


Article

New Order-Revealing Encryption with Shorter Ciphertexts

Kee Sung Kim 

School of Computer Software, Daegu Catholic University, Gyeongbuk 38430, Korea; kee21@cu.ac.kr

Received: 18 August 2020; Accepted: 21 September 2020; Published: 23 September 2020



Abstract: As data outsourcing services have been becoming common recently, developing skills to search over encrypted data has received a lot of attention. Order-revealing encryption (OREnc) enables performing a range of queries on encrypted data through a publicly computable function that outputs the ordering information of the underlying plaintexts. In 2016, Lewi et al. proposed an OREnc scheme that is more secure than the existing practical (stateless and non-interactive) schemes by constructing an ideally-secure OREnc scheme for small domains and a “domain-extension” scheme for obtaining the final OREnc scheme for large domains. They encoded a large message into small message blocks of equal size to apply them to their small-domain scheme, thus their resulting OREnc scheme reveals the index of the first differing message block. In this work, we introduce a new ideally-secure OREnc scheme for small domains with shorter ciphertexts. We also present an alternative message-block encoding technique. Combining the proposed constructions with the domain-extension scheme of Lewi et al., we can obtain a new large-domain OREnc scheme with shorter ciphertexts or with different leakage information, but longer ciphertexts.

Keywords: order-preserving encryption; order-revealing encryption; database encryption

1. Introduction

Database encryption has received increased attention recently because of the enormous amount of sensitive data stored in outsourcing cloud databases. One of the promising solutions to protect the confidentiality of sensitive data is to use encryption and perform query evaluation over encrypted data.

Order-Preserving Encryption. Property-preserving encryption, which preserves some property of plaintexts, enables performing query evaluation on ciphertexts. Among them, order-preserving encryption (OPEnc) [1–5], whose ciphertexts preserve the numerical ordering of their underlying plaintexts, has received a lot of attention as it can support efficient query operation on encrypted data such as sorting and range queries using the ordering information. In 2004, Agrawal et al. [1] first proposed the concept of OPEnc. Later, Boldyreva et al. [2] provided the security notions of OPEnc formally and showed that any immutable OPEnc schemes with ideal security must have the ciphertext length that grows exponentially in the plaintext length. Recently, some ideally-secure OPEnc schemes [3–5] whose ciphertexts reveal no additional information beyond the order of the underlying plaintexts have been proposed. However, these schemes are stateful and mutable, thus they require large communication and storage complexities.

Order-Revealing Encryption. Boneh et al. [6] introduced order-revealing encryption (OREnc), which can be viewed as a generalization of OPEnc. In the OREnc schemes, anyone can check the ordering information of the underlying plaintexts from ciphertexts through a publicly computable comparison function, thus the encrypted data are not constrained to any particular form. Their construction is the first stateless and non-interactive OREnc scheme that achieves the ideal security. However, their OREnc scheme relies on multilinear maps that require heavy computation and strong assumptions and suffer from security analysis [7], and thus are not efficiently implementable. As part of solving

this problem, Chenette et al. [8] presented the first efficiently-implementable OREnc scheme from pseudo-random functions. They also provided a novel security model of OREnc that precisely quantifies what information of the underlying plaintexts is leaked. Later, Lewi et al. [9] proposed a new OREnc scheme with reduced leakage as compared with the scheme of [8]. This result could be achieved by constructing an ideally-secure OREnc for polynomially-sized domains (OREncS) scheme and a “domain-extension” scheme for obtaining OREnc for exponentially-sized domains (OREncL) scheme. They encoded a large message into small message blocks of equal size to apply them to their OREncS scheme, thus their resulting OREncL scheme reveals the index of the first differing message block. The authors of [10] comprehensively analyzed and compared OP(R)Enc schemes described so far and provided their performance results.

Our Contribution. In this work, we begin by reviewing the constructions of [9] and then present a new ideally-secure OREncS scheme with shorter ciphertexts. Combining it with the domain-extension scheme of [9], we can obtain a new OREncL scheme with shorter ciphertexts under the same security level. We also present an alternative message-block encoding technique. In a similar way, we can also obtain a new OREncL scheme with a different security level, but the ciphertext length is getting longer. It is hard to claim that the resulting OREncL scheme is more secure than the scheme of [9]. However, these results provide a clue that there might exist more secure and efficient message-block encoding techniques.

2. Preliminaries

We write λ and $[n]$ as a security parameter and a set of integers $\{1, \dots, n\}$ where n is a positive integer, respectively. For any bit strings $x, y \in \{0, 1\}^*$, $x \parallel y$ means the concatenation of x and y . We write $x \leftarrow S$ to denote the sampling of a value x from the distribution S or a uniformly random sampling from the set S . Two distributions D_1 and D_2 are computationally indistinguishable if there is no efficient poly-time adversary to distinguish D_1 from D_2 , except with negligible probability. Similarly, if the statistical distance between D_1 and D_2 is negligible, we say that they are statistically indistinguishable. We now review the definition of a secure pseudo-random function F and a secure pseudo-random permutation π . A function $F: K \times X \rightarrow Y$ is a secure pseudo-random function if there is no polynomially-bounded adversary who can distinguish $F(k, \cdot)$, where $k \leftarrow K$ from a truly random function $f(\cdot)$ from X to Y except with negligible probability on arbitrary inputs chosen by the adversary. A secure pseudo-random permutation $\pi: K \times X \rightarrow X$ can be defined similarly as there is no polynomially-bounded adversary who can distinguish $\pi(k, \cdot)$, where $k \leftarrow K$ from a truly random permutation on X . All logarithms in this paper are to the base of 2.

2.1. Formal Notion of OREnc

An order-revealing encryption (OREnc) scheme Π consists of three probabilistic polynomial-time algorithms $\Pi = (\Pi.\text{Setup}, \Pi.\text{Encrypt}, \Pi.\text{Compare})$ satisfying the following properties on a well ordered domain D and range R .

- $\Pi.\text{Setup}(1^\lambda) \rightarrow \text{key}$: For a security parameter λ , this setup algorithm generates a secret key key .
- $\Pi.\text{Encrypt}(\text{key}, \text{msg}) \rightarrow \text{ctx}$: For a secret key key and a message $\text{msg} \in D$, this encryption algorithm generates a ciphertext $\text{ctx} \in R$.
- $\Pi.\text{Compare}(\text{ctx}_1, \text{ctx}_2) \rightarrow b$: On input two ciphertexts ctx_1 and ctx_2 , this comparison algorithm outputs a bit $b \in \{0, 1\}$ (here, $b = 1$ means $\text{msg}_1 < \text{msg}_2$).

Correctness. For a security parameter λ , a given OREnc scheme Π is correct if for $\text{key} \leftarrow \Pi.\text{Setup}(1^\lambda)$, and any messages $\text{msg}_1, \text{msg}_2 \in D$ ($\text{msg}_1 < \text{msg}_2$), $\Pi.\text{Compare}(\text{ctx}_1, \text{ctx}_2) = 1$ where $\text{ctx}_1 \leftarrow \Pi.\text{Encrypt}(\text{key}, \text{msg}_1)$ and $\text{ctx}_2 \leftarrow \Pi.\text{Encrypt}(\text{key}, \text{msg}_2)$.

2.2. Security of OREnc

In this section, we review a simulation-based OREnc security model of [8] that precisely quantifies what information of plaintexts is leaked by defining a leakage function. We denote an adversary and a simulator for some $q = \text{poly}(\lambda)$ by $A = (A_1, \dots, A_q)$ and $S = (S_0, \dots, S_q)$, respectively. Let $\Pi = (\Pi.\text{Setup}, \Pi.\text{Encrypt}, \Pi.\text{Compare})$ be an OREnc scheme and $L(\cdot)$ denotes a leakage function of Π . For a security parameter λ , the experiments $\text{REAL}_A^\Pi(\lambda)$ and $\text{SIM}_{A,S,L}^\Pi(\lambda)$ are defined as follows:

$\text{REAL}_A^\Pi(\lambda)$: <ol style="list-style-type: none"> 1. $\text{key} \leftarrow \Pi.\text{Setup}(1^\lambda)$ 2. $(\text{msg}_1, \text{state}_A) \leftarrow A_1(1^\lambda)$ 3. $\text{ctx}_1 \leftarrow \Pi.\text{Encrypt}(\text{key}, \text{msg}_1)$ 4. for $2 \leq i \leq q$: <ul style="list-style-type: none"> $(\text{msg}_i, \text{state}_A) \leftarrow A_i(\text{state}_A, \text{ctx}_1, \dots, \text{ctx}_{i-1})$ $\text{ctx}_i \leftarrow \Pi.\text{Encrypt}(\text{key}, \text{msg}_i)$ 5. output $(\text{ctx}_1, \dots, \text{ctx}_q)$ and state_A 	$\text{SIM}_{A,S,L}^\Pi(\lambda)$: <ol style="list-style-type: none"> 1. $\text{state}_S \leftarrow S_0(1^\lambda)$ 2. $(\text{msg}_1, \text{state}_A) \leftarrow A_1(1^\lambda)$ 3. $(\text{ctx}_1, \text{state}_S) \leftarrow S_1(\text{state}_S, L(\text{msg}_1))$ 4. for $2 \leq i \leq q$: <ul style="list-style-type: none"> $(\text{msg}_i, \text{state}_A) \leftarrow A_i(\text{state}_A, \text{ctx}_1, \dots, \text{ctx}_{i-1})$ $(\text{ctx}_i, \text{state}_S) \leftarrow S_i(\text{state}_S, L(\text{msg}_1, \dots, \text{msg}_i))$ 5. output $(\text{ctx}_1, \dots, \text{ctx}_q)$ and state_A
--	---

The given OREnc scheme Π is secure with leakage function $L(\cdot)$ if, for all polynomially-bounded adversaries A , there exists a simulator S such that the two distributions $\text{REAL}_A^\Pi(\lambda)$ and $\text{SIM}_{A,S,L}^\Pi(\lambda)$ are computationally indistinguishable. From the security notion, we say that Π is ideally-secure if the leakage function $L(\cdot)$ reveals only the relative order of the underlying plaintexts.

3. OREnc for Small Domains

In 2016, Lewi et al. [9] proposed a new OREnc scheme to solve the problem of [8] “revealing the index of the first bit that differs between two underlying plaintexts”. The starting point of their construction was presenting an ideally-secure OREncS scheme. Now, we review briefly their ideally-secure OREncS scheme. Let H , F , and π denote a hash function with an output space $\{0, 1, 2\}$, a secure pseudo-random function, and a fixed random permutation, respectively. The ciphertext ctx for a given message msg consists of the following two parts:

$$\text{ctx}_L = (F(\text{key}, \pi(\text{msg})), \pi(\text{msg})) \text{ and}$$

$$\text{ctx}_R = (r, v_1, \dots, v_N) \text{ where } v_i = \text{CMP}(\pi^{-1}(i), \text{msg}) + H(F(\text{key}, i), r) \bmod 3 \text{ and } r \leftarrow \{0, 1\}^\lambda.$$

Here, $\text{CMP}(\text{msg}_1, \text{msg}_2)$ outputs -1 if $\text{msg}_1 < \text{msg}_2$, 0 if $\text{msg}_1 = \text{msg}_2$, and 1 if $\text{msg}_1 > \text{msg}_2$. Let $\text{ctx}_L^1 = (a, b)$ and $\text{ctx}_R^2 = (r, v_1, \dots, v_N)$ denote the left encryption part of msg_1 and the right encryption part of msg_2 , respectively. Then, the $\Pi.\text{Compare}$ algorithm can obtain $\text{CMP}(\text{msg}_1, \text{msg}_2)$ by computing $v_b - H(a, r)$. The main idea of this construction is that ctx_L contains message information hidden by the pseudo-random permutation and ctx_R is the encryption of the all relative order information to each message. Thus, the ciphertext size should grow linearly with the size of the domain space. More specifically, the size of each ciphertext is $2\lambda + \lceil \log N \rceil + \lceil N \log 3 \rceil$ for a security parameter λ and a domain $[N]$.

3.1. Proposed OREncS Scheme

In this section, we propose a new ideally-secure OREncS scheme with shorter ciphertexts. The main idea of our construction is to reduce the length of ciphertexts by replacing a random value r of ctx_R by $F(\text{key}, \pi(\text{msg}))$ of ctx_L and by eliminating the $\pi(\text{msg})$ term of ctx_L using a new ciphertext form. For a fixed security parameter λ and a message space $[N]$, let $F: \{0, 1\}^\lambda \times [N] \rightarrow \{0, 1\}^\lambda$ be a secure pseudo-random function and $H: \{0, 1\}^\lambda \times \{0, 1\}^\lambda \rightarrow \{0, 1\}$ be a 1-bit output hash function modeled as a random oracle. In our scheme, $\text{CMP}(\text{msg}_1, \text{msg}_2)$ outputs 1 if $\text{msg}_1 \leq \text{msg}_2$, and 0 otherwise. As described in [9], the order relation can be clarified by combining of the two results $\text{CMP}(\text{msg}_1, \text{msg}_2)$ and $\text{CMP}(\text{msg}_2, \text{msg}_1)$. The details of our proposed OREncS scheme Π are defined as follows:

- $\Pi.\text{Setup}(1^\lambda) \rightarrow \text{sk}_{\text{key}}$: This setup algorithm draws a pseudo-random function key $k \leftarrow \{0, 1\}^\lambda$ and a random permutation π on $[N]$. Then, it outputs sk_{key} as (k, π) .
- $\Pi.\text{Encrypt}(\text{sk}_{\text{key}}, \text{msg}) \rightarrow \text{ctx}$: For each $i \in [N]$, a bit v_i can be computed as

$$\text{CMP}(\pi^{-1}(i), \text{msg}) \oplus H(F(k, \text{msg}), F(k, \pi^{-1}(i))) \text{ where } \pi^{-1}(i) \neq \text{msg}.$$

This encryption algorithm outputs $\text{ctx}_L = (F(k, \text{msg}), v_1, \dots, v_{\pi(\text{msg})-1})$ and $\text{ctx}_R = (v_{\pi(\text{msg})+1}, \dots, v_N)$ as a ciphertext ctx .

- $\Pi.\text{Compare}(\text{ctx}_1, \text{ctx}_2) \rightarrow b$: On input two ciphertexts $\text{ctx}_1 = ((a, v_1, \dots, v_{k_1-1}), (v_{k_1+1}, \dots, v_N))$ and $\text{ctx}_2 = ((a', v'_1, \dots, v'_{k_2-1}), (v'_{k_2+1}, \dots, v'_N))$, the result of $\text{CMP}(\text{msg}_1, \text{msg}_2)$ can be computed by $v'_{k_1} \oplus H(a', a)$. In a similar way, the result of $\text{CMP}(\text{msg}_2, \text{msg}_1)$ can be obtained.

3.2. Analysis of Proposed OREncS

Theorem 1. (Correctness) *The proposed OREncL Π defined in Section 3.1 is correct.*

Proof of Theorem 1. We assume that there exists a message pair $(\text{msg}_1, \text{msg}_2)$ such that the $\Pi.\text{Compare}(\text{ctx}_1, \text{ctx}_2)$ algorithm does not output 1 where $\text{msg}_1 < \text{msg}_2$. For two ciphertexts $\text{ctx}_1 = ((a, v_1, \dots, v_{k_1-1}), (v_{k_1+1}, \dots, v_N))$ and $\text{ctx}_2 = ((a', v'_1, \dots, v'_{k_2-1}), (v'_{k_2+1}, \dots, v'_N))$, v'_{k_0} is computed as $\text{CMP}(\text{msg}_1, \text{msg}_2) \oplus H(F(k, \text{msg}_2), F(k, \text{msg}_1))$ by the $\Pi.\text{Encrypt}$ algorithm. Thus, $\Pi.\text{Compare}(\text{ctx}_1, \text{ctx}_2)$ can recover $\text{CMP}(\text{msg}_1, \text{msg}_2)$ by the following equation correctly.

$$v'_{k_0} \oplus H(a', a) = \text{CMP}(\text{msg}_1, \text{msg}_2) \oplus H(F(k, \text{msg}_2), F(k, \text{msg}_1)) \oplus H(a', a) = \text{CMP}(\text{msg}_1, \text{msg}_2)$$

By the definition, $\text{CMP}(\text{msg}_1, \text{msg}_2)$ is defined as 1 for $\text{msg}_1 \leq \text{msg}_2$. In the identical way, we can prove that $\text{CMP}(\text{msg}_2, \text{msg}_1) = 0$ ($\text{msg}_2 > \text{msg}_1$) can also be recovered correctly from the ciphertexts. Therefore, $\Pi.\text{Compare}(\text{ctx}_1, \text{ctx}_2)$ must output 1, which is a contradiction of our assumption. \square

Efficiency. Table 1 shows some comparison results of our scheme and the OREncS scheme of [9]. The ciphertext of our OREncS consists of a λ -bit output of the pseudo-random function F and N encrypted order information bit, thus the length of the ciphertext is $\lambda + N$. Compared with [9], the ciphertext of our scheme does not need to maintain the λ -bit random value and the $\lceil \log N \rceil$ -bit permuted message information.

Table 1. Comparison of our order-revealing encryption for polynomially-sized domains (OREncS) scheme defined in Section 3.1 and the existing scheme of [9]. Note that this is the result when the same 1-bit output hash function is applied to their scheme.

OREncS	Bit Size of ctx	Security
[9]	$2\lambda + N + \lceil \log N \rceil$	Ideal
Ours (Section 3.1)	$\lambda + N$	Ideal

Theorem 2. (Security) *The proposed OREncS scheme Π defined in Section 3.1 is ideally-secure.*

Proof of Theorem 2. To show that our proposed OREncS scheme guarantees the ideal security, it should be shown that the ciphertexts indistinguishable from real can be simulated using only the ordering information of the underlying messages. More formally, we should prove that there exists a simulator $S = (S_0, \dots, S_q)$ such that two distributions $\text{REAL}_A^\Pi(\lambda)$ and $\text{SIM}_{A,S,L}^\Pi(\lambda)$ are computationally indistinguishable for some $q = \text{poly}(\lambda)$ and an adversary $A = (A_1, \dots, A_q)$ defined in the OREnc security experiment. \square

Simulator Modeling. On input of a security parameter λ , the following two tables are maintained to ensure this simulation consistency throughout the proof.

- The table H_T : ($\alpha \in \{0, 1\}^\lambda$, $\beta \in \{0, 1\}^\lambda$, $\gamma \in \{0, 1\}$) maintains the simulated input to output mappings of the random oracle.
- The table $F\pi_T$: ($a \in [q]$, $b \in \{0, 1\}^\lambda$, $c \in [N]$) maintains the simulated outputs of the pseudo-random function and the fixed random permutation.

The initial state $state_S$ of S_0 consists of the two empty tables ($H_T, F\pi_T$). For an i ($\in [q]$)-th message msg_i of encryption query, ctx_i can be returned if $msg_i = msg_j$ for some $j < i$. Without any loss of generality, only distinct queried messages are considered in the proof. We now describe how to simulate ctx_i responding to the i -th queried message msg_i using $state_S$ and the relative order information of $(msg_1, \dots, msg_{i-1})$.

- Let M be a set of $\{c_1, \dots, c_{i-1}\}$ where each c_i is a third component in the table $F\pi_T$. The simulator S_i first draws $c \leftarrow [N] \setminus M$ and $b \leftarrow \{0, 1\}^\lambda$, then stores a tuple (i, b, c) to the table $F\pi_T$. Here, this experiment is aborted if there already exists a tuple (b, \cdot, \cdot) or (\cdot, b, \cdot) in the table H_T .
- The simulator S_i samples $v_i \leftarrow \{0, 1\}$ where $i \neq c$ and outputs $((b, v_1, \dots, v_{c-1}), (v_{c+1}, \dots, v_N))$ as a ciphertext ctx_i .

Random Oracle Modeling. We now give a description of the random oracle H . On an i -th input (α, β) , and an output bit can be simulated as follows:

- If there already exists (α, β, \cdot) in the table H_T , then H returns the third component of (α, β, \cdot) .
- Otherwise, if there exist both $(a, b = \beta, c)$ and $(a', b' = \alpha, c')$ in the table $F\pi_T$, the simulator first checks $\text{CMP}(msg_a, msg_{a'})$ from the leakage function $L(\cdot)$ and then searches a' -th previous simulated ciphertext $ctx_{a'}$. Finally, it returns $\text{CMP}(msg_a, msg_{a'}) \oplus v'_c$ as a hash output, where v'_c is an encrypted bit component in $ctx_{a'}$, and stores $(\alpha, \beta, \text{CMP}(msg_a, msg_{a'}) \oplus v'_c)$ in the table H_T .
- Otherwise, the simulator returns γ where $\gamma \leftarrow \{0, 1\}$, then stores (α, β, γ) in the table H_T .

Indistinguishability. To complete our security proof, we now show that two distributions $\text{REAL}^\Pi_A(\lambda)$ and $\text{SIM}^\Pi_{A,S,L}(\lambda)$ are computationally indistinguishable by defining a series of the below hybrid games:

- Game G_0 : This game is $\text{REAL}^\Pi_A(\lambda)$.
- Game G_1 : Same as G_0 , except the pseudo-random function F is switched by a truly random function $f: [N] \rightarrow \{0, 1\}^\lambda$.
- Game G_2 : Same as G_1 , except the game aborts if the adversary queries $(f(msg), \cdot)$ or $(\cdot, f(msg))$ to the random oracle H before simulating the ciphertext of msg .
- Game G_3 : This game is $\text{SIM}^\Pi_{A,S,L}(\lambda)$.

Lemma 1. Game G_0 and G_1 are computationally indistinguishable if F is a secure pseudo-random function.

Proof of Lemma 1. It is trivial from the definition of the secure pseudo-random functions. \square

Lemma 2. Game G_1 and G_2 are statistically indistinguishable if H is a random oracle.

Proof of Lemma 2. To prove lemma 2, we should show that the abort probability of Game G_2 is negligible. We clearly know that all components of the returned ciphertexts are distributed independently from $f(\pi(msg))$ before issuing a message msg to the encryption query input. Because $f(\cdot)$ is a truly random function, the probability that the adversary queries $(f(msg), \cdot)$ or $(\cdot, f(msg))$ to the random oracle H before simulating the ciphertext of the message msg is at most $\text{poly}(\lambda)/2^\lambda$. \square

Lemma 3. Game G_2 and G_3 are statistically indistinguishable if H is a random oracle.

Proof of Lemma 3. Let $((a, v_1, \dots, v_{k_1-1}), (v_{k_1+1}, \dots, v_N))$ and $((a', v'_1, \dots, v'_{k_2-1}), (v'_{k_2+1}, \dots, v'_N))$ be the ciphertexts from G_2 and G_3 . We now show these two distributions are statistically indistinguishable and the ciphertext under G_3 is valid. The value a is an output of a random function f and a' is uniformly sampled from $\{0, 1\}^\lambda$, thus they are statistically indistinguishable by the definition of the random functions. A bit v_i is computed as $\text{CMP}(\pi^{-1}(i), \text{msg}) \oplus H(f(\text{msg}), f(\pi^{-1}(i)))$ in Game 2 and the output of H is uniformly random on $\{0, 1\}$, thus each v_i is also distributed uniformly in $\{0, 1\}$. That is, v_i and v'_i are statistically indistinguishable unless $H(f(\text{msg}), \cdot)$ or $H(\cdot, f(\text{msg}))$ is revealed to the adversary before simulating the ciphertext of msg , but this will never happen by the definition of Game 2 and Game 3. Finally, the bit positions k_1 and k_2 are also statistically indistinguishable because they are the outputs of the random permutation. We now show that the simulated ciphertext in Game 3 is correct. Let $((a, v_1, \dots, v_{k_1-1}), (v_{k_1+1}, \dots, v_N))$ of msg_1 and $((a', v'_1, \dots, v'_{k_2-1}), (v'_{k_2+1}, \dots, v'_N))$ of msg_2 be the two simulated ciphertexts from Game G_3 . From the definition of the simulation, $H(a', a)$ is defined $\text{CMP}(\text{msg}_1, \text{msg}_2) \oplus v'_{k_1}$ by the random oracle modeling, thus the $\Pi.\text{Compare}$ algorithm can obtain the correct result as follows:

$$v'_{k_1} \oplus H(a', a) = v'_{k_1} \oplus \text{CMP}(\text{msg}_1, \text{msg}_2) \oplus v'_{k_1} = \text{CMP}(\text{msg}_1, \text{msg}_2)$$

Combining lemmas 1–3, we conclude that our proposed OREncS scheme is ideally-secure. \square

4. Alternative Message-Block Encoding Technique

The domain-extension algorithm of [9] is quite straightforward. At a high level, when message msg is represented in $x_1 \parallel x_2 \parallel \dots \parallel x_n$ as the d -ary strings, the corresponding ciphertext can be constructed as $\text{ctx}_1 \parallel \text{ctx}_2 \parallel \dots \parallel \text{ctx}_n$, where each ctx_i is an encryption of x_i by OREncS with a domain size d . One thing to note is that a pseudo-random permutation is applied (not a fixed random permutation) and the key part in the pseudo-random permutation is derived from the prefix of each block x_i to reveal only the index of the first block that differs between two plaintexts. Actually, the construction of [8] can be seen as taking an ideally secure OREnc scheme for 1-bit domains and extending it to the OREnc scheme for n -bit domains. The authors of [9] applied this general extension technique to their OREncS scheme. The ciphertext consists essentially of n ciphertexts of the OREncS scheme with domain size d , thus the total ciphertext size on domain size $N \leq d^n$ is $(n+1)\lambda + n(\lceil \log d \rceil + \lceil d \log 3 \rceil)$. Interested readers should refer to the paper [8,9] for more details.

4.1. Proposed OREncL Scheme

In this section, we introduce a new message-block encoding technique to construct a new OREncL scheme from our proposed OREncS scheme. We first show how to divide an exponential-size message into polynomial-size blocks to use them as inputs of our proposed scheme. In the construction of [9], it caused the “revealing the index of the first differing block” problem, because a message is divided into small message blocks of equal size. This means that an adversary can infer the approximate distance between the underlying two messages and a message block can be recovered if he obtains d ciphertexts that have the same leakage information. To alleviate these problems, we provide an alternative message-block encoding technique. The main idea of our scheme is that a message is divided by the position and the size of consecutive 1’s. Let $1(i, j)$ denote the size of j consecutive 1’s starting from the i -th bit position. Here, the index i is counted from the least significant bit. For example, a message 011101, 011100, 111,111 can be represented as $\{1(5, 3), 1(1, 1), 1(0, 0)\}$, $\{1(5, 3), 1(0, 0), 1(0, 0)\}$ and $\{1(6, 6), 1(0, 0), 1(0, 0)\}$. We can get the ordering information to check the same level component. In our example, we can know $011,101 > 011,100$ from $(5 = 5, 3 = 3, 1 > 0)$. To hide the exact number of $1(i, j)$, we use $1(0, 0)$ padding. Note that 3 is the largest possible element number for a 6-bit message space. More formally, an (even) n -bit message containing $\{1(i_1, j_1), 1(i_2, j_2), \dots, 1(i_k, j_k)\}$ can be encoded as $\{i_1, j_1, i_2, j_2, \dots, i_{n/2}, j_{n/2}\}$, where the elements of $i_{k+1}, j_{k+1}, \dots, i_{n/2}, j_{n/2}$ are 0. Our final ciphertext of a message encoded as $\{i_1, j_1, i_2, j_2, \dots, i_{n/2}, j_{n/2}\}$ can be computed as follows:

- For a given message msg , we first encode it as $\{i_1, j_1, i_2, j_2, \dots, i_{n/2}, j_{n/2}\}$ by our proposed technique.
- We generate ciphertext ctx_1, \dots, ctx_n by n -size domain OREnc scheme for each element in $\{i_1, j_1, i_2, j_2, \dots, i_{n/2}, j_{n/2}\}$.
- By applying the domain-extension algorithm of [8,9] to (ctx_1, \dots, ctx_n) , we can obtain our final ciphertext.

4.2. Analysis of Proposed OREncL

Because it is essentially identical to the OREncL scheme of [9], except for the way of generating message blocks, presenting the concrete description of our full OREncL scheme and the details of security proof is not necessarily required. The leakage information “CP of $1(i, j)$ ’s” of our scheme can be defined as the common prefix $\{1(i_1, j_1), \dots, 1(i_{k-1}, j_{k-1})\}$ of the underlying two messages where k is the index of the first $1(i, j)$ that differs. The equality information of i_k and j_k is also revealed. Compared with the leakage information of “revealing the index of the first differing block”, it is difficult to determine which leakage information is more critical, thus we thought that it could be another alternative option. Furthermore, this result provides a clue that there might exist more secure and efficient message-block encoding techniques.

In this analysis chapter, we present the result of the efficiency analysis. The following theorem shows that our message-block encoding technique preserves the order of messages correctly.

Theorem 3. *Our proposed message-block encoding technique preserves ordering information correctly.*

Proof of Theorem 3. First of all, every between $1(i, j)$ and $1(i', j')$ requires at least 1-bit 0, thus there is no message that contains more than $k-1(i, j)$ blocks where $k > n/2$. For any two messages msg_1 encoded as $\{i_1, j_1, i_2, j_2, \dots, i_{n/2}, j_{n/2}\}$ and msg_2 encoded as $\{i'_1, j'_1, i'_2, j'_2, \dots, i'_{n/2}, j'_{n/2}\}$, where $msg_1 < msg_2$, assume that i is the index of the first bit that differs, that is, the i -th bit of msg_1 is 0 and of msg_2 is 1.

- $(i-1)$ -th bit is 0: Let $\{i_1, j_1, \dots, i_k, j_k\}$ be a common part of the encoding of msg_1 and msg_2 . Because i -th bit of msg_1 is 0 and of msg_2 is 1, we conclude $i_{k+1} < i'_{k+1}$.
- $(i-1)$ -th bit is 1: Let $\{i_1, j_1, \dots, i_k, j_k\}$ be a common part of the encoding of msg_1 and msg_2 . Similar to the above case, because i -th bit of msg_1 is 0 and of msg_2 is 1, we conclude $i_{k+1} = i'_{k+1}$ and $j_{k+1} < j'_{k+1}$.

□

Efficiency. Table 2 shows some comparison results our OREncL schemes and the scheme of [9]. Ours I and II denote the OREncL schemes with our OREncS under the normal d -size message-block encoding of [9] and the proposed message-block encoding, respectively. Because a ciphertext of Ours I consists essentially of n ciphertexts of our OREncS whose ctx size is $\lambda + d$, as described in Section 3.2, the size of the resulting ciphertext is $n(\lambda + d)$. In the case of Ours II, the size of the resulting ciphertexts is $n \lceil \log d \rceil (\lambda + n \lceil \log d \rceil)$ because the message can be represented in $n \lceil \log d \rceil$ bits, and thus $n \lceil \log d \rceil$ ciphertexts of $n \lceil \log d \rceil$ -size domain OREncS are required. Taking $\lceil \log d \rceil = d/n$, the size of ciphertexts with our proposed message-block encoding is asymptotically longer by a multiplicative factor $\Omega(\log d)$ compared with the existing d -bit message-block encoding of [9].

Table 2. Comparison of our ORE for exponentially-sized domains (OREncL) and the existing scheme.

OREncL	Bit Size of ctx	Leakage
[9]	$n(\lambda + d) + \lambda + \lceil \log d \rceil$	Ideal
Ours I	$n(\lambda + d)$	First block that differs
Ours II	$n \lceil \log d \rceil (\lambda + n \lceil \log d \rceil)$	CP of $1(i, j)$ ’s

Simple Implementation. Figure 1 shows the percentage of the requiring ciphertext information until obtaining the II. Compare algorithm output for any two ciphertexts of Ours I and Ours II on two different domain sizes. For example, 98.76% of the ciphertexts of Ours II ($N = 2^{16}$) require only 25% of their ciphertext information to check the relative order on average. This result means the relative order information of two messages can be derived with slightly less ciphertext information when applying our proposed message-block encoding technique. However, because the ciphertext size of Ours II is longer, it does not mean our proposed encoding technique can guarantee a more efficient search time.

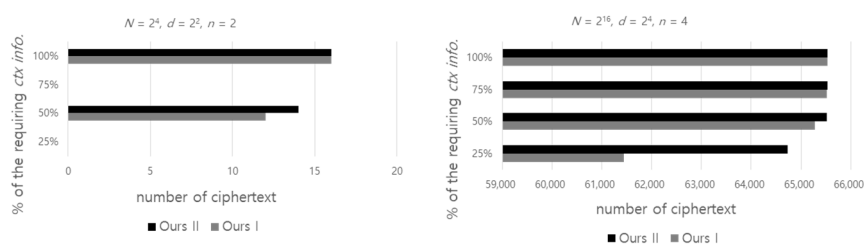


Figure 1. Simple implementation results of our order-revealing encryption for exponentially-sized domains (OREncL) schemes.

5. Conclusions

In this work, we introduced a new ideally-secure OREncS scheme with shorter ciphertexts compared with the existing scheme. We also presented an alternative message-block encoding technique for extending our OREncS to large-domains. Combining the proposed constructions with the existing Lewi et al.'s “domain-extension” scheme, we could obtain a new OREncL scheme with shorter ciphertexts whose security is the same as the existing scheme and a new OREncL scheme with longer ciphertexts whose leakage is the information of the common prefix consecutive 1' before the first differing bit. Moreover, we gave the efficiency and security analysis of our proposed schemes as well as a simple implementation result.

Funding: This work was supported by research grants from Daegu Catholic University in 2020.

Conflicts of Interest: The author declares that he has no conflict of interest regarding the publication of this paper.

References

1. Agrawal, R.; Kiernan, J.; Srikant, R.; Xu, Y. Order preserving encryption for numeric data. In Proceedings of the ACM SIGMOD International Conference on Management of Data ACM, SIGMOD '04, New York, NY, USA, 13–18 June 2004; pp. 563–574. [\[CrossRef\]](#)
2. Boldyreva, A.; Chenette, N.; Lee, Y.; O'Neill, A. Order-Preserving Symmetric Encryption. In *Advances in Cryptology—EUROCRYPT 2009*; Joux, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 224–241.
3. Popa, R.A.; Li, F.H.; Zeldovich, N. An ideal-security protocol for order-preserving encoding. In Proceedings of the IEEE Symposium on Security and Privacy, IEEE Computer Society, SP '13, Washington, DC, USA, 19–22 May 2013; pp. 463–477. [\[CrossRef\]](#)
4. Kerschbaum, F.; Schroepfer, A. Optimal average-complexity ideal-security order-preserving encryption. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, ACM CCS '14, New York, NY, USA, 3–7 November 2014; pp. 275–286. [\[CrossRef\]](#)
5. Kerschbaum, F. Frequency-Hiding Order-preserving encryption. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, CCS '15, New York, NY, USA, 12–16 October 2015; pp. 656–667. [\[CrossRef\]](#)
6. Boneh, D.; Lewi, K.; Raykova, M.; Sahai, A.; Zhandry, M.; Zimmerman, J. Semantically Secure Order-Revealing Encryption: Multi-input Functional Encryption without Obfuscation. Available online: <https://eprint.iacr.org/2014/834.pdf> (accessed on 23 September 2020).

7. Miles, E.; Sahai, A.; Zhandry, M. Annihilation Attacks for Multilinear Maps: Cryptanalysis of Indistinguishability Obfuscation over GGH13. Available online: <https://eprint.iacr.org/2016/147.pdf> (accessed on 23 September 2020).
8. Chenette, N.; Lewi, K.; Weis, S.A.; Wu, D.J. Practical Order-Revealing Encryption with Limited Leakage. In *Fast Software Encryption*; LNCS 9783; Springer: Berlin/Heidelberg, Germany, 2016; pp. 474–493.
9. Lewi, K.; Wu, D.J. Order-revealing encryption: New constructions, applications, and lower bounds. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, ACM, CCS '16, New York, NY, USA, 24–28 October 2016; pp. 1167–1178. [[CrossRef](#)]
10. Bogatov, D.; Kollios, G.; Reyzin, L. A comparative evaluation of order-revealing encryption schemes and secure range-query protocols. *Proc. VLDB Endow.* **2019**, *12*, 8. [[CrossRef](#)]



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).