Cybersecurity

RESEARCH

Open Access

Research on privacy information retrieval model based on hybrid homomorphic encryption

Wei-tao Song[1,2,3*], Guang Zeng[4], Wen-zheng Zhang[1] and Dian-hua Tang[1]

Abstract

The computational complexity of privacy information retrieval protocols is often linearly related to database size. When the database size is large, the efficiency of privacy information retrieval protocols is relatively low. This paper designs an effective privacy information retrieval model based on hybrid fully homomorphic encryption. The assignment method is cleverly used to replace a large number of homomorphic encryption operations. At the same time, the multiplicative homomorphic encryption scheme is first used to deal with the large-scale serialization in the search, and then the fully homomorphic encryption scheme is used to deal with the remaining simple operations. The depth of operations supported by the fully homomorphic scheme no longer depends on the size of the database, but only needs to support the single homomorphic encryption scheme to decrypt the circuit depth. Based on this hybrid homomorphic encryption retrieval model, the efficiency of homomorphic privacy information retrieval model can be greatly improved.

Keywords Cryptography, Hybrid homomorphic encryption, Privacy protection, Private information retrieval

Introduction

Fully homomorphic encryption (FHE) comes from the concept "privacy homomorphism", It was first proposed by Rivest et al. (1978). FHE refers to the ability of the operator to perform various operations on dense data without decrypting it, and the result is the same as that of corresponding operations on the plaintext after decryption, which really fundamentally solves the security problem when the data and operations are entrusted to the third party. So that people can not only make full use of the powerful computing/storage capacity of cloud computing to provide users with mass ciphertext processing services, but also manage their own keys to ensure data security, implementing "secure computing of data in untrusted environment (service)" (Rout et al. 2022; Akbar et al. 2023). At the same time, most FHE schemes are based on lattice difficult problems, and lattice cryptography is an important part of anti-quantum cryptography, so FHE is also one of the components of post-quantum cryptography (PQC) (Mosca 2014). Therefore, FHE password has gradually become a "strategic commanding point" in the field of cryptography contested by European and American countries, which can play an important role in the new service modes such as big data and cloud computing. (Sinha et al. 2023; Gautam and Shivhare 2022).

The concept of "private information retrieval (PIR)" was first proposed by Chor et al. (1995). The concept was proposed to solve this problem: the user can complete the query application to the database server on the premise

*Correspondence:
Wei-tao Song
weitaosong@163.com
[1] Science and Technology on Communication Security Laboratory, Chengdu 610041, China
[2] Zhejiang University, Hangzhou 311200, China
[3] PLA SSF Information Engineering University, Zhengzhou 450000, China
[4] College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China

that the query information is not leaked, that is, during the whole query process, the database server cannot obtain the relevant information of the user query statement and the specific information of the retrieval project. Among them, the communication complexity and computational complexity are two important indicators to evaluate the performance of the PIR protocol.

Privacy information retrieval plays a very important role in privacy outsourcing storage and computing. It means that when users retrieve information on the database, they should use certain methods to prevent database server managers from knowing the relevant information of query statements and the specific information of items to be retrieved, so as to protect users' query privacy. In real life, such as patent database, medical database, online census, real-time stock quotes and address location services, which have high requirements for search privacy, have a large application space. However, with the increase of the amount of data in the cloud, how to quickly and accurately retrieve the data needed by users from the massive ciphertext data in the cloud without disclosing users' privacy will be an urgent problem to be solved.

Chor et al. (1995) proved that when a database is used to realize the retrieval of absolute privacy information, the communication complexity is very high, reaching $\Omega(n)$, where n is the data scale of the database. This cost is far higher than the actual application requirements. At the same time, they also give a communication cost optimization scheme based on multi-server. The privacy query is completed through the common protocol of $k(k > 2)$ non-communicating database copy, which reduces the communication complexity to $O(n^{1/\log k})$. In 1997, Ambainis constructed a multi-server PIR protocol with communication complexity of $O(n^{1/(2k-1)})(k > 2)$ (Ambainis 1997). Since then, there have been various improvement schemes (Beimel and Ishai 2001; Itoh 1999; Ishai and Kushilevitz 1999). But they have little improvement on the communication complexity.

The above PIR research method implemented through multiple non-communicating database copies is referred to as information theory-based privacy information retrieval (IPIR). In 1997, Chor and Gilboa (1997) first proposed PIR (CPIR) model under computational security. In this model, the privacy requirements for users are relaxed, and the server is required to be unable to know the privacy of users' queries in polynomial time, and a specific CPIR protocol scheme is given, and the communication complexity is $O(n^{\varepsilon})$ (ε is an arbitrary constant greater than 0). But the solution requires two copies of the database. Subsequently, Kushilevitz et al. (1997) pointed out that database copy is not necessary. Based on the quadratic residual hypothesis problem, they constructed a single-server CPIR protocol with the communication complexity of $O(n^{\varepsilon})$ (ε is an arbitrary constant greater than 0). Since then, CPIR protocol schemes based on hiding hypothesis, discrete logarithm and single trap gate permutation have appeared but most of their communication complexity is $O(n^{\varepsilon})$ (ε is an arbitrary constant greater than 0), but the difference is the computational complexity (Cachin et al. 1999; Kushilevitz and Ostrovsky 2000; Wang et al. 2010).

The emergence of FHE scheme provides a new method to construct CPIR protocol, and through FHE scheme, the communication complexity can be reduced to $O(log\ n)$, which is a great improvement. Specifically, in 2009, Gentry proposed the first FHE scheme, and based on the FHE scheme, roughly proposed a sublinear communication complexity CPIR scheme (Gentry 2009). In 2011, Brakerdski and Vaikuntanathan (2011a) proposed the LWE-based FHE scheme for the first time. Compared with Gentry's scheme, the ciphertext scale is smaller. Taking advantage of this feature, they constructed a CPIR scheme with communication complexity of $\lambda \cdot \mathrm{poly}\log(\lambda) + \log n$, in which λ is the security parameter (Brakerski and Vaikuntanathan 2011b). In 2013, Xun et al. proposed a general, simple and efficient CPIR construction scheme based on FHE in literature (Yi et al. 2013). The CPIR example of FHE scheme based on integer AGCD problem is given, and the communication complexity of the scheme is $O(log\ n)$. Up to now, most of the existing FHE-based CPIR schemes follow the construction framework of Xun et al., who focus on the optimization of the FHE scheme itself, and then obtain a CPIR scheme with good performance by selecting appropriate parameter settings based on the optimized FHE scheme (Ichibane et al. 2015; Eltarjaman and Annadata 2016). For example, Doroz constructed a CPIR scheme with low communication complexity based on NTRU's SWHE scheme in 2014, but the computational complexity of their scheme is particularly high (Doröz et al. 2014). In 2016, Melchor et al. further optimized the efficiency of CPIR scheme by using fast FFT transformation and batch processing technology based on the FHE of ring LWE (Aguilar-Melchor et al. 2016). In 2017, Li et al. further optimized the efficiency of CPIR scheme by using the GSW batch processing method proposed by HAO15 scheme (Li et al. 2017). In 2018, Angel used plaintext multiplication to avoid complex ciphertext multiplication, but caused a larger amount of downloaded data (Angel et al. 2018). In 2019, Gentry et al. proposed the CPIR scheme based on the FHE scheme whose compressed plain-ciphertext expansion rate is close to 1 (Gentry and Halevi 2019). In 2021, Mughees et al. gave a variant of Xun et al.'s scheme by using the method of linear growth of external

homomorphic multiplicative noise (Mughees et al. 2021); In 2022, Menon et al. optimized Mughees et al.'s scheme using a homomorphic matrix version with high compression rates (Menon and Wu 2022).

In summary, when the size of the database is relatively large, a high-dimensional database storage structure is required. Fully homomorphic ciphertext multiplication takes up a lot of overhead on the server. Based on the advantages of multiplicative homomorphic encryption scheme in dealing with ciphertext multiplication, this paper studies an efficient hybrid homomorphic encryption privacy information retrieval mode.

Contributions

- The assignment method is cleverly used to replace a large number of homomorphic encryption operations. In the generation stage of YKPB-PIR protocol model response algorithm, database indexes need to be traversed, and FHE encryption algorithm is run on each index for bit-by-bit encryption. In fact, the encryption operation link is not necessary, can be based on the customer query message through the assignment to replace, which greatly reduces the number of homomorphic encryption and homomorphic addition number of operations.
- An effective privacy information retrieval model for large-scale database based on hybrid fully homomorphic encryption is given. The multiplicative homomorphic encryption scheme is first used to deal with the large-scale serialization in the search, and then the fully homomorphic encryption scheme is used to deal with the remaining simple operations. The depth of operations supported by the fully homomorphic scheme no longer depends on the size of the database, but only needs to support the single homomorphic encryption scheme to decrypt the circuit depth. By this way, the efficiency of homomorphic privacy information retrieval model can be greatly improved.

Preliminaries

Fully homomorphic encryption

A fully homomorphic encryption scheme can be described as a 4-tuple of algorithms FHE = (FHE.Keygen, FHE.Enc, FHE.Dec, FHE.Eval) as follows.

- **FHE.Keygen**(1^{λ}): Input security parameterλ, compute and output $(sk, pk, evk) \leftarrow$ **FHE.Keygen**(1^{λ}), where sk is the private key, pk is the public key, and evk is the homomorphic evaluation key.
- **FHE.Enc**(μ, pk): Input plaintext μ and public key pk, compute and output ciphertext $c \leftarrow$ **FHE.Enc**(μ, pk) .
- **FHE.Dec**(c, sk):Input private key sk and ciphertext c, compute and output plaintext $\mu \leftarrow$ **FHE.Dec**(c, sk).
- **FHE.Eval**$(f, evk, c_1, \ldots c_\ell)$: Enter the homomorphic arithmetic key evk, a set of ciphertext $c_1, \ldots, c_\ell$ and homomorphic operation function f, computes and outputs a ciphertext c_f.

Definition 1 (*IND-CPA Secure* (Gentry 2009)). Let HE be any homomorphic encryption scheme, the plaintext space of HE is $\mathbb{Z}_p$, μ_1 and μ_2 are any two distinct plaintexts on $\mathbb{Z}_p$, if for any polynomial time adversary $\mathcal{A}$, there is

$$\begin{aligned} \mathrm{Adv}_{\mathrm{CPA}}[\mathcal{A}] \triangleq & \big|\Pr\left[\mathcal{A}(pk, evk, \mathrm{HE.Enc}_{pk}(\mu_1)) = \mu_1\right] \\ & - \Pr\left[\mathcal{A}(pk, evk, \mathrm{HE.Enc}_{pk}(\mu_2)) = \mu_1\right]\big| \\ & = \mathrm{negl}(\lambda) \end{aligned}$$

, where $(pk, evk, sk) \leftarrow$ **HE.Keygen**(1^{λ}), then the scheme HE is IND-CPA secure.

Definition 2 ($\mathcal{C}$-*Homomorphism* (Gentry 2009)). Suppose FHE is an arbitrary fully homomorphic encryption scheme, $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ is a collection of arithmetic circuits. FHE is called $\mathcal{C}$-Homomorphic, if you take any sequence of circuits $f_\lambda \in \mathcal{C}_\lambda$ and input $\mu_1, \ldots, \mu_\ell \in \{0, 1\}$ with $\ell = \ell(\lambda)$, such that

$$\Pr[\mathbf{FHE.Dec}_{sk}(\mathbf{FHE.Eval}_{evk}(f, c_1, \ldots, c_\ell)) \neq f(\mu_1, \ldots, \mu_\ell)] = negl(\lambda),$$

where $(pk, evk, sk) \leftarrow$ **FHE.Keygen**(1^{λ}), $c_i \leftarrow$ **FHE.Enc**$_{pk}(\mu_i)$, $i \in [\ell]$.

PIR model

The server-side database is often a*bstrac*ted as an n-bit binary string x, namely $x \in \{0, 1\}^n$. The client owns a query index $i \in [n]$. The goal of the PIR protocol is that the client wants to query the server for d_i, which has index $i \in [n]$, without revealing the "i" to the server. To further increase the practicality of PIR, the retrieval data corresponding to the index in the database of server S is usually generalized to the multi-bit case, that is, the database is formalized as n records $d_1 d_2 \cdots d_n$, where d_i is ℓ bit, $i \in [n]$. For the former, the database index corresponds to the single-bit case and is abbreviated as bPIR, while for the latter, the database index corresponds to the multi-bit case and is abbreviated as BPIR.

A PIR agreement usually consists of three parts: $P = (Q, A, C)$,Where Q refers to the query generation algorithm, A refers to the query response algorithm, and C refers to the query result reconstruction algorithm. The specific protocol process is as follows:

- *Step 1* The user determines the query index $i \in [n]$, runs the query generation algorithm Q, and generates k query results $(q_1, q_2, \ldots, q_k) = Q(i)$, where q_j can be expressed as $Q(i, j)$, $j \in [k]$, and k is the number of server copies (usually $k > 1$ for information-theoretic PIR and $k = 1$ for computational PIR). It should be emphasized here that the query generation algorithm Q is a probabilistic generation algorithm, that is, the output of the same i is different each time. This is often achieved by introducing random private factors or probabilistic encryption algorithms (such as fully homomorphic encryption algorithms).
- *Step 2* The user sends query request q_j to server S_j, $j = 1, 2, \ldots k$ respectively.
- *Step 3* After receiving query request q_j, server S_j runs query response algorithm to generate query response $a_j = A(q_j, x)$ based on local database and sends it to user.
- *Step 4* The user runs the query reformulation algorithm C to compute x_i, namely $x_i = C(i, a_1, \ldots, a_k)$.

The definitions of correctness and privacy for PIR protocols are given below.

Definition 3 (*Correctness* (Chor and Gilboa 1997)). Suppose that the size of the database is n, and the protocol participants are client C and k semi-honest servers $S_1, \ldots, S_k, k \geq 1$. A PIR protocol $P = (Q, A, C)$ is correct if and only if $C(i, a_1, \ldots, a_k) = x_i$ for any query index i of client C, where a_j is the response result generated by server S_j running protocol response algorithm A on client C's query.

And for privacy, it is divided into information theory based PIR(IPIR) and computing power based PIR(CPIR). The former mainly means that under any circumstances, even if the server has unlimited computing power, it cannot get any information about the client's query, which guarantees the complete privacy of the user's query. The latter means that the server cannot get any information about the client's query in polynomial time, which guarantees the computational privacy of the user's query. The relevant formal definition is as follows:

Definition 4 (*Complete privacy of user queries* (Chor and Gilboa 1997)). Suppose the size of the database is n, and the protocol participants are client C and k semi-honest servers $S_1, \ldots, S_k$, $k \geq 1$. A PIR protocol $P = (Q, A, C)$ satisfies complete privacy of user queries if and only if for any two query indexes $i_1, i_2 \in [n]$ of client C, and any possible k query requests $(q_1, q_2, \ldots, q_k)$, server S_j cannot distinguish whether query request q_j is generated by client query index i_1 or i_2, which is formally denoted as.

$$\Pr\left[\forall_{j\in[k]} Q(i_1, j) = q_j\right] - \Pr\left[\forall_{j\in[k]} Q(i_2, j) = q_j\right] = negl(n).$$

In addition, the complete privacy of user queries should be based on the assumption that servers do not collusive and communicate with each other.

Definition 5 (*Computational privacy of user queries* (Chor and Gilboa 1997)). Suppose the size of the database is n, and the protocol participants are a client C and a semi-honest server S. A PIR protocol $P = (Q, A, C)$ satisfies complete privacy of user queries if and only if for any two query indices $i_1, i_2 \in [n]$ of client C, with any possible query request q, server S cannot distinguish in polynomial time whether query request q is generated by client query index i_1 or i_1, formally denoted as.

$$|\Pr[Q(i_1) = q] - \Pr[Q(i_2) = q]| = negl(n).$$

Although IPIR protocol can provide absolute privacy protection, it is not necessary in many cases. At the same time, another key problem is that IPIR requires multiple servers to participate in the protocol and assumes that they do not collusive communication with each other, which is too high to be true in reality. This outstanding problem has stimulated research enthusiasm for CPIR. At present, the design of the existing CPIR protocol is mostly based on hard problems, such as quadratic residue, discrete logarithm and lattice problems, etc. This paper also mainly studies the CPIR protocol, and uses the FHE scheme to construct the PIR protocol that satisfies the computational privacy. Unless otherwise emphasized, all PIR protocols presented below are those satisfying computational privacy.

Analysis of homomorphic PIR protocol model-YKPB-PIR

In 2013, Yi et al. proposed a simple FHE-based PIR protocol construction model, referred to as YKPB-PIR protocol model (Yi et al. 2013). Most of the existing FHE-based PIR schemes follow the YKPB-PIR protocol model, and focus on the optimization of the FHE scheme itself. Then based on the optimized FHE scheme, the CPIR scheme with better performance is obtained by selecting appropriate parameter Settings. The following article will specifically introduce the YKPB-PIR protocol model.

Suppose that FHE = (FHE.Keygen, FHE.Enc, FHE.Dec, FHE.Eval), the plaintext space is $\mathbb{Z}_2$, and the maximum supported circuit depth is L. $\boxplus$ and $\boxtimes$ denote homomorphic addition and multiplication operations, respectively. The server-side database t is a binary string x of bits, and the client wants to retrieve the *kbit* of information,$k \in [t]$, whose binary can be expressed as $k = (k_{\ell-1}k_{\ell-2}\cdots k_0)_2$, where $\ell = \lceil \log k \rceil$, $k_i \in \{0,1\}$, or $k = \sum_{0 \le i \le \ell} k_i 2^i$. The YKPB-PIR protocol model can be based on any FHE scheme and consists of the following four algorithms:

- **YKPB−PIR.Keygen**$(1^\lambda, 1^L)$: λ is a security parameter. In the phase of client parameter generation, user A generates the query public and private key pair $(pk, sk) \leftarrow$ **FHE.KeyGen**$(1^\lambda, 1^L)$ based on FHE encryption algorithm and sends the query public key pk to the server.
- **YKPB−PIR.Query**(pk, k): In the client query generation phase, client A first encrypts the secret index $k = (k_{\ell-1}k_{\ell-2}\cdots k_0)_2$: $C_i =$ **FHE.Enc**(pk, k_i), $0 \le i \le \ell - 1$ bit by bit based on the FHE encryption algorithm, generates the query message $Q = (C_0, \ldots, C_{\ell-1})$, and sends it to the server S.
- **YKPB−PIR.Response**(DB, pk, Q): In the server-side query response phase, when the server S receives the query message Q from the client A, the S generates the query response message R according to Algorithm 1.
- **YKPB−PIR.Decode**(sk, R): in the last stage of client decoding, when a customer A server S received was sent after the query response of R news, to decrypt the message R FHE operation algorithm, and then obtained the information retrieval by $a_k =$ **FHE.Dec**(sk, R).

By analyzing YKPB-PIR protocol model, this paper finds the following two main problems:

- In the generation stage of YKPB-PIR protocol model response algorithm, database indexes need to be traversed, and FHE encryption algorithm is run on each index for bit-by-bit encryption. In fact, the encryption operation link is not necessary, can be based on the customer query message $Q = (C_0, \ldots, C_{\ell-1})$ through the assignment to replace, which greatly reduces the number of homomorphic encryption and homomorphic addition number of operations.
- YKPB-PIR protocol response algorithm uses a large number of homomorphic ciphertext concatenation operations, the number of concatenation is the bit length of database scale, and the number of concatenation is positively correlated with homomorphic circuit depth. In order to ensure the correct decryption after homomorphic calculation of these conjunction operations in FHE scheme, it is necessary to sacrifice the parameter size of FHE scheme or intro-

Algorithm 1 YKPB-PIR protocol model query response algorithm

Input: database $DB = a_1 a_2 \cdots a_t$, query $Q = (C_0, C_1, \cdots C_{\ell-1})$, query public key pk

Output: Query response R

Step 1: Traverse $r \in [t]$ and encrypt $r = (r_{\ell-1}r_{\ell-2}\cdots r_0)_2$ bit-by-bit based on FHE encryption algorithm: $C_{r,i} =$ **FHE.Enc**(pk, r_i), $0 \le i \le \ell - 1$;

Step 2: Calculate $\hat{\psi}_r = \boxtimes_{i=0}^{\ell-1} (C_i \boxplus C_{r,i} \boxplus \bar{1})$, where $\bar{1}$ is FHE ciphertext of 1;

Step 3: Calculate $R = \boxplus_{a_r=1} \hat{\psi}_r$.

duce a large number of extra computations to reduce noise, resulting in a high computational complexity of YKPB-PIR protocol model.

PIR protocol model based on mixed homomorphic encryption

This paper optimizes YKPB-PIR protocol model and proposes a PIR protocol model based on mixed homomorphic encryption. Firstly, the main construction idea of PIR protocol model based on mixed homomorphic encryption is given, and then the existing protocol model is given, and the correctness and security are analyzed.

The main idea

In view of the first problem of YKPB-PIR protocol model proposed in the previous section, YKPB-PIR protocol uses a lot of homomorphic encryption operations, which can be replaced by simple assignment operations. The details are as follows: Set r as any index of the database, and the following assignment operation is performed on it in this paper. For any bit of r_i, $0 \le i \le \ell - 1$, if $r_i = 1$, it is denoted as $C_{r,i} = C_i$; otherwise, it is denoted as $C_{r,i} = C_i + \overline{1}$, where $\overline{1}$ is FHE ciphertext of 1. So, if $r = k$, then

$$\hat{\psi}_r = \boxtimes_{i=0}^{\ell-1} C_{r,i} = \overline{1},$$

otherwise

$$\hat{\psi}_r = \boxtimes_{i=0}^{\ell-1} C_{r,i} = \overline{0}.$$

The $\hat{\psi}_r$ obtained by the assignment method is exactly the same as the $\hat{\psi}_r$ generated by the original YKPB-PIR protocol model after traversing the database index and encrypting it bit-by-bit. The assignment method not only avoids the batch homomorphic encryption operation, but also simplifies the ciphertext homomorphic operation when generating $\hat{\psi}_r$.

Aiming at the second problem of YKPB-PIR protocol model proposed in the previous section: YKPB-PIR protocol uses batch serialization operation, this paper proposes an efficiency optimization method based on hybrid FHE encryption scheme. First, the privacy query user generates the privacy query index based on the single multiplication homomorphic encryption scheme (MHE) and sends it to the server. Then the server processes the conjunction operation based on the single multiplication homomorphic encryption scheme (MHE), and then transforms it into the FHE scheme to process the remaining simple operations. The advantage of this method is that the multiplicative operation circuit depth of FHE scheme is independent of the database size and only related to the decryption circuit depth of the single multiplicative homomorphic encryption scheme, so a single multiplicative homomorphic encryption scheme with low decryption circuit complexity can be selected to improve the efficiency of the model.

But the MHE scheme cannot perform homomorphic addition operations. Therefore, the following homomorphic assignment operation cannot be performed

$$C_{r,i} = C_i + \overline{1}.$$

In order to solve this problem, this paper changes the client query generation algorithm in the YKPB-PIR protocol model, and generates the privacy query index $k = (k_{m-1}k_{m-2}\cdots k_0)_2$ as follows:

- Traversal $i = 0, 1, \ldots m-1$, calculation
- $C_i = \mathbf{MHE.Enc}(pk, k_i)$, $C'_i = \mathbf{MHE.Enc}(pk, (k_i \oplus 1))$;
- Generated privacy query index
- $Q = (C_0, C'_0, \ldots, C_{m-1}, C'_{m-1})$.

In this way, r is set as any index of the database. For any bit r_i, the server can perform the following assignment operation: if $r_i = 1$, it is called $C_{r,i} = C_i$; otherwise, it is called $C_{r,i} = C'_i$.

The above are the main construction ideas of the privacy information retrieval technology model based on mixed homomorphic encryption. It can be seen from the above introduction that this model is more suitable for large-scale databases, and the larger the database size, the more obvious the efficiency advantage. The following first gives the bPIR protocol model of the database single bit corresponding to the index, and then extends it to the BPIR protocol model of the database multi-bit corresponding to the index to enhance the practical model.

bPIR protocol model based on mixed homomorphic encryption

Let MHE = (MHE.Keygen, MHE.Enc, MHE.Dec, MHE.Mult), is a single multiplication homomorphic encryption scheme whose decryption circuit depth on $\mathbb{Z}_2$ is d in plaintext space. The decryption function of MHE scheme is denoted as f_{Dec}, and the ciphertext homomorphic multiplication operation is denoted as $\odot$. FHE = (FHE.Keygen, FHE.Enc, FHE.Dec, FHE.Eval) is a FHE scheme on the plaintext space $\mathbb{Z}_2$, which supports the ciphertext homomorphism operation with the maximum depth of the circuit L, $\boxplus$ and $\boxtimes$ represents the homomorphism addition and multiplication of FHE ciphertext respectively. The server-side database DB is the binary string x of t bit. Meanwhile, the client wants

to retrieve the information of k bit, $k \in [t]$, whose binary can be expressed as $k = (k_{\ell-1}k_{\ell-2}\cdots k_0)_2$, where $\ell = \lceil \log\ k \rceil$, $k_i \in \{0,1\}$, $0 \le i \le \ell - 1$. The bPIR protocol model based on mixed homomorphic encryption, Hyb-FHE-bPIR protocol model for short, is composed of the following four algorithms:

- **HybFHE−bPIR.Keygen**$(1^\lambda, 1^L)$: λ for safety parameters, on the client side parameter generation phase, user A run $(sk_{MHE}, pk_{MHE}) \leftarrow$ **MHE.KeyGen**(1^λ) and $(sk_{FHE}, pk_{FHE}) \leftarrow$ **FHE.KeyGen**$(1^\lambda, 1^L)$, query generation public and private key $(pk_{Hyb}, sk_{Hyb}) = (\{pk_{FHE}, \overline{sk_{MHE}}, f_{Dec}\}, sk_{FHE})$, $\overline{sk_{MHE}} \leftarrow$ ***SWHE.Enc***$(pk_{FHE}, sk_{MHE}^{(i)})$ indicates that the sk_{FHE} key is encrypted bit-by-bit. The query public key $pk_{Hyb} = \{pk_{MHE}, pk_{FHE}, \overline{sk_{MHE}}, f_{Dec}\}$ is sent to server S.
- **HybFHE−bPIR.Query**(pk_{Hyb}, k): the client query generation phase, the first customer A index based on single MHE homomorphic encryption algorithm based on secret $k = (k_{\ell-1}k_{\ell-2}\cdots k_0)_2$ by bits encryption: $C_i =$ **MHE.Enc**(pk_{MHE}, k_i), $C_i' =$ **MHE.Enc**$(pk_{MHE}, (k_i \oplus 1))$, $0 \le i \le \ell - 1$, query generation $Q = (C_0, C_0', \ldots, C_{\ell-1}, C_{\ell-1}')$ news, and sent to the server S.
- **HybFHE−bPIR.Response**(DB, pk_{Hyb}, Q): on the server side query response phase, when the server receives the S customer A query message: Q, S according to the algorithm 2 to generate query response message R.
- **HybFHE−bPIR.Decode**(sk_{Hyb}, R): In the final client decoding stage, when the client A receives the query response message R sent by the server S, it runs the FHE decryption algorithm on the message R and gets the retrieved information $b_k =$ **FHE.Dec**(sk_{FHE}, R).

Note that, MHE plaintext space does not have to be $\mathbb{Z}_2$, it can also be $\mathbb{Z}_p$.

Theorem 1 (correctness) *Let MHE and FHE be single multiplication homomorphic encryption scheme and fully homomorphic encryption scheme on* $\mathbb{Z}_2$, *MHE scheme supports homomorphic multiplication operation of any number of times, the circuit depth of decryption function* f_{Dec} *is* d, *FHE maximum supports ciphertext homomorphic operation with circuit depth of* L, $(pk_{Hyb}, sk_{Hyb}) \leftarrow$ **HybFHE−bPIR.Keygen**$(1^\lambda, 1^L)$, *For any* t *bit size database* $DB = b_1 b_2 \cdots b_t$, *any query index* $k \in [t]$, *let* $Q \leftarrow$ **HybFHE−bPIR.Query**(pk_{Hyb}, k), $R \leftarrow$ **HybFHE−bPIR.Response**(DB, pk_{Hyb}, Q), *if* $L \ge d$, *then*

$$\textbf{HybFHE−bPIR.Decode}(sk_{Hyb}, R) = b_k.$$

Proof According to the HybFHE-bPIR protocol model, for any $r \in [t]$, when $r = k$, there is **MHE.Dec**$(sk_{MHE}, \hat{\psi}_r) = 1$, otherwise **MHE.Dec**$(sk_{MHE}, \hat{\psi}_r) = 0$. Because of $L \ge d$, then for any $r \in [t]$, when $r = k$ has **FHE.Dec**$(sk_{FHE}, \hat{\psi}_r') = 1$, otherwise **FHE.Dec**$(sk_{FHE}, \hat{\psi}_r') = 0$. And because of $b_k \in \{0,1\}$, the following cases to discuss it. When $b_k = 1$, then

Algorithm 2 Query response algorithm of HybFHE-bPIR protocol model

Input: database $DB = b_1 b_2 \cdots b_t$, query $Q = (C_0, C_0', \cdots, C_{\ell-1}, C_{\ell-1}')$, query public key pk_{Hyb}.

Output: Query response R.

Step 1: Traverse $r \in [t]$, r_i, $0 \le i \le \ell - 1$ for any bit of r, $C_{r,i} = C_i$ if $r_i = 1$, otherwise, $C_{r,i} = C_i'$;

Step 2: Calculate $\hat{\psi}_r = \bigodot_{i=0}^{\ell-1} C_{r,i}$;

Step 3: Walk $r \in [t]$ and calculate

$$\overline{\hat{\psi}_r} \leftarrow \textbf{FHE.Enc}(pk_{FHE}, \hat{\psi}_r^{(i)}),\ \overline{\psi_r'} = f_{Dec}(\overline{sk_{MHE}}, \overline{\hat{\psi}_r});$$

Step 4: Calculate and output $R = \boxplus_{r \in [t], b_r = 1} \overline{\psi_r'}$.

$$R = \boxplus_{r\in[n],b_r=1} \overline{\psi'_r} = \overline{\psi'_k} = \overline{1},$$

where $\overline{1}$ indicates $\mathbf{FHE.Dec}(sk_{FHE},\overline{1}) = 1$.

When $a_k = 0$, then.

$$R = \boxplus_{r\in[n],b_r=1} \overline{\psi'_r} = \overline{0},$$

where $\overline{0}$ indicates $\mathbf{FHE.Dec}(sk_{FHE},\overline{0}) = 0$.

Theorem 2 (Security). *Assuming that both MHE and FHE schemes based on HybFHE-bPIR protocol model meet IND-CPA security, HybFHE-bPIR protocol model also meets IND-CPA security.*

Proof Let single homomorphic encryption scheme MHE=(MHE.Keygen, MHE.Enc, MHE.Dec, MHE.Mult) and partial homomorphic encryption scheme FHE=(FHE.Keygen, FHE.Enc, FHE.Dec, FHE.Eval) is the basic encryption scheme used in the construction of HybFHE-bPIR protocol model. If FHE scheme meets IND-CPA security, Next, it is proved that for the HybFHE-bPIR protocol model proposed in this section, if there is a probability polynomial adversary $\mathcal{A}$ that ε attacks successfully with non-negligible advantage, then there must be a probability polynomial adversary $\mathcal{A}'$ based on adversary $\mathcal{A}$ that can successfully attack MHE encryption scheme with non-negligible advantage.

First, the opponent $\mathcal{A}'$ and a challenger $\mathcal{C}'$ instantiate the semantic security game of the MHE encryption scheme as follows: The challenger $\mathcal{C}'$ sends to the opponent $\mathcal{A}'$ to query the public key pk_{MHE}. For message items m_0 and m_1, say $m_0 = 0$, $m_1 = 1$, send m_0 and m_1 to challenger $\mathcal{C}'$. Challenger $\mathcal{C}'$ randomly selected $b \in \{0,1\}$, generated $e_b = \mathbf{MHE.Enc}(pk_{MHE}, m_b)$, and sent to the opponent $\mathcal{A}'$.

Then, adversary $\mathcal{A}'$ plays the semantic security game of the Challenger and adversary $\mathcal{A}$ instantiating the HybFHE-bPIR protocol model: Adversary $\mathcal{A}$ sends to $\mathcal{A}'$ two different database indexes, $1 \le i,j \le t$. Let's say $x_0 = i$, $x_1 = j$. $\mathcal{A}'$ then randomly select $q \in \{0,1\}$, and generate the query Q_q as follows: Let x_q binary expression as $(\alpha_{q,\ell-1}\alpha_{q,\ell-2}\cdots\alpha_{q,0})_2$, $\ell = \lceil \log t \rceil$, x_q bit-by-bit encryption: Traversing $i \in [\ell]$, if $\alpha_{q,i} = 0$, let $C_{q,i} = \hat{0}$, $C'_{q,i} = e_b$, and vice versa, let $C_{q,i} = e_b$, $C'_{q,i} = \hat{0}$, i.e. $Q_q = (C_{q,0}, C'_{q,0}, \ldots, C_{q,\ell-1}, C'_{q,\ell-1})$, where $\hat{0}$ and $\hat{1}$ stand for $\mathbf{MHE.Dec}(sk_{MHE},\hat{0}) = 0$, $\mathbf{MHE.Dec}(sk_{MHE},\hat{1}) = 1$ respectively, then $\mathcal{A}'$ sends Q_q to the adversary $\mathcal{A}$, and then the adversary $\mathcal{A}$ returns a guess q'.

Because the probability that e_b is $\hat{0}$ or $\hat{1}$ is 1/2. When $e_b = \hat{0}$ is used, the elements in Q_q are all $\hat{0}$, According to the HybFHE-bPIR protocol model query response algorithm, For all $r \in [t]$, there is $\hat{\psi}_r = \bigodot_{i=0}^{\ell-1} C_{r,i} = \hat{0}$, thus $\overline{\psi'_r} = f_{Dec}(\overline{sk_{MHE}}, \overline{\hat{\psi}_r}) = \overline{0}$, where $\overline{0}$ and $\overline{1}$ stand for $\mathbf{FHE.Dec}(sk_{FHE},\overline{0}) = 0$ and $\mathbf{FHE.Dec}(sk_{FHE},\overline{1}) = 1$ respectively, and finally there is $R = \boxplus_{r\in[n],b_r=1} \overline{\psi'_r} = 0$. In this case, since the FHE scheme meets IND-CPA security, the guess of the adversary $\mathcal{A}$ is independent of q, that is, the probability of the adversary $\mathcal{A}$ guessing $q' = q$ is 1/2.

When $e_b = \hat{1}$, Q_q is the privacy query of x_q, namely $Q_q \leftarrow \mathbf{HybFHE-bPIR.Query}(pk_{Hyb}, x_q)$. As in this event, $\mathcal{A}'$ behaved no differently from the actual challenger $\mathcal{C}$. So let's say that $\mathcal{A}$ has a success rate of $1/2 + \epsilon$.

$\mathcal{A}'$ guesses b' according to the following scheme, if the opponent $\mathcal{A}$ guesses $q' = q$ correctly, $\mathcal{A}'$ will order $b' = 1$, otherwise, $b' = 0$. In summary, the probability of $\mathcal{A}'$ guessing correctly can be calculated as.

$$\Pr(b' = b) = \frac{1}{2}\left(\frac{1}{2}\right) + \frac{1}{2}\left(\frac{1}{2} + \epsilon\right) = \frac{1}{2} + \frac{\epsilon}{2}.$$

Therefore, $\mathcal{A}'$ can attack successful MHE encryption schemes with non-negligible advantage, which contradicts the conditions assumed by the theorem. Therefore, HybFHE-bPIR query response algorithm based on mixed homorphic encryption satisfies IND-CPA security.

In order to further enhance the practicability of HybFHE-bPIR protocol model, this paper extends it to the case of multi-bit index corresponding database, which is called HybFHE-BPIR protocol model.

BPIR protocol model based on mixed homomorphic encryption

MHE=(MHE.Keygen, MHE.Enc, MHE.Dec, MHE.Mult) be a single multiplicative homomorphic encryption scheme whose plaintext space is $\mathbb{Z}_2$ and the decryption circuit depth on $\mathbb{Z}_2$ is d. The decryption function of MHE scheme is denoted as f_{Dec}, and the ciphertext homomorphic multiplication operation is denoted as $\odot$. FHE=(FHE. Keygen, FHE. Enc, FHE. Dec, FHE. Eval) is a partial homomorphic encryption scheme on $\mathbb{Z}_2$, which supports a maximum ciphertext homomorphic operation

with circuit depth of L, and represents homomorphic addition and multiplication respectively.

The size of the server database is t bits, which is evenly divided into m 1-bit data blocks:$DB = B_1|B_2\cdots|B_m$, $t = m\cdot\ell$, $B_i = (b_{i,1}, b_{i,2}, \ldots, b_{i,\ell})$. The customer wants to retrieve the kth data block B_k, $k \in [m]$, whose binary can be expressed as $k = (k_{l-1}k_{l-2}\cdots k_0)_2$, where $k_i \in \{0,1\}$, $0 \le i \le \ell - 1$. Then, BPIR privacy information retrieval technology model based on mixed homomorphic encryption, which is referred to as HybFHE-BPIR protocol model for short in this chapter, is composed of the following four algorithms:

- **HybFHE−BPIR.Keygen**$(1^\lambda, 1^L)$: In the client parameter generation phase, user A runs and generates the query public and private key pair $(pk_{Hyb} = \{pk_{MHE}, pk_{FHE}, \overline{sk_{MHE}}, f_{Dec}\}, sk_{Hyb}) \leftarrow$ **HybFHE−bPIR.Keygen**$(1^\lambda, 1^L)$, and sends the query public key pk_{Hyb} to the server.
- **HybFHE−BPIR.Query**(sk, k): In the client query generation phase, user A selects the query index $k \in [1, n]$, generates a query message $k \in [1, n]$, and sends it to the server S.
- **HybFHE−BPIR.Response**(DB, pk, Q): In the server-side query response phase, when server S receives the query message Q from client A, server Q generates the query response message R according to Algorithm 3.
- **HybFHE−BPIR.Decode**(sk, R): In the final client decoding phase, when client A receives the query response message R sent by server S, it runs the FHE decryption algorithm on the message R, and gets the retrieved message block $B' =$ (**FHE.Dec**(skF_{SWHE}, R_1), **FHE.Dec**(sk_{FHE}, R_2), $\ldots$, **FHE.Dec**(sk_{FHE}, R_ℓ).

In essence, the HybFHE-BPIR protocol model can be regarded as the parallel operation of one HybFHE-bPIR protocol: the query user $DB = B_1|B_2\ldots|B_{2^m}$ extracts one bit of data from the same position in each data block to form the following m - bit database $DB_i = (b_{1,i}, b_{2,i}, \ldots, b_{m,i})$, $1 \le i \le \ell$. Then the user only needs to retrieve the k - bit data from each DB_i. Therefore, the proof of correctness and security of the HybFHE-BPIR protocol model is similar to the above section, and the details will not be repeated.

Conclusions

This paper focuses on the research of PIR protocol model based on homomorphism, especially for large-scale database retrieval. The proposed mixed homomorphic encryption is beneficial to the noiseless single multiplicative single homomorphic encryption scheme to deal with large-scale serialization operations. Then, the homomorphic scheme is used to process the remaining simple operations. The homomorphic operation of

Algorithm 3 HybFHE-BPIR Protocol model query response algorithm

Input : database $DB = B_1 \,||\, B_2 \cdots ||\, B_m$,among them, $B_j = (b_{j,1}, b_{j,2}, \cdots, b_{j,\ell})$, $j \in [\ell]$ inquire $Q = (C_0, C_0', \cdots, C_{\ell-1}, C_{\ell-1}')$, Public Key pk_{Hyb} .

Output : Query Response R .

Step1 : traversal $r \in [m]$, For any bit r_i of r , 0^1 $0 \le i \le \ell - 1$, if $r_i = 1$, then $C_{r,i} = C_i$, otherwise, $C_{r,i} = C_i'$;

Step2 : calculate $\hat{\psi}_r = \bigodot_{i=0}^{\ell-1} C_{r,i}$;

Step3 : traversal $r \in [t]$, calculate

$$\overline{\hat{\psi}_r} \leftarrow \textbf{\textit{FHE.Enc}}(pk_{FHE}, \hat{\psi}_r^{(i)}) \ , \quad \hat{\psi}_r' = f_{Dec}(\overline{sk_{MHE}}, \overline{\hat{\psi}_r}) \ ;$$

Step4 : traversal $c \in [\ell]$, calculate $R_c = \boxplus_{r\in[t], b_{r,c}=1} \hat{\psi}_r'$, output $R = (R_1, R_2, \cdots, R_\ell)$.

the homomorphic scheme only needs to support the decryption circuit of the single homomorphic encryption scheme, which no longer depends on the size of the database, and can greatly improve the efficiency of the homomorphic privacy information retrieval model. Note that, for small-scale database retrieval, our model will lose its advantages due to the need for homomorphic transformation from MHE scheme to FHE Scheme, which is relatively expensive in this situation. Next, in order to further improve the practicality of the model, we will provide effective implementation examples.

Acknowledgements
The authors would like to thank the anonymous reviewers for helpful comments. This work was sponsored in part by the National Natural Science Foundation of *. And all data that support the findings of this study is included in this manuscript. Besides, the authors declare that there is no conflict of interest regarding.

Author contributions
All authors have seen the manuscript and approved to submit to your journal.

Funding
This work was sponsored in part by the National Natural Science Foundation of China [Grant-Nos. 61902428, 6210071026, 62202493].

Declarations

Competing interests
The authors declare that there is no conflict of interest regarding the publication of this paper.

Received: 13 April 2023 Accepted: 9 June 2023
Published online: 01 December 2023

References

Aguilar-Melchor C, Barrier J, Fousse L et al (2016) XPIR: private information retrieval for everyone. Proc Priv Enhancing Technol 2:155–174
Akbar H, Zubair M, Malik MS (2023) The security issues and challenges in cloud computing. Int J Electron Crime Investig 7(1):13–32
Ambainis A (1997) Upper bound on the communication complexity of private information retrieval. In: International colloquium on automata, languages, and programming. Springer, Berlin, pp 401–407
Angel S, Chen H, Laine K et al (2018) PIR with compressed queries and amortized query processing. In: 2018 IEEE symposium on security and privacy (SP). IEEE, pp 962–979
Beimel A, Ishai Y (2001) Information-theoretic private information retrieval: a unified construction. In: Proceeding of the 28th international colloquium on automata, languages and programming, Crete, Greece. Springer, Berlin, pp 912–924
Brakerski Z, Vaikuntanathan V (2011a) Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Proceedings of the 31st annual conference on advances in cryptology. Springer, Berlin, pp 505–524
Brakerski Z, Vaikuntanathan V (2011b) Efficient fully homomorphic encryption from (standard) LWE. In: Proceedings of the 52th annual symposium on foundations of computer science. IEEE Computer Society, Washington, DC, pp 97–106
Cachin C, Micali S, Stadler M (1999) Computationally private information retrieval with polylogarithmic communication. In: Proceedings of 17th international conference on the theory and application of cryptographic techniques, Prague, Czech Republic. Springer, Berlin, pp 402–414
Chor B, Gilboa N (1997) Computationally private information retrieval. In: Proceedings of the 29th annual ACM symposium on theory of computing, El Paso, TX, USA. ACM, New York, NY, pp 304–313
Chor B, Goldreich O, Kushilevitz E et al (1995) Private information retrieval. In: Proceeding of the 36th annual symposium on foundations of computer science. IEEE, pp 41–50
Doröz Y, Sunar B, Hammouri G (2014) Bandwidth efficient PIR from NTRU. In: International conference on financial cryptography and data security. Springer, Berlin, pp 195–207
Eltarjaman W, Annadata P (2016) Comparative study of private information retrieval protocols. In: Proceedings of the 6th international multi-conference on complexity, informatics and cybernetics, pp 204–209
Gautam D, Shivhare R (2022) Cloud security aspects using homomorphic encryption: a review. Res J Eng Technol Med Sci 5(04). ISSN: 2582-6212
Gentry C (2009) Fully homomorphic encryption using ideal lattices. STOC 9:169–178
Gentry C, Halevi S (2019) Compressible FHE with applications to PIR. In: Theory of cryptography: 17th international conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, proceedings, part II. Springer, Cham, pp 438–464
Ichibane Y, Gahi Y, Guennoun M et al (2015) Performance analysis of private information retrieval scheme based on homomorphic encryption. In: Proceedings of the 5th international conference on information communication technology and accessibility (ICTA). IEEE, pp 1–6
Ishai Y, Kushilevitz E (1999) Improved upper bounds on information-theoretic private information retrieval. In: Proceedings of the 31th annual ACM symposium on theory of computing, Atlanta, GA, USA. ACM, New York, NY, pp 79–88
Itoh T (1999) Efficient private information retrieval. IEICE Trans Fundam Electron Commun Comput Sci E82-A(1):11–20
Kushilevitz E, Ostrovsky R (2000) One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In: Proceedings of advances in cryptology, Bruges, Belgium. Springer, Berlin, pp 104–121
Kushilevitz E, Ostrovsky R (1997) Replication is not needed: single database, computationally-private information retrieval. In: Proceedings of the 38th IEEE symposium on foundations of computer science, Miami Beach, FL, USA. IEEE, Los Alamitos, CA, pp 364–373
Li Z, Ma C, Wang D et al (2017) Toward single-server private information retrieval protocol via learning with errors. J Inf Secur Appl 34:280–284
Menon SJ, Wu DJ (2022) Spiral: fast, high-rate single-server PIR via FHE composition. In: 2022 IEEE symposium on security and privacy (SP). IEEE, pp 930–947
Mosca M (2014) Post-quantum cryptography. Springer, Cham
Mughees MH, Chen H, Ren L (2021) OnionPIR: response efficient single-server PIR. In: Proceedings of the 2021 ACM SIGSAC conference on computer and communications security, pp 2292–2306
Rivest RL, Adlman L, Dertouzos ML (1978) On data banks and privacy homomorphisms. Found Secur Comput 4(11):169–180
Rout C, Sethi S, Sahoo RK et al (2022) Empirical analysis of the impact of homomorphic encryption on cloud computing. Intell Syst Appl Sel Proc ICISA 2023:107–120
Sinha A, Singh NK, Srivastava A et al (2023) Cloud computing security, risk, and challenges: a detailed analysis of preventive measures and applications. In:Machine intelligence, big data analytics, and IoT in image processing: practical applications, p 225
Wang S, Agrawal D, El Abbadi A (2010) Generalizing PIR for practical private retrieval of public data. In: Proceedings of the 24th IFIP annual conference on data and applications security and privacy, Rome, Italy. Springer, Heidelberg, pp 1–16
Yi X, Kaosar MG, Paulet R et al (2013) Single-database private information retrieval from fully homomorphic encryption. IEEE Trans Knowl Data Eng 25(5):1125–1134

Publisher's Note

Wei-tao Song born in 1989, Ph.D., his research interests include cryptography and privacy protection.