Copyright ⓒ 2024 KSII

An Efficient PSI-CA Protocol Under the Malicious Model

Jingjie Liu[1], Suzhen Cao[1*], Caifen Wang[2], and Chenxu Liu[1]
[1] College of Computer science and Engineering, Northwest Normal University
Lanzhou, Gansu 730070 China
[e-mail: 2022212154@nwnu.edu.cn]
[2] College of Big Data and Internet, Shenzhen Technology University
Shenzhen, Guangdong 518118 China
[e-mail: wangcf@nwnu.edu.cn]
*Corresponding author: Suzhen Cao

Received June 26, 2023; revised August 31, 2023; revised December 11, 2023; accepted March 6, 2024; published March 31, 2024

Abstract

Private set intersection cardinality (PSI-CA) is a typical problem in the field of secure multi-party computation, which enables two parties calculate the cardinality of intersection securely without revealing any information about their sets. And it is suitable for private data protection scenarios where only the cardinality of the set intersection needs to be calculated. However, most of the currently available PSI-CA protocols only meet the security under the semi-honest model and can't resist the malicious behaviors of participants. To solve the problems above, by the application of the variant of Elgamal cryptography and Bloom filter, we propose an efficient PSI-CA protocol with high security. We also present two new operations on Bloom filter called IBF and BIBF, which could further enhance the safety of private data. Using zero-knowledge proof to ensure the safety under malicious adversary model. Moreover, in order to minimize the error in the results caused by the false positive problem, we use Garbled Bloom Filter and key-value pair packing creatively and present an improved PSI-CA protocol. Through experimental comparison with several existing representative protocols, our protocol runs with linear time complexity and more excellent characters, which is more suitable for practical application scenarios.

Keywords: Bloom filter, malicious model, private set intersection cardinality, zero-knowledge proof, secure multi-party computation.

The National Natural Science Foundation of China (No. 62262060); Industrial support plan project of Gansu Provincial Department of Education (2022CYZC-17).

http://doi.org/10.3837/tiis.2024.03.011 **ISSN : 1976-7277**

1. Introduction

Private set intersection (PSI) is a typical problem among the secure multi-party computation domain [1], which enables two or more participants to securely compute the intersection of their private sets without revealing any other information about their private sets [2]. For example, for two mutually distrustful participants P_1 and P_2, their private sets are S_1 and S_2, which $|S_1| = m$, $|S_2| = n$, and they want to compute $S_1 \cap S_2$. Private set intersection cardinality (PSI-CA) and private set union cardinality (PSU-CA) are the important variants of PSI which have caused attention of some researchers. The PSI-CA problem is when the participants are P_1 and P_2, and the sets they hold are S_1 and S_2, trying to compute $|S_1 \cap S_2|$. In many scenarios where data privacy needs to be protected, the technology of PSI-CA has significant meanings and has applied in many realistic scenarios such as advertisement conversion rate calculation [3], social network contacts exploration [4], gene sequence match detection [5], infectious disease patient tracing [6].

Take the problem of calculating the advertisement conversion rate for instance, the calculation of advertisement conversion rate is a kind of important application in PSI-CA which means the proportion of users who are influenced by the ad to make a purchase or register to the total number of clicks on the ad. In the real world, the merchants hold the information of people who purchase the products while the advertisers own the information of people who click the advertisement. In order to calculate the conversion rate, the advertisers need to compute the intersection of the set it holds with the set of the merchant side, so as to find the total number of people who have seen the advertisement and completed the transaction. Finally, the advertisement could calculate the advertisement conversion rate. In this scenario, PSI-CA is more applicable to solve this problem compared to PSI, because using PSI-CA could compute directly the number of people who have seen the advertisement and completed the transaction, and then divide that number by the total number of clicks on the advertisement. However, using the technology of PSI could only obtain the concrete information of users who have seen the advertisement and finished the transaction. But it could definitely leakage private data of users.

Nowadays, most researchers are dedicating to study PSI protocols and so neglecting the research on PSI-CA and PSU-CA. Therefore, it's high time to expand the content of the private sets protection field to meet the current privacy protection needs of society better. What's more, most of the currently available PSI protocols only meet the security under the semi-honest model, and it can't guarantee security under the malicious model. Therefore, it can hardly apply in practical scenarios.

In addition, most of the current PSI-CA protocols are constructed by Bloom Filter and the homomorphic encryption algorithm. However, the Bloom Filter's problem of false positive makes the final results exist error. In some practical application environments with high precision requirements for data, such as national defense and military. This kind of error is not allowed to exist. These existing protocols are therefore difficult to apply in this type of environment.

To address these issues above, using Elgamal cryptography and Bloom filter, we firstly present a local two-party PSI-CA protocol. Bloom filter is a special data structure which enables users to map the set elements to an array according to hash functions. To enhance the security of data, we propose two new operations on Bloom filter called IBF and BIBF. Using the variant of Elgamal cryptography for the sake of the security of data during interaction. The plaintext is not the message when decrypted, which is located in the index part of the

decryption result. This property makes it easy for the participants to determine whether the element belongs to the set intersection. Utilizing zero-knowledge proof, our protocol satisfy security not only under the semi-honest model, but also the malicious model. After the protocol has been executed, only one side gets the result of intersection cardinality, the other gets nothing. To further improve the accuracy of the final results, we present an improved PSI-CA protocol. Innovatively using key-value pair packing technology and Garbled Bloom Filter, we have achieved a significant reduction in the mean relative error of the protocol. Finally, the accuracy of the resulting intersection cardinality is significantly improved.

1.1 Our Contributions

In this paper, we present two efficient two-party PSI-CA protocols with high security. Our main contributions are:

(1) Present two new operations on Bloom filter called IBF and BIBF, which could improve privacy of data;
(2) Design two efficient two-party PSI-CA protocols, the first protocol is secure under the malicious adversary model, which could resist the malicious behavior. The improved protocol has higher result accuracy compared to the previous one;
(3) Using ideal-reality simulation paradigm, we prove that our first protocol is secure under the malicious adversary model and provide a complete security proof process.

This paper is organized as follows: in section 2, we introduce some related work about PSI and PSI-CA, section 3 presents some main techniques and security model used in our scheme, while section 4 presents our two PSI-CA protocols. In section 5, we prove the security under malicious model of the protocol. Section 6 presents the analysis of efficiency, the accuracy of results and the functional comparison with other schemes. Finally, we summarize the paper and prospect the future research directions.

2. Related Work

Using homomorphic encryption and inadvertent polynomial valuation, the research of protocol was first put up by Freedman et al. [7] in 2004. And then a lot of researches on PSI have followed. In 2019, Pinkas et al. [8] presented a notation of multi-point OPRF and depended on the construction of high order polynomials, which could reduce communication cost while reducing the number of times the sender encrypts elements. Song et al. [9] presented a series of protocols on the set operations, which dramatically reduce the computational associated with traditional public key operations using oblivious transfer. However, the above protocols are all traditional local two-party PSI protocols. To better accommodate the involvement of multiple parties, scholars have presented a series of multi-party PSI protocols. Vos et al. [10] realized the “union” operation of private set elements based on elliptic curves, and implement a multi-party PSI protocol for large and small sets respectively. Zhang et al. [11] presented a three-party PSI protocol against semi-honest model, which based on bilinear mapping and three-party key negotiation protocol.

To better resist the malicious behaviors of the participants, using garbled bloom filter, Ben-Efraim et al. [12] presented a malicious secure multi-party protocol that can be used against any number of corrupt parties.

In the era of big data, with the surge in data processing, the advantages of cloud computing in the era of big data are coming to the fore, many more solutions have emerged as scholars have begun to investigate on outsourcing large amounts of heavy computing tasks to cloud servers. Abadi et al. [13] presented an outsourcing PSI protocol called O-PSI, which let the

cloud server perform a large number of complex calculations. After that, Abadi et al. [14] presented a verifiable delegated PSI protocol named VD-PSI. This protocol introduced a verification protocol into the outsourcing PSI protocol, where the participants can verify the correctness of the results after receiving them. Based on the O-PSI protocol, Yang et al. [15] presented a delegated PSI protocol, which has more advantages in computational efficiency compared to O-PSI. In 2022, Wei et al. [16] presented a PSI protocol based on semi-trusted cloud server with the help of the oblivious pseudo-random functions.

PSI-CA, as a branch of the PSI, has not had much related research work compared to PSI. Egert et al. [17] presented a local two-party PSI-CA protocol based on Bloom filter and Elgamal cryptography, but it only satisfies the security under the semi-honest model. Mihaela et al. [18] proposed a two-party PSI-CA scheme based on Paillier homomorphic encryption algorithm. This protocol is secure under honest but curious model. However, the computation cost is too high because the Paillier homomorphic encryption algorithm. Davidson et al. [19] proposed a toolkit about the set operation, including the calculation of set union, set intersection and the cardinality of the intersection and union. It enriches the functionality of set operations but it can't resist the malicious adversary attacks. To resist the malicious behaviors of adversary, combining zero-knowledge proof and homomorphic encryption, Debnath et al. [20] proposed a two-party PSI-CA protocol which has poor performance in terms of efficiency. Using zero-knowledge proofs, GM cryptography algorithm, Debnath et al. [21] also proposed another PSI-CA scheme under semi-honest model. However, all of these protocols have the disadvantage of not being able to resist the malicious behaviors of the participants or having high computation cost.

3. Preliminaries

Our protocol primarily utilizes Elgamal cryptography and Bloom filter and ideal-realistic simulation paradigm in the security proof process. Therefore, in this section, we mainly introduce Elgamal cryptography, Bloom filter, ideal-realistic simulation paradigm and the malicious model.

3.1 Elgamal cryptography

Elgamal cryptography is an algorithm based on DDH assumption which was proposed by Tather Elgamal [22] in 1985. We use a variant of Elgamal cryptography in our protocol. The concrete algorithm process is as follows:

(1) Key Generation: as for a multiplicative cyclic group G of order q, where g is a generator of the group G. Choose $x \leftarrow Z_q$ randomly, compute $y = g^x \bmod q$. Then $(pk, sk) = (y, x)$.

(2) Encryption: for plaintext m, compute $c = Enc_{pk}(m) = (c_1, c_2)$, where $c_1 = g^r \bmod q$, $c_2 = g^m y^r \bmod q$, $r \leftarrow Z_q$.

(3) Decryption: as for ciphertext c, it can be decrypted as $g^m = Dec_{sk}(c) = (c_1)^{-x} \cdot (c_2) \bmod q$.

Note that the result of the decryption of this algorithm is g^m, not plaintext m.

The Elgamal algorithm has additive homomorphic property. Let E be the Elgamal encryption algorithm, As for the ciphertext $C_1 = E(m_1)$ and $C_2 = E(m_2)$, it has $E(m_1 + m_2) = E(m_1) \cdot E(m_2) = C_1 \cdot C_2$.

3.2 Bloom Filter

Bloom filter is a special data structure which was proposed by Burton Bloom [23] in 1970. It can be used to represent set elements and easy to query them. Bloom filter is composed by an array of m bits and k hash functions $\{h_1(),...,h_k()\}$. Initially all the bits in the filter are set to zero. If an element $x \in S$ intends to insert into the Bloom filter, it should set the bits $h_i(x)$ to one, where $1 \le i \le k$. **Fig. 1** presents the concrete algorithm about the insertion operation. As for an element, only when all the k positions it is mapped to are set to one [24], the element can be judged to belong to the set, otherwise it does not belong to the set. **Fig. 2** presents the algorithm for the membership test process.

However, the Bloom filter exists false positive problem, which is a situation that an element does not belong to the set S but can be tested successfully in the Bloom filter. **False positive** probability rate is related to the number of hash functions, number of bits in the Bloom filter and the number of elements to be inserted.

Algorithm 1: Insert an element into Bloom filter

```
for each i ∈ [1, k] do
    h(x) = i
    if BF[i] == 0 then
        BF[i] = 1
    end
end
```

Fig. 1. Algorithm for Bloom filter insertion

Algorithm 2: Bloom filter member test

```
j = 1;
flag = 1;
while flag == 1  and  j <= k do
    h_j(x) = i;
    if BF[i] == 0 then
        flag = 0;
    end
    j = j + 1
end
return flag
```

Fig. 2. Algorithm for Bloom filter member test

We present two operations on Bloom filter named IBF and BIBF.

Let BF be a Bloom filter, IBF is the inversed BF by bits. If a bit in BF is set to 1, then set it to 0. Otherwise, if a bit is set to 0, then set it to 1. **Fig. 3** presents the algorithm for the process of inversing Bloom filter.

Algorithm 3: Inverse Bloom filter

```
j = 0
while j <= m do
    if B[j] == 1 then
        BF[j] = 0;
    else
        BF[j] = 1;
    end
    j = j + 1
end
return IBF
```

Fig. 3. Algorithm for inversing the Bloom filter.

On the basis of IBF, blinding each bit of the IBF by multiplying it by a random number, the obtained result is called BIBF. **Fig. 4** shows the algorithm for the process of blinding the inversed Bloom filter.

Algorithm 4: Blind inversed Bloom filter

```
j = 0
while j <= m do
    BIBF[j] = IBF[j] * r_j
    j = j + 1
end
return BIBF
```

Fig. 4. Algorithm for blinding inversed Bloom filter.

Garbled Bloom Filter is the variant of the traditional Bloom Filter. Comparing to the traditional Bloom Filter, the Garbled Bloom Filter has an array which contains random string rather than the character of 0 or 1. This feature could not only decrease the error caused by the problem of false positive but also improve the security of data storage. In our improved protocol, before the client C construct the Garbled Bloom Filter GBF_C, it should construct a set of key-value pair and then pack them into the GBF_C. As for a key-value pair (x, y), it has $y = \sum_{i=1}^{t} GBF(h_i(x))$. The positions in the GBF_C that do not satisfy this condition are stored as random strings.

3.3 Ideal-realistic Simulation Paradigm

The ideal-realistic simulation paradigm is the main method used for security proofs in the domain of secure multi-party computing. It compares the implementation of the PSI-CA protocol by simulating an ideal model with a realistic situation, thus, it can indirectly prove the security of the protocol [25].

In the ideal model, the function of the protocol is computed by the trusted third party, and then sends the result to the participant. However, in the real model, it splits the function into multiple message functions and communicates between the participants to complete the computation. Finally, the security of the PSI-CA protocol is demonstrated by proving that the view of the ideal world achieves indistinguishability from the view of the real world.

3.4 the Malicious Model

The malicious model is another typical adversary model in secure multi-party computing. In the malicious adversary model, comparing to the honest model or semi-honest model, the participant will not execute the protocol honestly, but will perform malicious operations in the course of executing the protocol such as tampering the input information, terminating the protocol early, and refusing to participate in the protocol [25].

Our protocol contains two party named the client and the server which the client holds the set X and the server holds the set Y. Also, $|X|=m$, $|Y|=n$. Therefore, we define a two-party protocol π computing function f where $f:(\{0,1\}^*)^m\times(\{0,1\}^*)^n\to f_{|\cap|}\times\perp$, where $\{0,1\}^*$ denotes the field of input elements, m and n denote the cardinality of two sets respectively, and $f_{|\cap|}$ denotes the cardinality of intersection of two sets. We can conclude that the client obtains the cardinality of intersection $|X\cap Y|$ and the server obtains nothing.

Specifically speaking, the malicious party may execute the following types of attacks:

(1) A malicious party can tamper with the starting input. The malicious client C could forge the Bloom filter BF_C represented by its set in the first phase of the protocol to obtain more information about the set held by the server S. The malicious client will try to insert all elements of the universe set U into the Bloom filter, so that each bit of the resulting Bloom filter is set to 1. When performing subsequent steps, there has $|X\cap Y|=|U\cap Y|=|Y|$. That is, the size of the intersection is the size of the server's set. So, it will leak the cardinality of the set held by server S.

(2) A malicious party could tamper the intermediate results or terminate the protocol in advance. It is possible for both S and C to execute the operation.

In order to resist both of these attacks, using the technology of zero-knowledge proof, constructing proofs to guarantee the correctness of transmitted messages before interaction. When the receiver obtains the message, it should verify the validity of the proof firstly, if the verification is successful, the receiver receives the message. Otherwise, the party terminates the protocol. This method can effectively resist the malicious behaviors of malicious parties.

3.5 the Zero-knowledge Proof

Our protocol uses zero-knowledge proofs techniques to ensure the security of the protocol under malicious models. The following describes a general zero-knowledge proof of the basic construction process [26].

In our scheme, the form of the proof is shown below

$$\pi=\mathrm{Pok}\{(a_1,..,a_l)\,|\,\wedge_{i=1}^{m}K_i=f_i(a_1,..,a_l)\}$$

The specific interaction process between the prover and the verifier is described below.

(1) Commitment:

Firstly, the Prover picks $t_1,...,t_l$ uniformly at random, then the prover computes a commitment $\overline{K_i}$:

$$\overline{K_i}=g_i(t_1,...,t_l),\ i=1,...,m$$

After that, the prover sends the commitment $\overline{K_i}$ to the verifier.

(2) Challenge:

Verifier picks a challenge number h randomly from space C, and sends h to the prover.

After that, the prover computes $n_j = t_j + c \cdot a_j$, where $j = 1,...l$. The prover then sends $\{n_1,...,n_l\}$ to the verifier.

(3) Verify

After receiving $\{n_1,...,n_l\}$, The verifier checks if $g_l(n_1,...n_l) = \overline{K_i} \cdot K_i^h$ exist or not, where $l = 1,...,m$. If the equation exists then the verifier accepts the result otherwise rejects.

4. Our PSI-CA Protocol

In our PSI-CA protocol, the parties are the client C and the server S, and they hold sets $X = \{x_1,...,x_m\}$ and $Y = \{y_1,...,y_n\}$ respectively, which $|X| = m$, $|Y| = n$. C wants to find the set intersection cardinality with S. After the protocol is executed, the client C gets the output as the intersection cardinality.

The following **Table 1** shows the relevant symbols and descriptions required in the protocol.

Table 1. Description of symbols

Symbol	Description
C	The client
S	The server
$X = \{x_1,...,x_m\}$	The set held by the client
$Y = \{y_1,...,y_n\}$	The set held by the server
m	X 's cardinality
n	Y 's cardinality
G	The multiplicative cyclic group
q	The order of the group G
g	The generator of the group G
x	private key of C
y	public key of C
$r_1,...,r_m$	Random numbers generated by C
$z_1,...,z_n$	Random numbers generated by S
$H_0 : \{0,1\}^* \to \{0,1\}^\kappa$	The hash function chosen by C

The PSI-CA protocol under malicious model.

Input: The client C inputs the private input set $X = \{x_1,...,x_m\}$, the server inputs the private input set $Y = \{y_1,...y_n\}$. And the public parameters $P = (G,q,g)$ as their common input. The security parameter κ, λ.

Output: The client C outputs $|X \cap Y|$; the server S outputs $\perp$.

The general structure of our protocol is shown in **Fig. 5**:

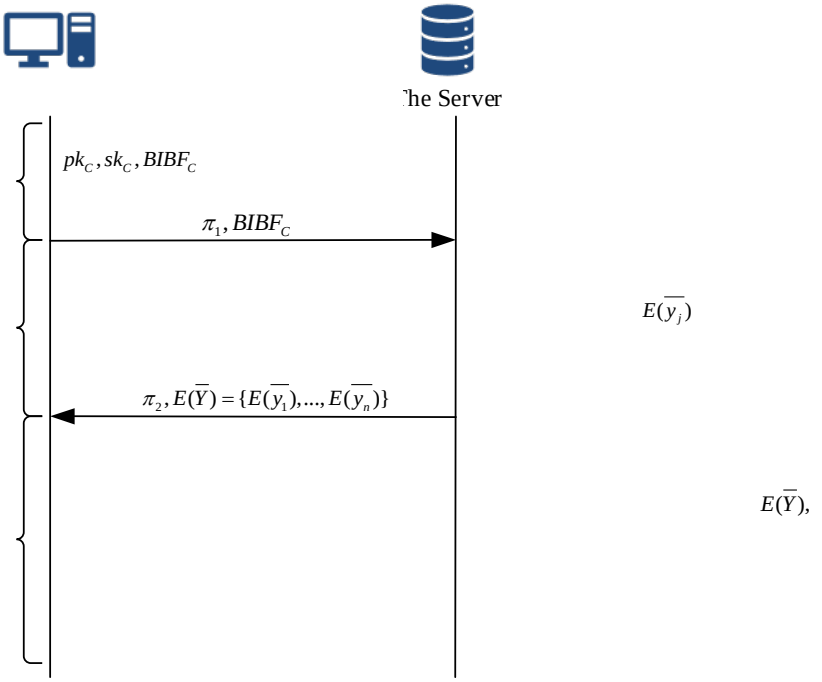

Fig. 5. Structure of PSI-CA protocol

Step 1. Setup phase

(1) For a multiplicative cyclic group G with order q, and g is its generator. The client C picks an uniformly random value $x \xleftarrow{R} Z_q$ and computes $y = g^x \bmod q$. The client C's key pair is $(pk_C, sk_C) = (y, x)$.

(2) The client C inserts every element $x_i \in X$ $(1 \le i \le m)$ from the set X into the Bloom filter, executes the algorithm 1 shown in **Fig. 1**. And then gets the result BF_C.

(3) The client C performs the inversing algorithm shown in **Fig. 3** for each bit of the BF_C. This gets IBF_C.

(4) Picking uniformly random values $r_1, r_2, ..., r_m \xleftarrow{R} Z_q^*$ to blind every bit of IBF_C, where $\overline{x_i} = r_i \cdot IBF_C(x_i)$, $1 \le i \le m$.
The client then gets $BIBF_C$.

(5) Using zero-knowledge proof, the client C constructs the proof $\pi_1 = \text{PoK}\{(r_1, ..., r_m) \mid \wedge_{i=1}^{m}(BIBF_C[i] = r_i \times IBF_C[i])\}$, the construction and verification processes are illustrated in preliminary. Then, let the proof π_1, $\overline{X} = \{\overline{x_1}, \overline{x_2}, ..., \overline{x_m}\}$ and k hash functions all send to the server S.

Step 2. Computation phase

(1) After receiving the message from the client C, S verifies the correctness of π_1 firstly. If the verification passes, then S receives message and continues the subsequent steps. Otherwise the server S aborts it.

(2) The server S hashes every element from Y by k hash functions, where $\forall y_j \in Y$, computing $h_1(y_j), ..., h_k(y_j)$, $1 \le j \le n$.

(3) The server S finds the elements $a_{h_1(y_j)},...,a_{h_k(y_j)}$ from $\overline{X}=\{\overline{x_1},\overline{x_2},...,\overline{x_m}\}$.

(4) Encrypting the elements with the client's public key pk_C. That is

$E(\overline{y_j})=\{(g^{z_j},g^{a_{h_1(y_j)}}\cdot y^{z_j}),...,(g^{z_j},g^{a_{h_k(y_j)}}\cdot y^{z_j})\}$, $1\le j\le n$, where $z_j \xleftarrow{R} Z_q^*$.

The obtained results constitute the set $E(\overline{Y})=\{E(\overline{y_1}),...,E(\overline{y_n})\}$.

(5) Using zero-knowledge proof, the server S constructs the proof $\pi_2=\mathrm{PoK}\{(z_1,...z_j)\mid\wedge_{j=1}^{n}(c_j=g^{z_j})\}$. Send the proof π_2, $E(\overline{Y})=\{E(\overline{y_1}),...,E(\overline{y_n})\}$ to the client C.

Step 3. Intersection computation phase

(1) When receiving the message from the server S, the client C verifies the correctness of π_2 firstly. If the verification passes, then receives the following message and continues the subsequent steps. Otherwise the server S aborts it.

(2) The client C performs the following operations on each element of the set $E(\overline{Y})$:

$(g^{z_i})^{-x}(g^{a_{h_j(y_i)}}y^{z_i})=g^{a_{h_j(y_i)}}$, $1\le i\le n$, $1\le j\le k$, where x is the private key of C.

If the result of decrypting an item of the set $E(\overline{Y})$ is all 1, then the counter adds 1.

The PSI-CA scheme could be extended to the PSU-CA protocol. Taking advantage of the relationship $|X\cup Y|=|X|+|Y|-|X\cap Y|$, the client could find the private set union cardinality with ease.

Fig. 6 displays the specific interaction process between the two participants in our protocol:

The client C	The server S
Input $X=\{x_1,...,x_m\}$	Input $Y=\{y_1,...,y_n\}$
(1) Generate (pk_C,sk_C)	
(2) Insert the elements into BF_C	
(3) Inverse the BF_C, IBF_C	
(4) Blind the IBF_C, $BIBF_C$	
(5) Generate the proof $\pi_1=\mathrm{PoK}\{(r_1,...,r_m)\mid\wedge_{i=1}^{m}(BIBF_C[i]=r_i\times IBF_C[i])\}$	
π_1, $BIBF_C=\{\overline{x_1},\overline{x_2},...,\overline{x_m}\}$ →	
	(6) $\forall j, 1\le j\le n$ $h_1(y_j),h_2(y_j),...,h_k(y_j)$
	Find $a_{h_1(y_j)},...,a_{h_k(y_j)}$
	(7) Encrypt the elements $E(\overline{y_j})=\{(g^{z_j},g^{a_{h_1(y_j)}}\cdot y^{z_j}),...,(g^{z_j},g^{a_{h_k(y_j)}}\cdot y^{z_j})\}$
	(8) Generate the proof $\pi_2=\mathrm{PoK}\{(z_1,...z_j)\mid\wedge_{j=1}^{n}(c_j=g^{z_j})\}$
← π_2, $E(\overline{Y})=\{E(\overline{y_1}),...,E(\overline{y_n})\}$	
(9) Decrypt $\forall i$, $1\le i\le n$, $\forall j$, $1\le j\le k$,	

$(g^{z_i})^{-x}(g^{a_{h_j(y_i)}} y^{z_i}) = g^{a_{h_j(y_i)}}$.
If the result of decrypting an item
of the set $E(\overline{Y})$ is all 1, then the counter adds 1.

Fig. 6. Our first PSI-CA protocol

The above protocol uses the Bloom Filter to display the set. However, the Bloom Filter exists the problem of false positive and therefore there remain errors between the calculated results and the true results. We mainly use the Garbled Bloom Filter to solve the problem. Compared to the Bloom Filter, GBF's array holds strings rather than individual characters in each bit, this feature also further enhances the security. We next propose an improved PSI-CA protocol by using Garbled Bloom Filter and the additive homomorphic property of Elgamal algorithm.

Fig. 7 presents the process of improved PSI-CA protocol:

The client C	The server S
Input $X = \{x_1, ..., x_m\}$	Input $Y = \{y_1, ..., y_n\}$
(1) Generate (pk_C, sk_C) $b_1 = H_0(x_1), ..., b_m = H_0(x_m)$ (2) Construct the key-value pairs $\overline{X} = \{(x_1, b_1), ..., (x_m, b_m)\}$ (3) Insert elements into GBF_C, where $b_i = \sum_{j=1}^{t} GBF_C(h_j(x_i))$	
$\xrightarrow{GBF_C, \{h_0, ..., h_t\}}$	
	(4) $\forall j, 1 \le j \le n$, $h_1(y_j), h_2(y_j), ..., h_k(y_j)$ Find $a_{h_1(y_j)}, ..., a_{h_t(y_j)}$ (5) Encrypt the elements $E(\overline{y_j}) = \{(g^{z_j}, g^{a_{h_1(y_j)}} \cdot y^{z_j}), ..., (g^{z_j}, g^{a_{h_t(y_j)}} \cdot y^{z_j})\}$
$\xleftarrow{E(\overline{Y}) = \{E(\overline{y_1}), ..., E(\overline{y_n})\}}$	
(6) Decrypt $\forall i, 1 \le i \le n, \forall j, 1 \le j \le t$ $(g^{z_i})^{-x}(g^{a_{h_j(y_i)}} y^{z_i}) = g^{a_{h_j(y_i)}}$ (7) Compute $\prod_{j=1}^{t} g^{a_{h_j}(y_i)} = g^{\sum_{j=1}^{t} a_{h_j}(y_i)}$, if $g^{\sum_{j=1}^{t} a_{h_j}(y_i)} = g^{b_i}$, then the counter adds 1.	

Fig. 7. The improved PSI-CA protocol

5. Security analysis

We assume that A_C and A_S are the real world adversaries. SIM_C and SIM_S is the corresponding adversaries in the ideal world. The A_C and A_S can corrupt C, SIM_C and S, SIM_S respectively. Let $\overline{C}$ and $\overline{S}$ be the honest party in ideal world. In the real world, the trusted third party generates the public parameter $P=(G,q,g)$. However, in the ideal world, this process is realized by SIM_C and SIM_S. We define the combined output of $C,S,A_C(A_S)$ is $REAL_{\Theta,A_C(Z)(A_S(Z))}(X,Y)$ in real world, and the combined output of $\overline{C},\overline{S},SIM_C(SIM_S)$ is $IDEAL_{f,SIM_C(Z)(SIM_S(Z))}(X,Y)$ in ideal world.

Theorem. If the Elgamal cryptography algorithm is semantically secure, the proof protocol in our scheme is zero-knowledge proof, then our PSI-CA protocol could securely compute the function $f:(X,Y)\to(|X\cap Y|,\perp)$.

Proof. In order to prove the security of the PSI-CA protocol, we consider two cases which the client C is corrupted by A_C firstly and the server S is corrupted by A_S.

Case1. The client C is corrupted by A_C

We let Z be a distinguisher that can control the adversary A_C. The Z feeds the input of the receiver S and sees the output of S. In the real world, Z's view includes A_C's view and S's output. In the ideal world, Z's view includes A_C's view and $\overline{S}$'s output. We need to prove that Z's view in the real world is indistinguishable with the view in the ideal world. Considering a range of games $Game_0$, $Game_1$, $Game_2$, where $Game_{i+1}$ could slightly modify $Game_i(i=0,1)$. Let the probability that Z can successfully distinguish the view in $Game_i$ from the view in the real protocol be $Pr[i]$. And let the S_i be the simulator in $Game_i$.

$Game_0$: This game corresponds to the execution process of the protocol in the real world. And the simulator S_0 has all information about the server S, also, it can interact with A_C. Therefore, it exists $Pr[REAL_{\theta,A_C(Z)}(X,Y)]=Pr[Game_0]$.

$Game_1$: $Game_1$ has the same process as $Game_0$, if the proof π_1 is valid, then the simulator S_1 executes the algorithm for π_1 with the client C to calculate the multiplier $\{r_1,...,r_m\}$. Using $\{r_1,...,r_m\}$, the simulator S_1 builds $X=\{x_1,...,x_m\}$. The simulator S_1 then extract $\{r_1\cdot IBF_C[1],...,r_m\cdot IBF_C[m]\}$ from $BIBF_C$. Then the simulator S_1 computes $IBF_C=\{IBF_C[1]=\frac{BIBF_C[1]}{r_1},...,IBF_C[m]=\frac{BIBF_C[m]}{r_m}\}$ and then BF_C. After that, the simulator S_1 computes $X=\{x_1,...,x_m\}$. Since the simulation soundness property of the proof π_1, Z's views in $Game_0$ and $Game_1$ are indistinguishable. Thus, $|Pr[Game_1]-Pr[Game_0]|\le\theta_1(k)$, where $\theta_1(k)$ is a negligible function.

$Game_2$: The first few steps of $Game_2$ are exactly the same as those of $Game_1$. The only difference is that after constructing the set $X=\{x_1,...,x_m\}$, the simulator S_2 performs the steps as followed.

(1) calculate $|X\cap Y|$;

(2) construct the set $Y'=\{y_1',...,y_n'\}$, where the set Y' contains $|X\cap Y|$ random elements from the set X and $n-|X\cap Y|$ random elements from G;

(3) using the set X, construct $BIBF_C$;

(4) calculate $E(\overline{Y'}) = \{E(\overline{y_1'}),...,E(\overline{y_n'})\}$, where $E(\overline{y_j'}) = \{(g^{z_j}, g^{a_{h_1(y_j')}} \cdot y'^{z_j}),...,(g^{z_j}, g^{a_{h_k(y_j')}} \cdot y'^{z_j})\}$;

(5) sends $E(\overline{Y'}) = \{E(\overline{y_1'}),...,E(\overline{y_n'})\}$ as $E(Y) = \{E(y_1),...,E(y_n)\}$ and simulates the proof π_2;

Because the related Elgamal encryption scheme is semantically secure, the distributions of $E(\overline{Y'}) = \{E(\overline{y_1'}),...,E(\overline{y_n'})\}$ in $Game_1$ and $Game_2$ are identical. Also, since the zero-knowledge simulatability of the proof π_2 and indistinguishability of $\langle E(\overline{Y'}) = \{E(\overline{y_1'}),...,E(\overline{y_n'})\}, \{y_1',...,y_n'\}\rangle$ in $Game_1$ and $Game_2$. Z's views in $Game_1$ and $Game_2$ are indistinguishable. Therefore, it exists $\left|Pr[Game_2]\text{-}Pr[Game_1]\right| \le \theta_2(k)$, where $\theta_2(k)$ is a negligible function.

In the real world, the adversary SIM_C could simulate the honest party S and includes all steps from $Game_2$. The execution of the protocol in the real world is as follows:

(1) Firstly, SIM_C generates the public parameter $P = (G, q, g)$. Next, SIM_C invokes A_C, and input $X = \{x_1,...,x_m\}$;

(2) After receiving π_1 and $\overline{X} = \{\overline{x_1}, \overline{x_2},...,\overline{x_m}\}$, where $\overline{x_i} = r_i \cdot IBF_C(x_i)$. SIM_C verifies the validity of the proof π_1. If the verification succeeds, SIM_C hashes every element from the set Y, and finds $a_{h_1(y_j)},...,a_{h_k(y_j)}$ from the set $\overline{X} = \{\overline{x_1}, \overline{x_2},...,\overline{x_m}\}$;

(3) SIM_C sends X and $\overline{S}$ sends Y to the trusted third party T, T uses X and Y as input and computes the functionality f, returns $|X \cap Y|$ to SIM_C.

(4) After SIM_C receiving $|X \cap Y|$, the simulator performs the following operations:

(i) SIM_C builds $\overline{Y} = \{\overline{y_1},...,\overline{y_n}\}$, where $\overline{Y} = \{\overline{y_1},...,\overline{y_n}\}$ contains $|X \cap Y|$ random elements from the set X and $n - |X \cap Y|$ random elements from the group G;

(ii) using the set X, constructs $BIBF_C$;

(iii) computes $E(\overline{Y''}) = \{E(\overline{y_1''}),...,E(\overline{y_n''})\}$, where $E(\overline{y_j''}) = \{(g^{z_j}, g^{a_{h_1(\overline{y_j})}} \cdot y^{z_j}),...,(g^{z_j}, g^{a_{h_k(\overline{y_j})}} \cdot y^{z_j})\}$;

(iv) sends $E(\overline{Y''}) = \{E(\overline{y_1''}),...,E(\overline{y_n''})\}$ as $E(\overline{Y'}) = \{E(\overline{y_1'}),...,E(\overline{y_n'})\}$ and simulates the proof π_2.

Hence, the simulator SIM_C provides A_C the same simulation as the simulator S_2 in $Game_2$. Therefore, it has $Pr[IDEAL_{f,SIM_C(Z)}(X,Y)] = Pr[Game_2]$ and

$$\left|Pr[IDEAL_{f,SIM_C(Z)(X,Y)}] - Pr[REAL_{\theta,A_C(Z)}Game_2]\right|$$
$$\left|Pr[Game_2] - Pr[Game_0]\right|$$
$$\le \sum_{i=0}^{1}(\left|Pr[Game_{i+1}] - Pr[Game_i]\right|)$$
$$\le \theta_2(k) + \theta_1(k) = \theta(k),$$

where $\theta(k)$ is a negligible function. So, it exists

$$IDEAL_{f,SIM_C(z)}(X,Y) \overset{c}{\equiv} REAL_{\theta,A_C(z)}(X,Y) \tag{5}$$

where $\overset{c}{\equiv}$ means computationally indistinguishable.

The proof process of the case which A_S corrupts S is similar as the first case, therefore, the proof process is not described in detail.

6. Efficiency

6.1 Implementation details

We run our experiments on a laptop with an Intel i5-8300H 2.30Ghz, 8GB RAM, and Ubuntu 18.04.4 system. We have performed the PSI-CA protocols in C++. We choose the set element size of 128 bits, the safety parameter λ=40, and the computational security parameter κ=40.

6.2 Performance analysis

We choose the protocols proposed by Nan CHENG et al. [27] and Li H et al. [28] to compare the computation cost with our PSI-CA protocol. And the set sizes of 2^{10}, 2^{11}, 2^{12}, 2^{13} and 2^{14} are selected for the five cases. We separately test the time cost by different participants to execute the protocol. A comparison of the overall computation cost of our PSI-CA protocol with other representative protocols can be obtained as shown in the following **Table 2**:

Table 2. Computational cost comparison

Protocols	The party	The size of the set				
		2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
Li H [28]	P_1	2.79	4.26	7.63	15.96	32.85
	P_2	0.98	2.05	4.52	8.30	17.43
Nan CHENG [27]	P_1	1.86	3.76	7.42	13.5	31.65
	P_2	0.75	1.98	3.80	7.51	16.14
Ours	P_1	3.27	4.64	9.25	17.41	36.32
	P_2	0.87	2.10	4.45	8.41	18.32

Fig. 8 and **Fig. 9** show the time cost of the first protocol changes as the cardinality of the set held by each participant increases.

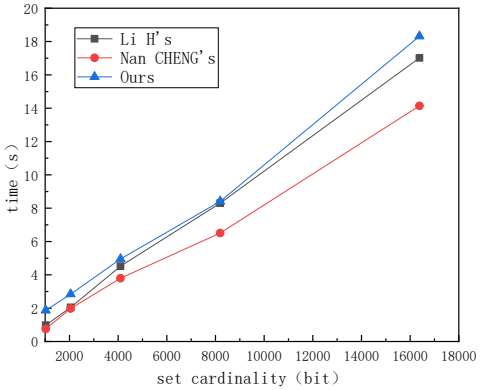

Fig. 8. Time cost of P_1

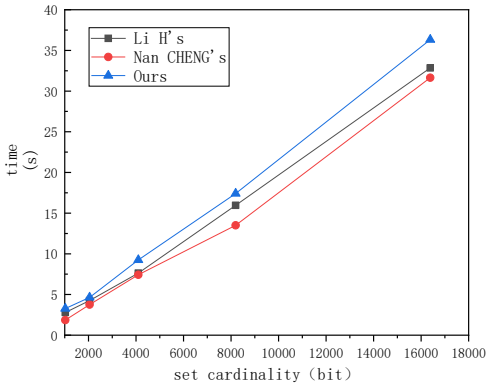

Fig. 9. Time cost of P_2

We can conclude that as the sets cardinality increases, the time cost by each participant in all three schemes increases linearly. Our scheme requires a slightly higher time cost than the other two schemes, but it can resist malicious behaviors by the malicious participants. And the interactive zero-knowledge proof protocol used in our scheme is bound to incur additional time cost. Therefore, the reader needs to make a reasonable tradeoff between security and time cost for this purpose.

In addition, we have also conducted comparative experiments on the accuracy of the results computed by the two protocols proposed in this paper under different conditions. The set size held by each participant is 2^{10}. The experimental results are shown in **Table 3**.

Table 3. Comparison of the accuracy of the results of the two protocols

		Size of the set					Mean relative error $\frac{1}{n}\cdot\sum_{j=1}^{n}\frac{\lvert R_j - I_j\rvert}{I_j}$
The real value (I)		2^2=4	2^4=16	2^6=64	2^8=256	2^9=512	
The first protocol	Results (R)	4	13	62	250	504	5.156%
		5	15	60	251	510	7.980%
The improved protocol	Results (R)	4	16	63	255	509	0.507%
		4	15	64	254	511	1.445%

Through two comparative experiments on the two protocols, it is not difficult to find that the mean relative error of the improved protocol's calculation results is smaller than the former, resulting in more accurate calculation results.

Also, we select the scheme of Lv S et al. [3], Nan CHENG et al. [27] for the functional comparison with our first protocol. In order to ensure fairness, our selected protocols are all local two-party PSI-CA protocols in recent years. The selected properties are: security model, the adversary model, security assumption, etc. We present the functional comparison results in **Table 4**. In recent years there has been less researches on PSI-CA protocols under the malicious model, and most PSI-CA protocols are based on the semi-honest model. **Table 4** shows that our protocol is secure under the standard model, and the adversary model used is malicious model, based on DDH assumption, and the set size is hidden. Therefore, our protocol

has higher practical application value compared with other protocols.

Note: the explanation of the meaning of abbreviations in the table. STD: Standard Model. ROM: Random Oracle Model. DDH: Decisional Diffie-Hellman assumption. ECDLP: Elliptic Curve Discrete Logarithm Problem.

Table 4. Function comparison

Property	Ours	Nan CHENG [27]	Li H [28]	Lv S [3]
Security Model	STD	STD	STD	ROM
Adversary Model	The malicious model	The semi-honest model	The semi-honest model	The semi-honest model
Security Assumption	DDH	DCRA	ECDLP	--
Set Size Hidden	√	√	√	√
Resist Malicious Behavior	√	×	×	×

7. Conclusion

In this paper, we come up with two local two-party PSI-CA protocol, and prove the security of the first protocol under the malicious model. Our schemes solve the main problem of most protocols in the PSI-CA research field being unable to resist malicious behaviors. And we innovatively utilize Garbled Bloom Filter to improve the accuracy of intersection cardinality. Meanwhile, the performance of our protocol is analyzed and compared functions with other protocols to demonstrate the excellence and practicality of our protocol. In the future research, we will consider how to refine our improved protocol to make it more practical for security under the malicious model.

References

[1] A. C. Yao, "Protocols for secure computations," in *Proc. of 23rd Annual Symposium on Foundations of Computer Science*, pp. 160-164, 1982. Article (CrossRef Link)

[2] GAO Ying, WANG Wei, "A Survey of Multi-party Private Set Intersection," *Journal of Electronics & Information Technology*, vol. 45, no. 5, pp. 1859-1872, 2023. Article (CrossRef Link)

[3] Lv S, Ye J, Yin S, "Unbalanced private set intersection cardinality protocol with low communication cost," *Future Generation Computer Systems*, vol. 102, pp. 1054-1061, 2020. Article (CrossRef Link)

[4] MEZZOUR G, PERRIG A, GLIGOR V D, "Privacy-Preserving Relationship Path Discovery in Social Networks," in *Proc. of International Conference on Cryptology and Network Security*, pp. 189-208, 2009. Article (CrossRef Link)

[5] SHEN L, CHEN X, WANG D, "Efficient and Private Set Intersection of Human Genomes," in *Proc. of the 2018 IEEE International Conference on Bioinformatics and Biomedicine*, pp. 761-764, 2018. Article (CrossRef Link)

[6] OU R, HAO M, "Efficient Private Set Intersection Using Point-Value Polynomial Representation," *Security and Communication Networks*, vol. 2020, no. 1, pp. 1-12, 2020. Article (CrossRef Link)

[7] FREEDMAN M J, NISSIM K, PINKAS B, "Efficient Private Matching and Set Intersection," in *Proc. of International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 1-19, 2004. Article (CrossRef Link)

[8] PINKAS B, ROSULEK M, TRIEU N, "SpOT-Light: Lightweight Private Set Intersection from Sparse OT Extension," in *Proc. of 39th Annual International Cryptology Conference*, vol. 11694, pp. 401-431, 2019. Article (CrossRef Link)
[9] SONG X, GAI M, ZHAO S, "Privacy-Preserving Statistics Protocol for Set-Based Computation," *Journal of Computer Research and Development*, vol. 57, no. 10, pp. 2221-2231, 2020. Article (CrossRef Link)
[10] VOS J, CONTI M, ERKIN Z, "Fast multi-party private set operations in the star topology from secure ANDs and ORs," *Cryptology ePrint Archive*, 2022.
[11] Zhang Lei, He Chongde, Wei Lifei, "Efficient and Malicious Secure Three-Party Private Set Intersection Computation Protocols for Small Sets," *Journal of Computer Research and Development*, vol. 59, no. 10, pp. 2286-2298, 2022. Article (CrossRef Link)
[12] Ben-Efraim A, Nissenbaum O, Omri E, "PSImple: Practical multiparty maliciously-secure private set intersection," in *Proc. of the 2022 ACM on Asia Conference on Computer and Communications Security*, pp. 1098-1112, May, 2022. Article (CrossRef Link)
[13] Abadi A, Terzis S, Dong C, "O-PSI: delegated private set intersection on outsourced datasets" in *Proc. of ICT Systems Security and Privacy Protection: 30th IFIP TC 11 International Conference*, pp. 3-17, 2015. Article (CrossRef Link)
[14] Abadi A, Terzis S, Dong C, "VD-PSI: verifiable delegated private set intersection on outsourced private datasets" in *Proc. of Financial Cryptography and Data Security: 20th International Conference*, pp. 149-168, 2017. Article (CrossRef Link)
[15] YANG X, LUO X, WAN X A, "Improved outsourced private set intersection protocol based on polynomial interpolation," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 1, 2017. Article (CrossRef Link)
[16] Wei LF, Wang Q, Zhang L, Chen CC, Chen YJ, Ning JT, "Efficient Private Set Intersection Protocols with Semi-trusted Cloud Server Aided," *Journal of Software*, vol. 34, no. 2, pp. 932-944, 2023. Article (CrossRef Link)
[17] Egert R, Fischlin M, Gens D, "Privately computing set-union and set-intersection cardinality via bloom filters," in *Proc. of Information Security and Privacy: 20th Australasian Conference*, pp. 413-430, 2015. Article (CrossRef Link)
[18] Ion M, Kreuter B, Nergiz E, "Private intersection-sum protocol with applications to attributing aggregate ad conversions," *Cryptology ePrint Archive*, 2017.
[19] Davidson A, Cid C, "An efficient toolkit for computing private set operations," in *Proc. of Information Security and Privacy: 22nd Australasian Conference*, pp. 261-278, 2017. Article (CrossRef Link)
[20] DEBNATH S K, DUTTA R, "Efficient Private Set Intersection Cardinality in the Presence of Malicious Adversaries," in *Proc. of the International Conference on Provable Security*, pp. 326-329, 2015. Article (CrossRef Link)
[21] DEBNATH S K, DUTTA R, "Secure and Efficient Private Set Intersection Cardinality Using Bloom Filter," in *Proc. of the International Conference on Information Security*, pp. 209-226, 2015. Article (CrossRef Link)
[22] ElGamal T, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469-472, 1985. Article (CrossRef Link)
[23] BLOOM, BURTON H, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422-426, 1970. Article (CrossRef Link)
[24] TARKOMA S, ROTHENBERG C E, LAGERSPETZ E, "Theory and Practice of Bloom Filters for Distributed Systems," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 131-155, 2012. Article (CrossRef Link)
[25] WEI L, LIU J, ZHANG L, "Survey of Privacy Preserving Oriented Set Intersection Computation," *Journal of Computer Research and Development*, vol. 59, no. 8, pp. 1782-1799, 2022. Article (CrossRef Link)
[26] DWIVEDI A D, SINGH R, GHOSH U, "Privacy preserving authentication system based on non-interactive zero knowledge proof suitable for Internet of Things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 10, pp. 4639-4649, 2022. Article (CrossRef Link)

[27] Nan CHENG, Yun-Lei ZHAO, "Efficient Approach Regarding Two-Party Privacy-Preserving Set Union/Intersection Cardinality," *Journal of Cryptologic Research*, vol. 8, no. 2, pp. 352-364, 2021.
[28] Li H, Gao Y, "Efficient Private Set Intersection Cardinality Protocol in the Reverse Unbalanced Setting," in *Proc. of Information Security: 25th International Conference*, pp. 20-39, 2022. Article (CrossRef Link)
[29] Chandran N, Dasgupta N, Gupta D, "Efficient Linear Multiparty PSI and Extensions to Circuit/Quorum PSI," in *Proc. of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1182-1204, 2021. Article (CrossRef Link)
[30] Davi Resende A C, de Freitas Aranha D, "Faster unbalanced Private Set Intersection in the semi-honest setting," *Journal of Cryptographic Engineering*, vol. 11, pp. 21-38, 2021. Article (CrossRef Link)

Jingjie Liu received the bachelor's degree from Northwest University, Xi'an, China, in 2022. She is currently working toward the master's degree in computer science and technology at Northwest Normal University. Her main research interests include information security, secure multi-party computing and cryptography.

Suzhen Cao received master's degree from Lanzhou Jiaotong University, Lanzhou, China, in 2010. Currently, she is an associate professor at the College of Computer Science and Engineering of Northwest Normal University. Her main research interests inlcude information security, privacy preserving and cryptography.

Caifen Wang received the master's degree in computational mathematics from Lanzhou University, Lanzhou, China in 1998, the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2003. She is a Professor at the College of Big Data and Internet of Shenzhen Technology University. Her main research interests include network and information security and cryptography.

Chenxu Liu received the bachelor's degree from Nanjing Forestry University, Nanjing, China, in 2021. He is currently working toward the master's degree in computer technology at Northwest Normal University, Lanzhou, China. His main research include information security and attribute-based encryption.