Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

A novel privacy protection scheme for internet of things based on blockchain and privacy set intersection technique

Qian Zhou, Chengzhe Lai*, Qili Guo, Haoyan Ma and Dong Zheng

Abstract

In the era of big data, an ocean of data generated by Internet of Things (IoT) devices will be analyzed and processed by cloud computing. However, outsourcing of data can lead to leakage of user privacy to those unreliable service providers. In this paper, we propose a novel privacy-preserving scheme for IoT device by employing privacy set intersection (PSI) and blockchain technique to achieve data privacy. First, a homomorphic encryption PSI technique based on 0-1 encoding is proposed, which well hides the set base to ensure data privacy. Second, combining blockchain structure and smart contract, the proposed scheme can improve the efficiency of data sharing by storing the shared data on a blockchain. Third, the security analysis shows that the scheme has extremely high control over the individual data and can ensure the security and privacy of the data. Finally, we compare the functionality with other relevant schemes and demonstrate that our scheme functions well with low communication and computational overhead.

Keywords: Privacy, PSI, Blockchain, Smart contracts, Internet of Things, Homomorphic encryption

Introduction

As the IoT and blockchain industries continue to grow [1], they bring convenience to people's lives. However, there is a great possibility of illegal access by stakeholders when IoT platforms, cloud computing infrastructures [2, 3] and smart devices are exchanging huge amounts of data [4]. Nowadays, most of our data is stored on the cloud and third-party data service providers to transfer the data. However, cloud storage is subject to various security threats such as malware, man-in-the-middle attacks, and sensitive data attacks [5]. In addition, users who want to use cloud storage have only few options to select a good and inexpensive data provider, and users are unable to participate in the supervision of data. Therefore, the security and privacy of sharing data in IoT has attracted great attention in academic and industry.

In recent years, blockchain technology has been widely used as a decentralized data storage system. By removing all central servers and achieving peer-to-peer interaction between all network nodes [6] , it can provide a solution for the storage and sharing of user data in IoT with traceability, tamper-proof, and unforgeability. Blockchain-based privacy protection for IoT will result in a significant improvement in data security, where users' data are recorded in a decentralized ledger and it is difficult for hackers to tamper with the ledger to overwrite the existing data [7, 8]. Blockchain provides transparency by allowing people with access to track past transactions that have occurred on the chain, which is a great tool for sharing user data.

Privacy Set Intersection (PSI) is a technique that allows secret sharing and encryption of data and does not reveal any data information during the computation process. In PSI technique, two users can obtain the intersection part, which can realize the sharing of user data. Therefore, PSI and blockchain can complement each other to some extent, and the existing computational research based on blockchain and PSI mainly designs privacy protection schemes

*Correspondence: lcz_xupt@163.com

School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an, China

for specific application scenarios, such as smart grid [9], medical data [10], etc. However, most of the existing PSI techniques are not efficient and users cannot store their data securely, especially when stored and computed on cloud servers, which cannot guarantee the security of user datasets. Moreover, PSI techniques on IoT need to consider computational complexity and efficiency to be applied in practical problems.Therefore, it is crucial to combine blockchain and PSI technique for data security, storage and sharing when it comes to privacy protection in IoT.

Related Work

We review related work through two aspects: 1) the design of the PSI protocol and 2) blockchain-based PSI.

Design of PSI Protocol

Firstly, PSI technique is a special application within the field of secure multi-party computing. The application scenario is that the data of the participants can be represented as a set, and the intersection of the incoming and outgoing sets can be computed collaboratively for data sharing without revealing the data of the respective participants. As a result, PSI techniques have received a lot of attention. PSI computation was proposed by Freedom et al. [11] in 2004 and was implemented with the help of inadvertent polynomial valuation and homomorphic encryption. However, the efficiency is low and the computational cost is high. Cristofaro and Tsudik [12] proposed a PKC-PSI protocol based on blind RSA, which allowed a great extension of the protocol in terms of the number of elements. In 2015, Debnath and Dutta [13] proposed a PSI, PSI base and certified PSI protocol based on multiplicative homomorphic PKC and Bloom filter [14]. In 2016, Freedman et al. [15] extended the approach of [11] by proposing a PSI protocol with linear communication and computational overhead, demonstrating in a malicious adversary model formal simulation-based security proofs and evaluate the practical efficiency of the proposed PSI protocol. In 2016, Abadi et al. [16] proposed an additive homomorphic PKC based on a protocol in which clients represent their datasets and independently as blind polynomials before encrypting them. In 2018, in the literature, Linming Gong et al. [17] proposed a homomorphic encryption based scheme for generalized secure two-sided comparisons using the millionaire problem extended to fractions for comparison. Combining the advantages of public key encryption PSI, Chen et al. [18] proposed a PSI protocol based on RLWE homomorphic encryption.

In 2012, Huang et al. [19] proposed several Boolean circuit-based PSI with significant improvement in data scaling. Zahur et al. [20] proposed a new approach to produce less interference than any of the current schemes. In 2018, Pinkas et al. [21] further optimized the circuit-based PSI protocol as a way to fight against semi-honest adversaries. Ciampi et al. [22] gave another PSI protocol based on two-party secure computation and this protocol possesses better performance than the scheme given by Pinkas et al.

Dong et al. [23] proposed a PSI protocol using Bloom filters and OT expansion protocol. The constructed protocol has the ability to operate on billion-scale ensembles and can be shown to be secure under the semi-honest and malicious models. However, Rindal et al. proposed the possibility of attacks on PSI protocols using Bloom filters under the malicious model [24, 25] and showed how to improve the existing protocols using Bloom filters to give the first PSI protocol that is secure under the malicious model [26]. In 2014, Pinkas et al. [27] optimized the semi-honest adversary version.

Blockchain-based PSI

As a good distributed ledger, blockchain is widely used in the scenarios of data sharing and protection. In literature [28], blockchain as a framework of deep learning, a deep learning framework Deepchain is proposed to ensure data privacy and auditability. In literature [29], as a decentralized architecture, blockchain proposes a decentralized framework based on blockchain, CrowdBC, so that users' privacy can be guaranteed. In literature [30], an application based on blockchain framework is designed and traded.

Due to the nature of blockchain, the issue of blockchain-based PSI has received a lot of attention. In the literature [31], the authors compare the differences between using blockchain and smart contract technologies and not using these two technologies. The results show that data integrity, better security and privacy are guaranteed in systems using blockchain technology and smart contracts. In the literature [32], a blockchain smart contract based approach is proposed for sharing IoT devices between the system and the user and the ownership of the device is continuously transferred and the smart contract bridges the information of the new owner with the public key, and finally the data released from the IoT device is kept private. In literature [33], Zhu et al. proposed a blockchain smart contract execution system based on secure multi-party computation, in which a smart contract framework based on secure multi-party computation and a secure multi-party protocol based on secret sharing are designed to standardize the execution process of smart contracts and ensure the privacy of input and correctness of computation in smart contracts. In the literature [34], the multi-party secret set intersection protocol and application generates public and private keys by transforming the privacy-secret set into a 0-1 vector using NTRU homomorphic encryption, and then encrypts the vector using the public key and sends it to the cloud server.

In literature [35], Chen et al. proposed a privacy set intersection problem based on obfuscated circuits in combination with blockchain, referring to YAO's universal obfuscated circuit valuation technique, and the computed ciphertext is simultaneously disclosed to each participant after combining with smart contracts, but the protocol is less efficient and occupies more memory. In the literature [36], Xiong et al. based on the blockchain privacy-preserving intersection algorithm BPSI, which can avoid the assumption of trust in cloud computing centers while providing high computational efficiency. The schemes proposed in the literature [37] and [38] first tried the *Gö* del random number and ELGamal encryption algorithms to construct a ranking protocol for confidential databases. The literature [39], proposed a blockchain-based data query scheme that uses secret sharing and smart contract design to achieve user control over the data.

Contribution

In this paper, we propose a blockchain-based privacy protection scheme for the IoT. The scheme combines blockchain and PSI to achieve private computing and sharing for both parties of the set. Therefore, the scheme can improve the security and efficiency of the blockchain nodes, and the results are saved on the blockchain to ensure that the data can be securely stored and shared.

The main contributions of this paper are as follows:

- First, we propose a homomorphic encryption PSI technique based on 0-1 coding. This technique uses 0-1 encoding to process the user's initial data and homomorphic encryption of each bit for interactive computation. Compared with the most of existing techniques, our PSI technique can pre-process complex data, and the encrypted ciphertext can be stored on the cloud server. As a result, the computational and communication overheads are reduced.
- Second, we design a blockchain-based privacy protection scheme for IoT. The scheme combines blockchain and PSI technique to achieve privacy protection and sharing. At the beginning of the scheme, we use smart contracts to ensure secure interaction between the two parties. Compared with most schemes, our scheme uses blockchain instead of cloud servers to store and share data, which improves security and efficiency.

Organization

The rest of this paper is organized as follows. In Preliminaries section, we describe the relevant techniques used in the scheme. In Homomorphic Encryption PSI Technique based on 0-1 Encoding section, we propose a homomorphic encryption PSI technique based on 0-1 encoding and perform correctness analysis. In Proposed Scheme section, we design a blockchain technique-based privacy protection scheme for IoT and perform correctness analysis. In Security Analysis section, we analyze the security of the proposed technique and scheme. In Performance Analysis section, we evaluate the performance of the scheme and compare it with other schemes. Finally, a conclusion of the scheme is drawn.

Preliminaries

In this section, we will introduce several preliminary tools used in this scheme.

Security Model for Secure Multiparty Computation

During the execution of a secure multi-party computation protocol, semi-honest participants retain valid information during the execution of the protcocol while faithfully performing it and try to deduce hidden information about other participants. If the participants in a secure multi-party computation protocol are all semi-honest participants, the computational model used in the protocol is said to be a semi-honest model. Since the computational models in the protocols in this paper are all semi-honest, models, the definition of the secure type of the two-party computational model under the semi-honest model is given below.

Let Alice and Bob have a protocol to compute a function f by private data as x, y, and π, respectively. And both parties want to compute the function F: $(x, y) \rightarrow (f_1(x, y), f_2(x, y))$ by cooperating without disclosing their respective private data, where there exists a probabilistic polynomial function f: $\{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^* \times \{0,1\}^*$, $f_1(x, y)$ and $f_2(x, y)$ are the two components of the resulting function F computed by Alice and Bob, respectively. The sequence of messages obtained by Alice during the execution of the protocol is denoted as $view_1^{\pi}(x, y)$, and similarly the sequence of messages obtained by Bob is denoted as $view_2^{\pi}(x, y)$, and the resulting outputs are denoted as $output_1^{\pi}(x, y)$ and $output_2^{\pi}(x, y)$, respectively.

Definition 1 For a function f, if there exist probability polynomials S_1 and S_2 satisfying Eqs. (1) and (2), it is said to be a π confidential computational function.

$$\{S_1(x, f_1(x, y)), f_2(x, y)\}_{x,y} \stackrel{c}{\equiv} \{view_1^{\pi}(x, y), output_2^{\pi}(x, y)\}_{x,y} \quad (1)$$

$$\{f_1(x, y), S_2(x, f_2(x, y)\}_{x,y}) \stackrel{c}{\equiv} \{output_1^{\pi}(x, y), view_2^{\pi}(x, y)\}_{x,y} \quad (2)$$

where $\stackrel{c}{\equiv}$ denotes computational indistinguishability. To prove that a secure multi-party computation protocol is secure, it is necessary to construct simulator S_1 and S_2 such that (1) and (2) hold.

Homomorphic Encryption

The notion of homomorphic encryption was introduced by Rivest [40], and its special properties make it possible to perform some operations directly on the ciphertext without decrypting it. Sander [41] defines additive and multiplicative homomorphic encryption over the ring of integers.

- Additive homomorphic encryption: If the PKC scheme ***(KeyGen, En, Dec)*** is additive homomorphic, then for any plaintext (m_1, m_2) and also any private key pair (sk, pk), there exists.

$$Enpk(m_1 + m_2) = Enpk(m_1) \times Enpk(m_2) \quad (3)$$

The additive homomorphic PKC scheme has multiplicative properties:

$$Enpk(m_1 \times m_2) = (Enpk(m_1))m_2 \quad (4)$$

Pailliar Algorithm

Pailliar algorithm is a basic probabilistic encryption algorithm and Pailliar encryption algorithm has additive homomorphism [42].

- ***KeyGen*() → (*pk*, *sk*)** Two independent large prime numbers p and q are randomly selected, which satisfy $gcd(pq, (p-1)(q-1)) = 1$, calculate $n = pq$, $\lambda = LCM(p-1, q-1)$, and randomly select $g \in Z^*_{n^2}$. In this case, the public key $pk = (n, g)$ and the private key $sk = (\lambda)$.
- ***En*(*pk*, *m*) → *c*** The ciphertext $c = g^m r^n (mod n^2)$ is calculated by randomly selecting r$\in Z^*_n$.
- ***Dec*(*sk*, *c*) → *c*** the function:

$$L(x) = \frac{x-1}{n} \quad (5)$$

To caculate:

$$m = \frac{L(c^{\lambda} mod n^2)}{L(g^{\lambda} mod n^2)} mod n \quad (6)$$

Bloom Filter

A Bloom filter is a very common query operation in software development [14] by querying whether an element belongs to a certain set or not. First define the parameters of the Bloom filter - according to the agreed capacity n of the two institutions A and B and the error rate p. Then calculate the length m of the Bloom filter and the number of hash functions k.

$$k = \frac{m}{n} ln2 \quad (7)$$

$$m = -\frac{nlnp}{(ln2)^2} \quad (8)$$

The probability of error has the following formula:

$$P = [1 - (1 - \frac{1}{m})^{nk}]^k \approx (1 - e^{-\frac{kn}{m}})^k \quad (9)$$

The structure of the Bloom filter is shown in Fig. 1.

Homomorphic Encryption PSI Technique based on 0-1 Encoding

In this section, we introduce the homomorphic encryption PSI technique based on 0-1 encoding.

Description

As shown in Fig. 2, the homomorphic encryption PSI technique based on 0-1 encoding consists of three stages: date initialization, date processing and date sharing.

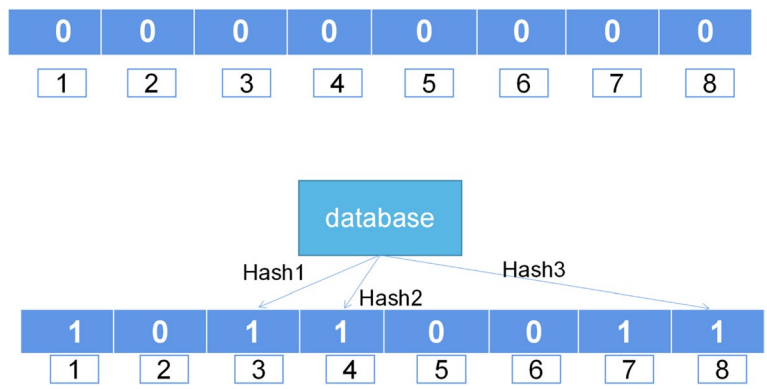

Fig. 1 Structure of bloom filter

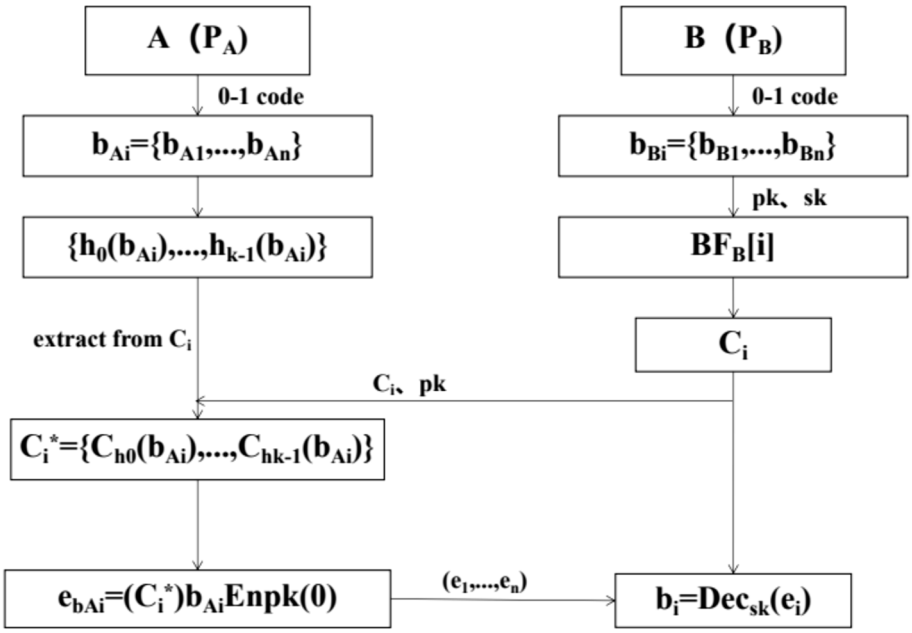

Fig. 2 Protocol procedure

1 **Data Initialization**: In this stage, we set participant A to hold data set P_A and participant B to hold data set P_B. Then, the sets of A and B are encoded 0-1, assuming that the participating parties have sets S_1, $S_2 \subseteq \{a_1, a_2, ..., a_n\}$=U, where U is the full-order set. When one of the parties encodes its set $S_i = \{s_1, s_2, ..., s_n\}$ as a new vector $b_i = \{b_1, ..., b_n\}$, where if b_i=1, then $S_i \in U$, if $b_i = 0$, then $S_i \notin U$. After this stage, the vectors involved are all of length n, which can well hide the length of the set.

2 **Data Processing**: In this stage, data processing is divided into the following four steps: key generation, Bloom filter construction, hash function generation and intersection caculation. The specific steps are described as follows: *Step 1: Key Generation.* For the set b_{Bi} obtained after the initial setup, Pailliar encryption is used, and then the public key *pk* and the private key *sk* are generated. *Step 2: Bloom filter Construction.* For the set b_{Bi} obtained Bloom filter, it is necessary to choose the appropriate values of k and m. The obtained Bloom filter is denoted as $BF_B[i]$. The $BF_B[i]$ is encrypted with the public key *pk* described above to obtain C_i, which satisfies:

$$C_i = En_{pk}(BF_B[i]) \tag{10}$$

Step 3: Hash function Generation. Use k hash functions to perform the calculation for the b_{Ai} set. The procedure is as follows:

$$b_{A_i} = \{b_{A_1}, ..., b_{A_n}\} \rightarrow \{h_0(b_{A_i}), ..., h_{k-1}(b_{A_i})\} \tag{11}$$

Step 4: Intersection Calculation. Use C_i and its public key *pk* obtained after B's calculation to send to A, in which A extracts C_i^*, satisfying the following:

$$C_i^* = \{C_{h_{0(b_{A_i})}}, ..., C_{h_{k-1(b_{A_i})}}\} \tag{12}$$

When A obtains C_i^*, the following operation is performed on C_i^* to obtain the desired $e_{b_{A_i}}$.

$$e_{b_{A_i}} = (C_i^*)^{b_{A_i}} En_{pk}(0) \tag{13}$$

3 **Data Sharing**: The A gets $e_{b_{A_i}}$ and sends (e_1,...,e_n) to B. The B receives (e_1,...,e_n) and decrypts e_i using the private key *sk* to obtain:

$$b_i = Dec_{sk}(e_i) \tag{14}$$

The obtained b_i is the base of $A \cap B$. It is because the representation and operations are performed with a fixed set of full order, so if $b_i = 1$, $b_i \in A \cap B$ and vice versa $s_i \notin A \cap B$.

Correctness Analysis

First, prove that $e_{b_{Ai}}$ and b_i in the PSI technique process:

$$\begin{aligned} e_{b_{A_i}} &= (C_i^*)^{b_{A_i}} Enpk(0) \\ &= \{C_{h_0}(b_{A_i}) r_{i,0}^n mod n^2, ..., C_{h_{k-1}}(b_{A_i}) r_{i,k-1}^n mod n^2\} \\ &= \{Enc_{pk}(BF_B[h_0(b_{A_i})]) Enc_{pk}(0), ..., Enc_{pk}(BF_B[h_{k-1}(b_{A_i})]) Enc_{pk}(0)\} \\ &= \{Enc_{pk}(BF_B[h_0(b_{A_i})]) + 0, ..., Enc_{pk}(BF_B[h_{k-1}(b_{A_i})]) + 0\} \\ &= \{Enc_{pk}(BF_B[h_0(b_{A_i})]), ..., Enc_{pk}(BF_B[h_{k-1}(b_{A_i})])\} \end{aligned} \tag{15}$$

where r is a random number that plays a crucial role in protecting the privacy of the data.

$$\begin{aligned} b_{B_i} \cap b_{A_i} &= b_{B_i} \times b_{A_i} \\ &= (b_{B_1}, ..., b_{B_i}) \times (b_{A_1}, ..., b_{A_i}) \\ &= (b_{B_1} \times b_{A_1}, ..., b_{B_i} \times b_{A_i}) \end{aligned} \tag{16}$$

$$\begin{aligned} b_i &= Dec(e_i) \\ &= Dec_{sk}(Enc_{pk}(BF_B[h_{k-1}(b_{A_i})])) \\ &= BF_B[h_{k-1}(b_{A_i})] \end{aligned} \tag{17}$$

The intersection part is obtained where 1 is shown in the result, and the number of 1s is the base of the intersection of the two sets. Therefore, this PSI technique is correct.

Proposed Scheme

In this section, we propose a scheme that combines 0-1 encoded homomorphic cryptographic PSI with blockchain to achieve security and privacy protection during data interaction. The notations used in this scheme are listed in Table 1.

Overview

Our scheme system model includes: receiver and sender, smart contract, blockchain, and other relevant information. The system model is shown in the Fig. 3.

Blockchain and smart contract play an indispensable role in the system model. In our scheme, the privacy information of both parties' data is well protected. First, the information on the blockchain is immutable. Once the information is verified and uploaded to the blockchain, it will be stored permanently. Moreover, the stability and reliability of blockchain data are very high, and it is not easy to cause data leakage, which makes the combination of PSI technique and blockchain have very good security. In addition, if one party does not follow the rules, there is a multiplication mechanism in place. Act as an intermediate manager during the project. Finally, the scheme invokes smart contract to make rules. The smart contract was used to ensure that both parties could abide by the agreed rules during data interaction, making the scheme more secure and reliable. Therefore, the process of data uploading to the blockchain in this scheme has high security and confidentiality.

There are two parties in the system model: the receiver and the sender. The subset of users is selected from a fixed domain and needs to be stored in our cloud servers. For example, in a large IoT environment. a) In the power grid, when user information is shared or the power grid is blacklisted with other companies for user queries, the user's blacklist is a subset of the total list, so it has a full set collection for queries and during the query process, it does not disclose any privacy of the people on the list and can store this blacklist in the cloud server for the next query. b) In connected car dating, if a user wants to select friends with the same hobbies or interests, he/she can use this PSI technique to input the user's attribute set and store it in the cloud server, which can be sent to other people for friend selection at any time, without revealing any private information about him/herself in the process. c) In smart healthcare, the privacy protection of medical data is very important, so the data needs to be secure when sharing medical data. When entering information about a patient's disease, symptoms and treatment, it is selected from a fixed domain and stored on the system for future treatment and information sharing with other doctors or hospitals, so it can be stored, calculated and shared using our PSI technique.

The overall scheme is described as follows.

Step 1: Stage Initialization

1 The smart contract is signed and deployed between the two parties according to the requirements.
2 The data set from the sender and receiver is represented as two new vectors by using the 0-1 encoding.

Table 1 Notations and descriptions

N	The length of the data set
pk, sk	B's public and private keys
BF_B[i]	B generated Bloom filter
C_i	Ciphertext generated after b's key is added to the dense BF
b_{Ai},b_{Bi}	The number generated when the ith bit in the original set of A or B is encoded with 0-1
$h_{k-1}(b_{Ai})$	The k-1th hash function is used to compute the set encoded by A
C_i^*	The new set extract from C_i
$Enpk$, $Decsk$	Encryption using the public key pk and decryption using the private key sk
T	The intersection of A and B

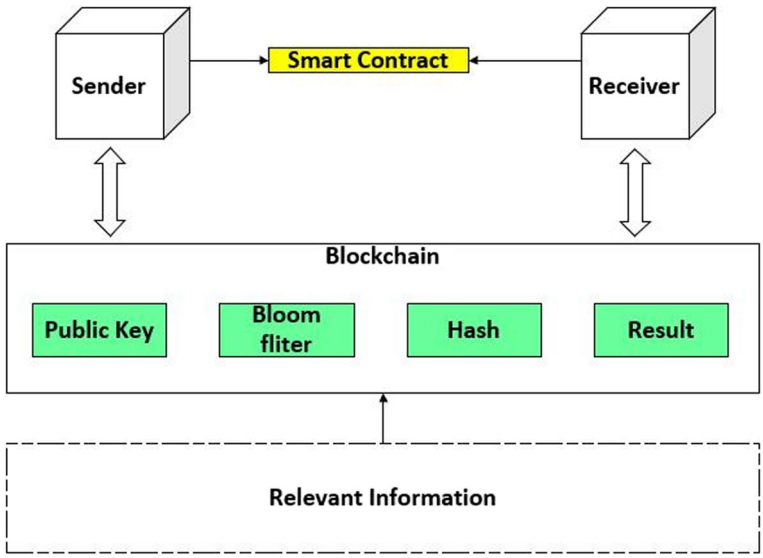

Fig. 3 System model

Step 2: Interaction Setting

1 The sender's new vector is hashed and the receiver generates a public-private key with homomorphic encryption. Its public key is then uploaded to the blockchain through a smart contract, and the sender downloads the public key on the blockchain through the smart contract.
2 The receiver builds a Bloom filter and uploades it to the blockchain, where it encrypts the Bloom filter with its public key.
3 The sender uploads its hash vector to the blockchain through the smart contract. The blockchain begins to extract and compute information through the deployment of smart contract.
4 The receiver downloads the computation result through the smart contract and decrypts the message with its private key.

Step 3: Results Distribution

1 The sender uploads its decrypted vector to the blockchain through the smart contract.
2 The receiver downloads its result.

Scheme Details

The procedure of scheme is shown in Fig. 4.

1 *Stage InitializationStep 1*: The set of both users is represented as two new vectors by the 0-1 code. It is important to note that there exist sets $S_1, S_2 \subseteq \{a_1, a_2, ..., a_n\} = U$ on both sides, where U is the full-order set. if when one of the parties encodes the set $P_A = \{s_1, ..., s_n\}$ as a new vector $b_i = \{b_1, ..., b_n\}$, where if b_i=1, then $S_i \in U$. if b_i =0, then $S_i \notin U$.*Step 2*: Both parties sign and deploy smart contracts according to their respective requirements.
2 *Interaction SettingStep 1*: After encoding the data P_B of sender B, the newly formed vector b_B adopts Pailliar homomorphic encryption to generate public key *pk* and private key *sk*. *Step 2*: The sender construct the Bloom filter $BF_B[i] = (BF_B[0], ..., BF_B[m-1])$. The receiver B's public key *pk* and $BF_B[i]$ are uploaded to the blockchain through a smart contract. The sender A downloads B's public key *pk* on the blockchain through the smart contract. The blockchain encrypts $BF[i]$ using the public key *pk* sent by B to get $C_i = Enpky(BF_B[i])$. *Step 3*: After P_A is encoded, the vector b_A and k hash functions $\{h_0, ..., h_{k-1}\}$ of each b_{Ai} are formed to obtain $\{h_0(b_{Ai}), ..., h_{k-1}(b_{Ai})\}$. The hash set is uploaded to the blockchain through the smart contract. The blockchain extracts the C_i calculated in the previous step to obtain $C_i{}^* = \{C_{h_0}(b_{Ai}), ..., C_{h_{k-1}}(b_{Ai})\}$ and perform the calculation to obtain $e_{b_{Ai}} = (C_i^*)b_{Ai}Enpk(0)$. *Step 4*: The receiver B downloads computation set $(e_1, .., e_n)$. Then decrypt it with own private key *sk*, and get $b_i = Dec_{sk}(e_i)$. Finally, the calculation result b_i is sent to the blockchain through the smart contract.
3 *Results Distribution* After receiving b_i on the blockchain, A downloads the result, and this result is the set of intersection T=$A \cap B$. The number of 1's in its final set is the intersection base.

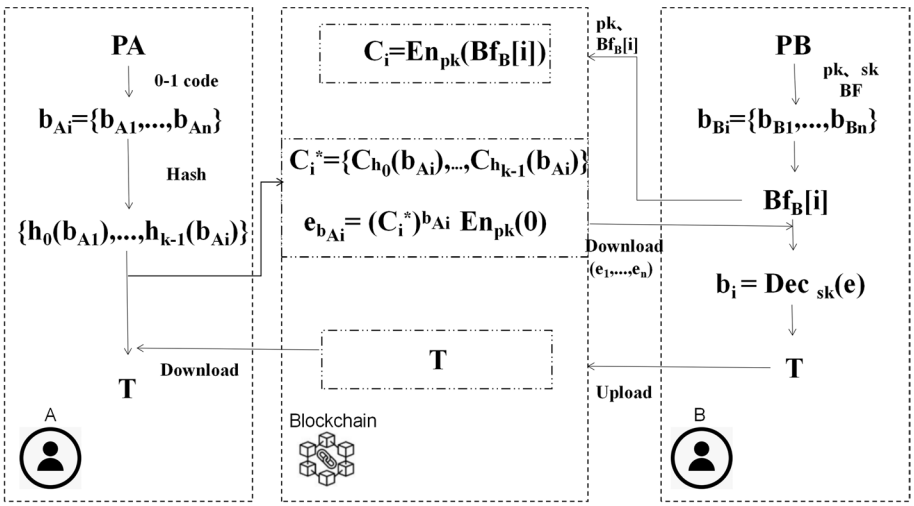

Fig. 4 Scheme procedure

Security Analysis

In this section, we analyze the security of the proposed PSI technique in Security of Proposed PSI section and the security of the proposed scheme in Security of the Proposed Scheme section, respectively.

Security of Proposed PSI

According to Definition 1: The above PSI technique is secure in the semi-honest model. Our evidence is given below, where one side is dishonest and the other honest. In each case, we will construct a simulator in the ideal model. When the PSI technique is performed, there is no difference between the ideal case and the real case when the calculation is made.

Case 1: Corrupted Party A

In such a case, a simulator Sim is constructed which is an ideal model in which one party A is dishonest and has the following cases.

1 Sim generates a set of public and private keys (pk, sk) and sends its public key pk to A.
2 Treat Sim as B and start the technique. $b_{Bi}{}^* = (1 \le i \le n)$ and construct the Bloom filter $BF_B[i]^*$ and encrypt it with $C_i{}^* = Enpk(BF_B[i]^*)$. Then send it to A.
3 After receiving e_{bAi} from A, Sim computes $b_i{}^* = \text{Decsk}(e_i) = BF_B[h(b_{Ai})]$.
4 The position and number of 1's in b_i obtained by Sim is the intersection part, ideal model and then obtained b_i.

In the real execution:
$view_A^{\pi} = (c_i(1 \le i \le n), b_i(1 \le i \le n))$
In the simulation:
$view_f^{\pi} = (c_i{}^*(1 \le i \le n), b_i{}^*(1 \le i \le n))$

After comparing the real execution with the simulated execution of this technique, we get the same results. Then, in Case 1, Sim's view is computationally indistinguishable from the real view. Therefore, the security mode is satisfied.

Case 2: Corrupted Party B

In such a case, a simulator Sim is constructed which is an ideal model in which one party B is dishonest and has the following scenario.

1 Sim generates a set of public and private keys (pk, sk) and sends (pk, sk) to B.
2 Treat Sim as A and initiate PSI technique. When $C_i(1 \le i \le n)$ is received, Sim uses the private key sk to calculate $Dec_{sk}(C_i) = BF_B[i]$; then the position of 1 in the filter is the data information of B.
3 Sim sends the input b_{Bi} of B to the trusted third party in the ideal model, and then obtains the output b.

In the real execution:
$view_A^{\pi} = (c_i(1 \le i \le n), b_i(1 \le i \le n))$
In the simulation:
$view_f^{\pi} = (c_i{}^*(1 \le i \le n), b_i{}^*(1 \le i \le n))$

After comparing the real execution with the simulated execution of PSI technique, we get the same result. Then, in Case 2, Sim's view is computationally indistinguishable from the real view.

Therefore, the PSI technique is secure.

Security of the Proposed Scheme

The security of the basic PSI technique has been proved in 5.1. Our discussion of the security of the scheme will demonstrate two aspects. 1) the security of the data on the blockchain. 2) the security of the scheme if one party is dishonest.

Theorem 1 *Assuming that the scheme is carried out in such a way that the private data of both participating parties are not available to any party. The proposed privacy set intersection technique securely implements the interactive computation on the blockchain.*

Proof

Data security for users: For A, throughout the homomorphic operation, $e_{b_{Ai}} = (c_i{}^*)^{b_{Ai}} En_{pk}(0)$ *in the calculation of* $r^n_{i,k-1}$*, it is a random number, which plays a protective role in protecting A's private data, and the initial data of A is encoded by 0-1 and then expressed by hash calculation, because the hash function has a one-way nature, which in turn ensures A's data security.*

For B, In the whole process of the protocol, B uses the public key $pk = (n, g)$ of the Pailliar encryption algorithm to encrypt $BF_B[i]$ to get $C_i = En_{pk}[BF_B[i]] = g^m r^n (mod n^2)$, which is then uploaded to the blockchain through a smart contract. Since the private key λ is in the hands of B, no one can decrypt C_i, and $g \in Z_n{}^*$ in the public key is chosen randomly for B. Therefore, it is also impossible to obtain B's private message PB from C_i. Therefore, B's data is secure.

Data security on the blockchain: Due to the immutability and traceability of the blockchain, this means that once data is written to the blockchain, no one can easily change the data information without permission. And the information is written to the blockchain in chronological order. Once there is any problem, we can trace back and check every link to ensure the data security of both parties.

Since both the private key *sk* and the λ in the public key are specified by B, the C_i uploaded by B will not cause data leakage even if it is public, so C_i^* is secure. The r in the blockchain e is a random number of A, so the data on the blockchain will not leak the private data of any party even if it is made public.

So that the scheme we propose to get the data is secure.

Theorem 2 *Under the semi-honest model, our blockchain-based PSI scheme is secure.*

Proof

When A does not comply with the scheme and has the malicious act of obtaining others' information, as in 5.1, $(c_i(1 \le i \le n), b_i(1 \le i \le n)) \overset{c}{\equiv} (c_i{}^*(1 \le i \le n), b_i{}^*(1 \le i \le n))$ *and* $view_A^{\pi} \overset{c}{\equiv} view_f^{\pi}$*, so if A does not comply with the scheme, the computational and real views are indistinguishable in Sim's view.*

If receiver B does not comply with the scheme, there is a malicious behavior to obtain information from others. Due to the nature of public key encryption, $(c_i(1 \le i \le n), b_i(1 \le i \le n)) \overset{c}{\equiv} (c_i{}^*(1 \le i \le n), b_i{}^*(1 \le i \le n))$ and $view_A^{\pi} \overset{c}{\equiv} view_f^{\pi}$,B sends the error message through the smart contract and then uploads it to the chain. However, since the vector of sender A is projected by hash function, it is computationally irreversible, so the privacy of A's data is also guaranteed.

Therefore, our proposed scheme is secure under the semi-honest model.

Performance Analysis

In this section, we analyze the performance of the PSI technique presented in Homomorphic Encryption PSI Technique based on 0-1 Encoding section and the performance of the scheme proposed in Proposed Scheme section, respectively.

Performance of Proposed PSI

Since the development of PSI so far, the communication complexity and computational complexity of PSI technique are the most concerned. Therefore, the comparison of the communication and computational complexity of PSI technique with other literatures is analyzed. We evaluate our PSI technique by comparing it with other existing PSI techniques in two major aspects.

1) Computational complexity and communication complexity.

In literature [12], Cristofaro and Tsudik proposed a blind RSA based PKC-PSI protocol with less communication complexity but higher computational overhead. In the literature [21], Pinkas et al. use hash tables to optimize the scheme of Huang [19] et al. but the

computational and communication complexity and is larger. In the literature [23], Dong et al. proposed PSI using Bloom filters and OT extension protocols, but requires a large number of OT protocols as subprotocols and has computationally complex mode index calculations. In protocol [22], Ciampi et al. used circuit-based PSI in two-room secure computation, which has better performance but due to the circuit scheme requires a large amount of memory, in order to reduce the memory consumption, we chose homomorphic encryption based PSI technique. The computational complexity and communication complexity are shown in Table 2.

2) Since our protocol can be used by securely storing the dataset to the cloud server, the encrypted data set can be used directly at the time of use.

In the literature [12, 19, 21, 27], when the data set is outsourced to be stored on a cloud server, it must be encrypted first and then needs to be decrypted in advance at the time of use. As shown in Table 3, our protocol stores the data securely on the cloud server and can directly use the encrypted dataset, thus making it more efficient and convenient, on top of which the use of 0-1 encoding also naturally hides the basis of the dataset.

In our protocol, 0-1 encoding, hashing and homomorphic encryption are used. Triple message encryption achieves great security in the process of interaction. After combining with blockchain, the constraint of smart contract makes the security higher. Therefore, ordinary channels are sufficient in our protocol and scheme.

Performance of the Proposed Scheme

IoT technique still faces various challenges as it is applied and developed. In terms of personal privacy, it is manifested in the fact that personal privacy data can be easily leaked by hackers. In terms of architecture, it is manifested in the rigidity of the architecture, and the current IoT data buildings are aggregated to a single central control system. In terms of transmission communication, the inconsistency of standards and other requirements between various IoTs leads to the formation of equipment access becomes difficult, so data can be called and access control becomes the focus.

According to the above difficulties of IoT, the performance of our solution is evaluated in four aspects: its privacy, user security, data recallability, and access control. In our evaluation, we draw conclusions by comparing the existing schemes with our technique.

Firstly, in the literature [34], the same 0-1 vector is used to generate public and private keys using NTRU homomorphic encryption afterwards. However, this scheme is suitable for multiparty data collection but not for querying data on the IoT, so the application scenario is less and the data is not callable.

Secondly, the proposed schemes in [37] and [38] use *Gödel* coding and ELGamal homomorphic encryption to construct the intersection/merge set of secure multiparty data sets in cloud environment, and random numbers and ELGamal encryption algorithm to construct the set ordering, but the connection between data and user is exposed during the operation, so it is difficult to ensure user security. In the literature [39], a blockchain-based data storage query scheme is proposed using secret

Table 2 Protocol comparison

Protocol	Computational complexity	Communication complexity
De Cristofaro and Tsudik [12]	2(NX+NY)pk[2]	$(NX+NY)\rho+NX\nu$
Pinkas et al. [21]	$3 \in NY + (k+s)NXsym$[1]	$512 \in NY + (k+s)NX\nu$
Dong et al. [23]	$3.6m^{3\kappa}sym$	$1 : 44m\kappa(\kappa+\lambda)$
Ciampi and Orlandi [22]	m(4σ log m+3σ)sym	m(2$\sigma\kappa + m\kappa$)
Ours	2(NX+NY)pk+2.5m κ sym	NYρ+NXν

[1] k,s: The parameters used in the hash function.

[2] κ: The security parameters.

[3] m: max(NX;NY)

Table 3 Security function comparison among existing protocols

Protocol	Secure storage in cloud servers[1]	Secure channel required[2]
De Cristofaro and Tsudik [12]	×	×
Huang et al. [19]	×	×
Abadi et al. [16]	✓	✓
Dong et al. [23]	×	✓
Ours	✓	×

[1] The technique can store users' data securely on cloud servers.

[2] The technique requires a secure connection channel between the two users

Table 4 Security function comparison among existing schemes

Scheme	Privacy[1]	User security[2]	Data callability[3]	Access control[4]
Luo et al. [34]	✓	✓	×	-
Li et al. [37]	✓	×	✓	-
Li et al. [38]	✓	×	✓	-
Zhang et al. [39]	✓	×	×	✓
Ours	✓	✓	✓	✓

[1] The privacy protection for user data held by data providers.

[2] The security of user personal information.

[3] The scheme can store and enable data to be recalled.

[4] The scheme can perform access control

Table 5 Comparison of the runtime with related protocols

Protocol	[12]	[19]	[27]	[21]	[16]	ours
Set size						
2^8	2.89	1.9	1.98	2.49	3.452	1.778
2^{12}	4.099	3.139	2.54	2.51	5.658	2.204
2^{16}	5.31	4.51	3.458	2.835	5.85	3.223
2^{20}	6.504	-	-	3.74	5.658	4.444

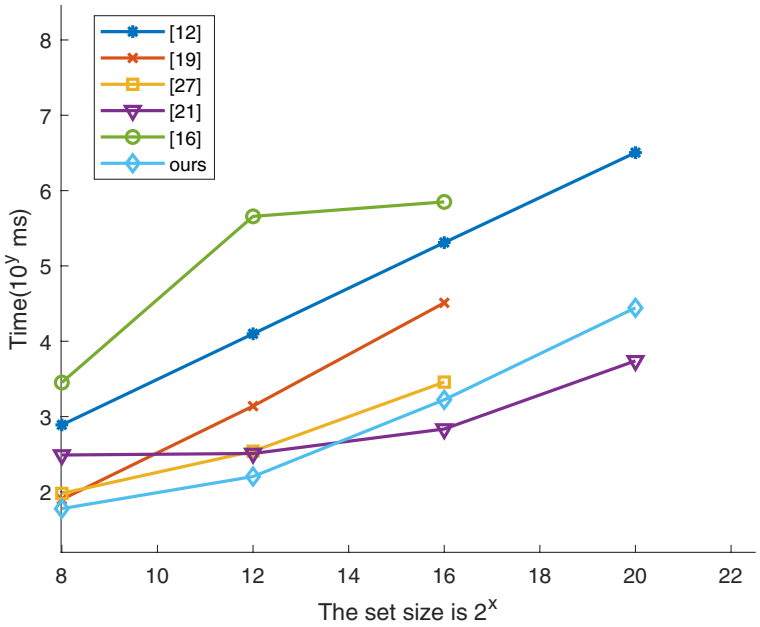

Fig. 5 Running time of related PSI protocols

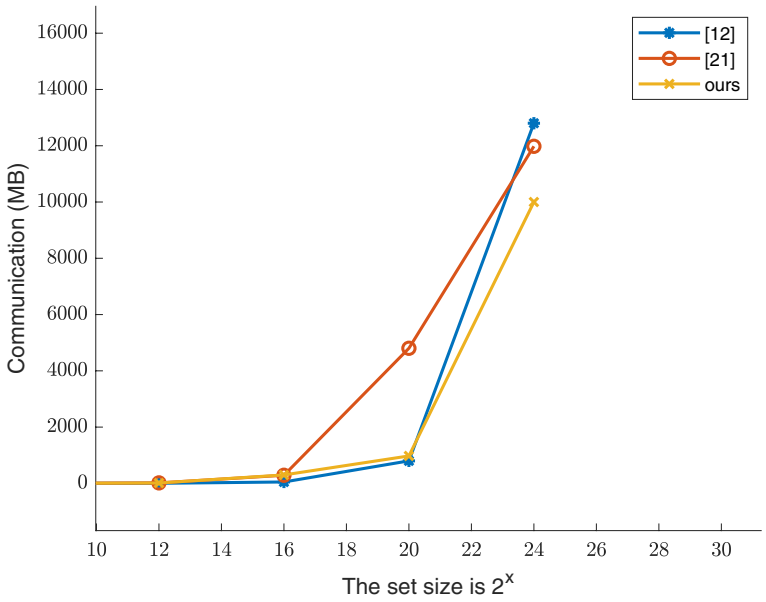

Fig. 6 Communication overhead of related PSI protocols

sharing and smart contracts, but it leaks the relationship between users and the data cannot be recalled while securing the blockchain.We can show from Table 4 that the comparison of security function in existing schemes.

Finally, our scheme is realized by the combination of 0-1 coding and blockchain, the data is callable on the chain with high privacy and user security, and access control can be achieved, which is well combined with the current status of the Internet of Things and related security issues, and has a better performance.

To more visually validate and compare the effectiveness of the technique and the scheme, we conducted the experiments using Windows 10 operating system, Intel(R) Core(TM) i7-5500U CPU @ 2.40 GHz processor, and 8.00 GB RAM. In Table 5, we show the running times under different set sizes after comparing the relevant literature with our protocols. In Fig. 5, the running time of our technique is compared with the literature [12, 16, 19, 21, 27] for different data set sizes. From them it can be seen that our PSI technique is more efficient than the circuit-based and OT-based PSI techniques. In Fig. 6, the communication is compared between our technique and the literature [12, 21] for different data set sizes. From them it can be seen that our scheme has some advantages in use as it is less expensive to communicate with other schemes.

Conclusion

In this paper, we have proposed a blockchain-based homomorphic encryption PSI technique in the IoT environment to achieve privacy protection. First, 0-1 encoding is used to represent the set, and then homomorphic encryption is used for data interaction, which is beneficial to securely store user data in the cloud environment and to directly call the data later. In addition, a data privacy protection scheme for the IoT scenario is designed based on blockchain and smart contract. The scheme uses smart contracts to bind both parties to the agreement, and uses blockchain to achieve data traceability and securely store data on the blockchain. We show that the protocol has strong security and correctness by proving the security and correctness of the protocol. Also, compared with other solutions, our solution has higher user security and callability, and therefore has a wider range of application scenarios.

Acknowledgements
This work is supported by the Basic Research Program of Qinghai Province under 2020-ZJ-701.

Authors' contributions
Qian Zhou and Chengzhe Lai wrote the main manuscript text. Qili Guo and Dong Zheng designed the PSI protocol. Haoyan Ma performed the performance analysis. All authors reviewed the manuscript. The authors read and approved the final manuscript.

Funding
This work is supported by the Basic Research Program of Qinghai Province under 2020-ZJ-701.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 27 September 2022 Accepted: 1 December 2022
Published online: 15 December 2022

References

1. Obaidat M, Khodiaeva M, Obeidat S, Salane D, Holst J (2019) Security Architecture Framework for Internet of Things (IoT). In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, New York, pp. 0154–0157
2. Feng J, Zhang W, Pei Q, Wu J, Lin X (2022) Heterogeneous Computation and Resource Allocation for Wireless Powered Federated Edge Learning Systems. In: IEEE Transactions on Communications. IEEE, New York, vol. 70, no. 5, pp. 3220–3233
3. Feng J, Liu L, Pei Q, Li K (2022) Min-Max Cost Optimization for Efficient Hierarchical Federated Learning in Wireless Edge Networks. In: IEEE Transactions on Parallel and Distributed Systems, IEEE, New York, vol. 33, no. 11. pp. 2687–2700
4. Krishna Kagita M (2019) Security and Privacy Issues for Business Intelligence in IOT. In: 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3). IEEE, New York, pp. 206–212
5. Zhao YL (2013) Research on Data Security Technology in Internet of Things. Appl Mech Mater 433-435:1752–1755
6. Premkumar R, Sathya PS (2021) A Blockchain based Framework for IoT Security. In: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). IEEE, New York, pp. 409-413
7. Du J, et al (2022) Resource Pricing and Allocation in MEC Enabled Blockchain Systems: An A3C Deep Reinforcement Learning Approach. In: IEEE Transactions on Network Science and Engineering, IEEE, New York, vol. 9, no. 1. pp. 33–44
8. Du J, Yu FR, Lu G, Wang J, Jiang J, Chu X (2020) MEC-Assisted Immersive VR Video Streaming Over Terahertz Wireless Networks: A Deep Reinforcement Learning Approach. In: IEEE Internet of Things Journal, IEEE, New York, vol. 7, no. 10. pp. 9517–9529
9. Konen J, et al (2016) Federated Learning: Strategies for Improving Communication Efficiency. Comput Res Repository arXiv:1610.05492v2:1–10
10. Fan GN, Dong P (2016) Research on construction technology of Trusted Execution Environment Based on Trust Zone. Inf Netw Secur 3:21–27
11. Freedman MJ, Nissim K, Pinkas B (2004) Efficient Private Matching and Set Intersection. In: International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, pp. 1–19
12. De Cristofaro E, Tsudik G (2010) Practical private set intersection protocols with linear complexity. In: Proc. Int. Conf. Financial Cryptogr. Data Secur. Tenerife, Springer, pp. 143–159
13. Debnath SK, Dutta R, (2015) Secure and efficient private set intersection cardinality using bloom filter. In: Proc. Int. Conf. Inf. Secur. Trondheim, Springer, pp. 209–226
14. Bloom BH (1970) Space/time trade-offs in hash coding with allowable errors. Commun ACM 13(7):422–426
15. Freedman MJ, Hazay C, Nissim K, Pinkas B (2016) Efficient set intersection with simulation-based security. J Cryptol 29(1):115–155
16. Abadi A, Terzis S, Dong C (2016) VD-PSI: V erifiable delegated private set intersection on outsourced private datasets. In: Proc. Int. Conf. Financial Cryptogr. Data Secur., Christ Church. Barbados, Springer, pp. 149–168
17. Gong L, Li S, Wu C, et al. (2018) Secure "Ratio" Computation and Efficient Protocol for General Secure Two-Party Comparison. IEEE Access 6:25532–25542

18. Chen H, Laine K, Rindal P (2017) Fast Private Set Intersection from Homomorphic Encryption. CCS 1243–1255
19. Huang Y, Evans D, Katz J (2012) Private set intersection: Are garbled circuits better than custom protocols? In: Proc. 19th Netw. Distrib. Syst. Secur. Symp. San Diego, Internet Society, pp. 1–15
20. Zahur S, Rosulek M, Evans D (2015) Two halves make a whole-Reducing data transfer in garbled circuits using half gates. EUROCRYPT (2):220–250
21. Pinkas B, Schneider T, Zohner M (2018) Scalable private set intersection based on OT extension. ACM Trans Privacy Secur 21(2):1–35
22. Ciampi M, Orlandi C (2018) Combining Private Set-Intersection with Secure Two-Party Computation. SCN 464–482
23. Dong C, Chen L, Wen Z (2013) When private set intersection meets big data: an efficient and scalable protocol. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security ACM, USA, pp. 789–800
24. Lambæk M (2016) Breaking and Fixing Private Set Intersection Protocols. IACR Cryptol. ePrint Arch 2016:665
25. Rindal P, Rosulek M (2017) Improved Private Set Intersection Against Malicious Adversaries. Springer, Cham, pp 235–259
26. Rosulek MJ, Rindal P (2016) Faster malicious 2-party secure computation with online/offline dual execution. USENIX Security Symposium, pp. 297-314
27. Pinkas B, Schneider T, Zohner M (2014) Faster private set intersection-based on OT extension. In: Proc. 23rd USENIX Secur Symp. San Diego, USENIX Association, pp 797–812
28. Weng J et al (2019) Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Trans Dependable Secure Comput 18(5):2438–2455
29. Li M et al (2018) CrowdBC: A blockchain-based decentralized framework for crowdsourcing. IEEE Trans Parallel Distrib Syst 30(6):1251–1266
30. Wang S, et al (2021) On private data collection of Hyperledger Fabric. In: 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS). IEEE
31. Fakhri D, Mutijarsa K (2018) Secure IoT Communication using Blockchain Technology. In: 2018 International Symposium on Electronics and Smart Devices (ISESD). IEEE, New York, pp. 1–6
32. Pouraghily A, Islam MN, Kundu S, Wolf T (2018) Poster Abstract: Privacy in Blockchain-Enabled IoT Devices. In: 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI). IEEE, New York, pp. 292-293
33. Zhu Y, et al (2019) Smart Contract Execution System over Blockchain Based on Secure Multi-party Computation. J Cryptologic Res 6(2):246–257
34. Luo Y, Chen Y, Li T, Wang Y, Yang Y (2021) Using information entropy to analyze secure multi-party computation protocol. In: 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress. IEEE, New York, pp 312–318
35. Chen W, et al (2019) Study on blockchain-based privacy collection and intersection scheme. Wirel Internet Technol 16(10):154–155
36. Xiong L, Yang Y et al (2020) Private Set Intersection Algorithm based on Blockchain. Commun Technol 53(7):1768–1773
37. Li SD et al (2016) Secure set computing in cloud environment. J Softw 27(6):1549–1565
38. Li SD et al (2018) String Sorting Based Efficient Secure Database Query. J Softw 28(7):1893–1908
39. Zhang X, et al (2021) Blockchain-based data storage query system. Nanjing University of Posts and Telecommunications
40. Rivest RL, Adleman L, Dertouzos ML (1978) On Data Banks and Privacy Homomorphisms. Found Secure Compuation 4(11):169–180
41. Sander T, Tschudin CF (1998) Protecting Mobile Agents Against Malicious Hosts. In: Mobile Agents and Security. Springer-Verlag, pp. 44-60
42. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. Adv Cryptol Leurocrypt. Springer-Verlag, pp. 223-238

Publisher's Note