game which is not weakly resolved.*

Since Def. 7 considers that the game is resolved under wider class of queries than Def. 5 does, it provides weaker security. But we consider this difference is in moderate extent. The impact of difference between Def. 7 and Def. 5 is analyzed in Subsection 4.3.

Theorem 2. *The proposed comparable encryption is* **weakly indistinguishable** *as long as* Hash *is a pseudorandom function.*

Proof. The proof is straightforward. We replace some of outputs of hash functions with random variable and then simply prove indistinguishability of them. The proof is given in Appendix A.

4.3 Comparison of Two Indistinguishability Notions

Although a comparable encryption that are only wInd-secure leaks more knowledge than ideal ones, ciphertexts in it reveal no knowledge without tokens. Hence, such a comparable encryption is still effective, unlike OPEs, even when encrypting numbers that are densely distributed in a table. But, as there is a chance for an adversary to obtain tokens, it is now essential to evaluate the amount of knowledge that these tokens leak.

From a simple observation, each token with respect to num leaks where num and num' branch away for each encryption of num'. This is a great amount of information if we insist on semantic security. But it is not clear in the context of such an encryption schemes that comparisons are already possible. Hence, we evaluate the how knowledge of these branching bits gives an impact in distinguishing numbers compared to the ideal comparable encryption. We do not consider ours is the only way to evaluate the impact and consider a lot of discussion is necessary. We hope our evaluation opens the problem.

Suppose that $0 \leq num_0^*, num_1^* < 2^n$ are given. Let $D(num_0^*, num_1^*)$ be the number of num that satisfies Eq. (1) and let $N(num_0^*, num_1^*)$ be the number of num that satisfies Eq. (2). Then $R(num_0^*, num_1^*) = N(num_0^*, num_1^*)/D(num_0^*, num_1^*)$ is the ratio of "the number of occasions when tokens of a weaker scheme leaks" to "the number of occasions when tokens of an ideal scheme leaks", which represents how much wInd-secure comparable encryption is weak compared to ideal comparable encryption. When the ratio is one, a wInd-secure comparable encryption has no worse than ideal comparable encryptions. But the ratio that is larger than one signifies the weakness of wInd-secure comparable encryption.

Since the ratio $R(num_0^*, num_1^*)$ varies over the choice of pair (num_0^*, num_1^*), the ratio at a single point cannot represents the total security of wInd-secure comparable encryptions. Hence, we evaluate its expected value over uniformly and randomly chosen (num_0^*, num_1^*) and consider it as a measure of the weakness of wInd-secure comparable encryptions. Although imposing uniform distribution is rather crude, we have no reasonable alternative choice.

Let $\ell(x,y)$ be largest ℓ such that $\mathsf{MSPBs}(x,\ell) = \mathsf{MSPBs}(y,\ell)$ holds. Then, the expected value of $R(num_0^*, num_1^*)$ is,

$$\frac{2}{2^n(2^n-1)} \sum_{0\le x<y<2^n} R(x,y) = \frac{2}{2^n(2^n-1)} \sum_{0\le x<y<2^n} \frac{2^{\ell(x,y)}-1}{y-x}$$

$$= \frac{2}{2^n(2^n-1)} \sum_{\ell=0}^{n-1} \sum_{\{x,y|\ell(x,y)=\ell\}} \frac{2^\ell-1}{y-x}$$

$$= \frac{2}{2^n(2^n-1)} \sum_{\ell=0}^{n-1} 2^{n-1-\ell} \sum_{0\le a,b<2^{\ell-1}} \frac{2^\ell-1}{a+b+1}$$

$$\lesssim \frac{2}{2^n(2^n-1)} \sum_{\ell=0}^{n-1} 2^{n+1+\ell} \frac{1}{2^{2(\ell-1)}} \int_{a=1}^{2^{\ell-1}} \int_{b=1}^{2^{\ell-1}} \frac{2^\ell}{a+b} dbda$$

$$\lesssim \frac{2}{2^n(2^n-1)} \sum_{\ell=0}^{n-1} 2^{n+1+\ell} \cdot 2\ln 2 = 4\ln 2 \lesssim 2.8$$

Therefore, we may conclude that, in average, the number of values that helps adversary distinguish num_0^* and num_1^* in wInd-secure comparable encryption is at most 2.8 times as large as that of values in ideal comparable encryptions. We consider this is not a considerable sacrifice for achieving practically efficiency of comparable encryption in most applications. This measure is based on rather crude assumption of the distribution but note that tokens are always deleted after their use.

As well as the expected ratio $N(num_0^*, num_1^*)/D(num_0^*, num_1^*)$, we give two more measures of comparison in Table 2. The expected value of $D(num_0^*, num_1^*)/N(num_0^*, num_1^*)$ is almost 1/2. The expected value of $N(num_0^*, num_1^*)$ divided by the expected value of $D(num_0^*, num_1^*)$ is at most 2. Although the interpretations of these measures are not as natural as that of the expected value of ratio $N(num_0^*, num_1^*)/D(num_0^*, num_1^*)$, they measures the security of wInd-secure schemes in some extent. Both measures indicate better security as they get closer to 1.

Table 2. Various comparison measures

Measures	value
Expected Value of "$N(num_0^*, num_1^*)/D(num_0^*, num_1^*)$"	≤ 2.8
Expected Value of "$D(num_0^*, num_1^*)/N(num_0^*, num_1^*)$"	≤ 2
"E.V. of $N(num_0^*, num_1^*)$"/ "E. V. of $D(num_0^*, num_1^*)$"	$\ge 1/2$

5 Summary and Open Problem

We introduced a novel type of encryption scheme called comparable encryption, which enables one to compare the numerical order of two encrypted numbers only when either of numbers is accompanied by a token. We presented an

ideal property and a weaker but reasonably nice property of comparable encryption. We also constructed a comparable encryption that satisfies only the weaker property but is practically efficient. We consider a comparable encryption is a useful primitive for encrypted DBs and consider proposing an efficient comparable encryption with the ideal property is a remaining important challenge. Our construction can be its starting point. By comparing efficiency of OPE and comparable encryption, we suggest to use an OPE in encrypted DBs when its positive result (shown by [9]) holds but suggest to use a comparable encryption when that positive result no longer holds.

References

1. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Order-preserving encryption for numeric data. In: Weikum, G., König, A.C., Deßloch, S. (eds.) SIGMOD Conference, pp. 563–574. ACM (2004)
2. Amanatidis, G., Boldyreva, A., O'Neill, A.: Provably-secure schemes for basic query support in outsourced databases. In: Barker, S., Ahn, G.-J. (eds.) Data and Applications Security 2007. LNCS, vol. 4602, pp. 14–30. Springer, Heidelberg (2007)
3. Belazzougui, D., Boldi, P., Pagh, R., Vigna, S.: Monotone minimal perfect hashing: searching a sorted table with $o(1)$ accesses. In: Mathieu, C. (ed.) SODA, pp. 785–794. SIAM (2009)
4. Bellare, M., Boldyreva, A., Knudsen, L.R., Namprempre, C.: Online ciphers and the hash-CBC construction. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 292–309. Springer, Heidelberg (2001)
5. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
6. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: Definitional equivalences and constructions without random oracles. In: Wagner (ed.) [33], pp. 360–378
7. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000)
8. Boldyreva, A., Chenette, N., Lee, Y., O'Neill, A.: Order-preserving symmetric encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 224–241. Springer, Heidelberg (2009)
9. Boldyreva, A., Chenette, N., O'Neill, A.: Order-preserving encryption revisited: Improved security analysis and alternative solutions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 578–595. Springer, Heidelberg (2011)
10. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner (ed.) [33], pp. 335–359
11. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
12. Ceselli, A., Damiani, E., di Vimercati, S.D.C., Jajodia, S., Paraboschi, S., Samarati, P.: Modeling and assessing inference exposure in encrypted databases. ACM Trans. Inf. Syst. Secur. 8(1), 119–152 (2005)

13. Chase, M., Kamara, S.: Structured encryption and controlled disclosure. IACR Cryptology ePrint Archive, 2011:10 (2011)
14. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: FOCS, pp. 41–50 (1995)
15. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. J. ACM 45(6), 965–981 (1998)
16. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM 13(6), 377–387 (1970)
17. Curtmola, R., Garay, J.A., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) ACM Conference on Computer and Communications Security, pp. 79–88. ACM (2006)
18. Ding, Y., Klein, K.: Model-driven application-level encryption for the privacy of e-health data. In: ARES, pp. 341–346. IEEE Computer Society (2010)
19. Goh, E.-J.: Secure indexes. Cryptology ePrint Archive, Report 2003/216 (2003), http://eprint.iacr.org/
20. Hacigümüs, H., Iyer, B.R., Li, C., Mehrotra, S.: Executing sql over encrypted data in the database-service-provider model. In: Franklin, M.J., Moon, B., Ailamaki, A. (eds.) SIGMOD Conference, pp. 216–227. ACM (2002)
21. Hacigümüs, H., Mehrotra, S., Iyer, B.R.: Providing database as a service. In: ICDE, pp. 21–38. IEEE Computer Society (2002)
22. Kamara, S., Papamanthou, C., Roeder, T.: Dynamic searchable symmetric encryption. In: Yu, T., Danezis, G., Gligor, V.D. (eds.) ACM Conference on Computer and Communications Security, pp. 965–976. ACM (2012)
23. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval. In: FOCS, pp. 364–373 (1997)
24. Li, J., Omiecinski, E.R.: Efficiency and security trade-off in supporting range queries on encrypted databases. In: Jajodia, S., Wijesekera, D. (eds.) Data and Applications Security 2005. LNCS, vol. 3654, pp. 69–83. Springer, Heidelberg (2005)
25. Liu, H., Wang, H., Chen, Y.: Ensuring data storage security against frequency-based attacks in wireless networks. In: Rajaraman, R., Moscibroda, T., Dunkels, A., Scaglione, A. (eds.) DCOSS 2010. LNCS, vol. 6131, pp. 201–215. Springer, Heidelberg (2010)
26. Lu, W., Varna, A.L., Wu, M.: Security analysis for privacy preserving search of multimedia. In: ICIP, pp. 2093–2096. IEEE (2010)
27. Oracle. Oracle database 11g, oracle advanced security, http://www.oracle.com/technology/global/jp/products/security/db_security/htdocs/aso.html
28. Popa, R.A., Li, F.H., Zeldovich, N.: An ideal-security protocol for order-preserving encoding. Cryptology ePrint Archive, Report 2013/129 (2013), http://eprint.iacr.org/
29. Popa, R.A., Redfield, C.M.S., Zeldovich, N., Balakrishnan, H.: Cryptdb: protecting confidentiality with encrypted query processing. In: Wobber, T., Druschel, P. (eds.) SOSP, pp. 85–100. ACM (2011)
30. Shi, E., Bethencourt, J., Chan, H.T.-H., Song, D.X., Perrig, A.: Multi-dimensional range query over encrypted data. In: IEEE Symposium on Security and Privacy, pp. 350–364. IEEE Computer Society (2007)
31. Tang, Q.: Privacy preserving mapping schemes supporting comparison (2010)
32. TPC-C. Transaction processing performance council, http://www.tpc.org/tpcc/
33. Wagner, D. (ed.): CRYPTO 2008. LNCS, vol. 5157. Springer, Heidelberg (2008)

34. Wang, C., Cao, N., Li, J., Ren, K., Lou, W.: Secure ranked keyword search over encrypted cloud data. In: ICDCS, pp. 253–262. IEEE Computer Society (2010)
35. Xu, J., Fan, J., Ammar, M.H., Moon, S.B.: Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In: ICNP, pp. 280–289. IEEE Computer Society (2002)

A Proof of Theorem 2

The proof is by contraposition. Suppose that there exists an adversary A^* such that $\mathsf{Adv}^{\kappa}_{C,A^*}$ is not negligible with respect to κ in the game which is not weakly resolved. Then, we show that Hash is distinguishable from the random function, which is against the assumption that they are pseudorandom function. In particular, we consider a sequence of games by challengers C, C_1, and C_2 and then prove the theorem by the hybrid argument. We let $\mathsf{Branch}(num, num')$ denote the largest ℓ such that $\mathsf{MSPBs}(num, \ell) = \mathsf{MSPBs}(num', \ell)$ holds.

Proof. From two lemmas 1 and 2 and the hybrid argument, $|\mathsf{Adv}^{\kappa}_{C,A^*} - \mathsf{Adv}^{\kappa}_{C_2,A^*}|$ is negligible in κ as long as Hash is a pseudorandom function. Since $\mathsf{Adv}^{\kappa}_{C_2,A^*} = 0$ from Lemma 3, $\mathsf{Adv}^{\kappa}_{C,A^*}$ is negligible in κ. Hence, the theorem is proved.

Definition 8. *Challenger C_1 is the same as the challenger C in Definition 3 except the following:*

- *At the beginning of the game, C_1 discards mkey.*
- *C_1 prepares a table and simulate hash function $\mathsf{Hash}(mkey, \cdot)$. That is, whenever C_1 generates $\mathtt{output} = \mathsf{Hash}(mkey, \mathtt{input})$ for some $\mathtt{input}$, C_1 let $\mathtt{output}$ be $\mathtt{output}'$ if an entry $(\mathtt{input}, \mathtt{output}')$ is in the table. Otherwise, C_1 randomly chooses $\mathtt{output} \in \{0,1\}^{\kappa}$ and writes $(\mathtt{input}, \mathtt{output})$ into the table.*

Note that $(d_i)_{i=0,\ldots,n}$ and $(e_i)_{i=0,\ldots,n}$ that C_1 outputs for every num is completely random.

Lemma 1. *Assume that Hash is a pseudorandom function. For every polynomial time A^*, $|\mathsf{Adv}^{\kappa}_{C_1,A^*} - \mathsf{Adv}^{\kappa}_{C,A^*}|$ is negligible in κ.*

Proof. Since $mkey$ is used for only input to hash functions and is never revealed to A^*, the lemma follows from the indistinguishability of pseudorandom function.

Definition 9. *Challenger C_2 is the same as the challenger C_1 except the following:*

- *Let $(\bar{d}_0, \ldots, \bar{d}_n)$ and $(\hat{d}_0, \ldots, \hat{d}_n)$ be*

$$(\bar{d}_0, \ldots, \bar{d}_n) = \mathsf{Der}(param, mkey, num_0^*)$$
$$(\hat{d}_0, \ldots, \hat{d}_n) = \mathsf{Der}(param, mkey, num_1^*).$$

  Note that $\bar{d}_i = \hat{d}_i$ for all i such that $\mathsf{Branch}(num_0^*, num_1^*) < i \leq n$
  C_2 prepares a table and simulate hash function $\mathsf{Hash}(\bar{d}_i, \cdot)$ and $\mathsf{Hash}(\hat{d}_i, \cdot)$ for all i such that $0 \leq i \leq \mathsf{Branch}(num_0^*, num_1^*)$. The simulation is as is the before.

Lemma 2. *Assume that* Hash *is a pseudorandom function. For every polynomial time* A^*, $|\mathsf{Adv}^{\kappa}_{C_2,A^*} - \mathsf{Adv}^{\kappa}_{C_1,A^*}|$ *is negligible in* κ.

Proof. Let $num_0^* = (\bar{b}_0, \ldots, \bar{b}_{n-1})$, $num_1^* = (\hat{b}_0, \ldots, \hat{b}_{n-1})$, and $B = \mathsf{Branch}(num_0^*, num_1^*)$. Then $\bar{b}_i = \hat{b}_i$ for all i such that $B < i \leq n$. Suppose that the adversary queries $(\mathsf{command}, num)$ for $num := (b_0, \ldots, b_{n-1})$. If $b_i = \bar{b}_i$ for all i such that $B < i < n$, then $\mathsf{Branch}(num_1^*, num) \leq B$. This implies that the distinguishing game is weakly resolved. Therefore, there exists i such that $b_i \neq \bar{b}_i$ and that $B < i < n$, as long as the distinguishing game is not weakly resolved.

- In the case when $type = I$, none of $\bar{d}_0, \ldots, \bar{d}_B, \hat{d}_0, \ldots, \hat{d}_B$ is revealed to the adversary. For such data to be revealed, all $\bar{d}_{B+1}, \ldots, \bar{d}_B$ needs to be revealed. But the existence of i such that $b_i \neq \bar{b}_i$ and that $B < i < n$ prevents it. Since, the values $\bar{d}_0, \ldots, \bar{d}_B, \hat{d}_0, \ldots, \hat{d}_B$ are randomly chosen and unrevealed, the hardness of distinguishing random values with outputs of $\mathsf{Hash}(\bar{d}_i, \cdot)$ and $\mathsf{Hash}(\hat{d}_i, \cdot)$ for all i such that $0 \leq i \leq \mathsf{Branch}(num_0^*, num_1^*) = B$ follows from the indistinguishability of pseudorandom function. This proves the lemma in the case $type = I$.
- In case when $type = II$, one of tuples $(\bar{d}_0, \ldots, \bar{d}_B)$ and $(\hat{d}_0, \ldots, \hat{d}_B)$ is given to the adversary depending on the value of b unlike the case when $type = I$. We assume $b = 0$ in the following without lose of generality. Then, $\bar{d}_0, \ldots, \bar{d}_B$ are given to A^* in this case. Unlike the case when $type = I$, $\mathsf{Hash}(mkey, (4, \bar{d}_{i+1}, 0))$ is used only for generating $\bar{e}_i := \mathsf{Hash}(mkey, (4, \bar{d}_{i+1}, 0)) + \bar{b}_i \bmod 3$ for $i = 0, \ldots, B-1$ in $ciph^*$. Hence, replacing $\mathsf{Hash}(\bar{d}_{i+1}, (5, I, 0))$ in $\bar{f}_i := \mathsf{Hash}(\bar{d}_{i+1}, (5, I, 0)) + \mathsf{Hash}(mkey, (4, \bar{d}_{i+1}, 0)) + \bar{b}_i \bmod 3$ with a random value for $i = 1, \ldots, B$ does not affect the distribution of $\bar{f}_i$. This is because the distribution of $\bar{f}_i$ for $i = 0, \ldots, B-1$ are already random. This proves the lemma in the case $type = II$.

Lemma 3. *For every polynomial time* A^*, $\mathsf{Adv}^{\kappa}_{C_2,A^*} = 0$.

Proof. The lemma follows from the fact that $ciph^*$ does not depend on b, which can be shown as follows. The difference in $ciph^*$ between num_0^* and num_1^* may occur only in (c_i, f_i) for $i = 0, \ldots, B$. Since each $\mathsf{Hash}(\bar{d}_i, \cdot)$ (we assume $b = 0$ w.l.g.) for $i = 0, \ldots, B$ is randomly chosen, every c_i for $i = 0, \ldots, B$ does not depend on b. Since each $\mathsf{Hash}(\bar{d}_i, \cdot)$ for $i = 0, \ldots, B$ is randomly chosen, every f_i for $i = 0, \ldots, B$ does not depend on b. Therefore, the lemma is proved.