electronics

MDPI

Article

A Non-Interactive Attribute-Based Access Control Scheme by Blockchain for IoT

Qiliang Yang [1,*], Mingrui Zhang [1], Yanwei Zhou [1], Tao Wang [1], Zhe Xia [2] and Bo Yang [1]

[1] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China; zmrcrypto@163.com (M.Z.); zyw@snnu.edu.cn (Y.Z.); water@snnu.edu.cn (T.W.); byang@snnu.edu.cn (B.Y.)
[2] School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China; xiazhe@whut.edu.cn
* Correspondence: yangqiliang@snnu.edu.cn

Abstract: As an important method of protecting data confidentiality in the Internet of Things (IoT), access control has been widely concerned. Because attribute-based access control mechanisms are dynamic, it is not only suitable to solve the dynamic access problem in IoT, but also to deal with the dynamic caused by node movement and access data change. The traditional centralized attribute-based access control mechanism has some problems: due to the large number of devices in IoT, the central trusted entity may become the bottleneck of the whole system. Moreover, when a central trusted entity is under distributed denial-of-service (DDoS) attack, the entire system may crash. Blockchain is a good way to solve the above problems. Therefore, we developed a non-interactive, attribute-based access control scheme that applies blockchain technology in IoT scenarios by using PSI technology. In addition, the attributes of data user and data holder are hidden, which protects the privacy of both parties' attributes and access policy. Furthermore, the experimental results indicate that our scheme has high efficiency.

Keywords: private set intersection; attribute-based access control; IoT; blockchain

Citation: Yang, Q.; Zhang, M.; Zhou, Y.; Wang, T.; Xia, Z.; Yang, B. A Non-Interactive Attribute-Based Access Control Scheme by Blockchain for IoT. *Electronics* **2021**, *10*, 1855. https://doi.org/10.3390/electronics10151855

Academic Editors: Hung-Yu Chien, Chun-I Fan, Chunhua Su and Pietro Manzoni

Received: 19 June 2021
Accepted: 30 July 2021
Published: 1 August 2021

1. Introduction

As the evolution of the Internet, the Internet of Things (IoT) [1] has been more and more widely used in people's lives. IoT generates a large amount of data, including personal data. Once these privacies are disclosed, it will bring great losses to users. As one of the important methods of data protection, access control mechanism can guarantee that data is only accessed by users with permission, which has made access control mechanism become an important research content in the security of IoT.

Attribute-based access control mechanism [2,3] is a dynamic access control model that uses attributes as determinants of access control. Compared with the identity-based access control mechanism, the attribute-based access control mechanism makes the attribute set be easily combined with the access structure to achieve fine-grained access control. Attribute sets can also easily represent the identities of certain groups of users, enabling one-to-many communication. Therefore, attribute-based access control can not only solve the dynamic access problem of nodes in the IoT, but also cope well with the dynamics caused by node movement and access data changes.

In traditional access control models, there is a centralized decision-maker to make access decisions based on access control policy and attribute information. Each access request is directed to the same central trusted entity, which holds all the information and makes all decisions based on the stored information. This approach has some drawbacks: when there are many devices in IoT networks, a central trusted entity may become the bottleneck of the entire system. Moreover, when a central trusted entity is under DDoS attacks, the entire system may be disabled.

Blockchain [4] is a good way to solve the above problems. Blockchain is well qualified to become the trusted third party in the access control mechanism in the IoT scenario due to its security, auditability, immutability, anonymity, and other characteristics. In terms of storage capacity, the storage capacity of blockchain is not cheap because it can only add blocks, not delete historical blocks, and as a distributed system, blockchain will keep the same content on every complete node. With the continuous development of blockchain, blockchain has evolved from a ledger database to a secure and trusted platform. The Ethereum-based blockchain has a Turing-complete virtual machine that can execute smart contracts for arbitrarily complex algorithms. Therefore, it is very practical to use smart contracts in the access control mechanism of the IoT.

To sum up, we propose a non-interactive, attribute-based access control scheme by blockchain for IoT. In our work, the data holder stores the data resources in the cloud server. When a user wants to access the data resources, the user first sends their own attribute set confidentially to the blockchain as a transaction. Subsequently, the smart contract of the blockchain will run the private set intersection (PSI) protocol to automatically determine whether the attribute set meets the access structure of the data holder. When the element number of the intersection achieves the threshold set by the data holder, the user is given access to the data holder's cloud data. In our scheme, instead of interacting with data users to verify that a data user is qualified, the data holder deploys their own access policy on the blockchain, and a smart contract automatically determines whether a user is qualified or not. By and large, our work can be summed up in three parts:

1. We developed a non-interactive, attribute-based access control scheme by blockchain for IoT by using PSI technology. In addition, the attributes of data user and data holder are hidden, which protects the privacy of both parties' attributes and access policy.
2. We provide complete security proof of our scheme.
3. We simulated our scheme under the Ethereum Truffle development framework and provide an efficiency analysis.

The rest of our work is shown below. The related work and preliminaries are given in Sections 2 and 3. In Sections 4 and 5, our system model and security model are introduced. In Section 6, we provide our concrete access control scheme. The complete security analysis is presented in Section 7. In Section 8, we present comparisons and performance analysis. In the end, we provide a summary in Section 9.

2. Related Work

Traditional centralized attribute-based access control mechanisms have emerged one after another. For example, Yuan et al. [5], in order to deal with the issues around the fact that the access control models at that time were mostly static and coarse-grained, and thus were not suitable for the dynamic and temporary network service-oriented environment of information access, they proposed an attribute-based access control model, which was depended on the attributes of subjects, environments, and so on. To protect data access in the IoT, Hemdi et al. [6] developed an attribute-based access control mechanism. Their system is able to apply policies to find unauthorized users. Ouechtati et al. [7] proposed an access control system for IoT named Trust ABAC to deal with problems such as the limited storage capacity of mobile devices in the IoT.

However, this type of centralized attribute-based access control mechanism has some drawbacks: firstly, when there are many devices in IoT networks, a central trusted entity may become the bottleneck of the entire system. Moreover, when a central trusted entity is under DDoS attacks, the entire system may be disabled. To solve these problems, blockchain technology has been extensively studied by many scholars and applied to access control mechanisms [8–12].

Blockchain has the ability to technically force all participants to comply with the integrity under the assumption that none of the participants are trustworthy, and it has immutability and privacy protection. Thus, blockchain can become a trusted third platform in the access control for IoT. Some researchers focus on the reliable storage capacity of blockchain. They make use of the characteristics of blockchain, such as immutability and auditability, to provide a secure storage space. Dorri et al. [13] came up with an access control scheme in which the access policies are stored on blockchain and the immutable property of blockchain is used to generate a chronological and immutable transaction history. Alansari et al. [14] used blockchain as a platform to store access policies and users' attributes. The computation-intensive part is executed in Intel SGX, which is a secure hardware external to the chain. Blockchain is only used as a trusted platform to prevent data tampering.

In terms of storage capacity, since blockchain can only add blocks, it cannot delete historical blocks. In addition, as a distributed system, blockchain stores the same content on every full node, and thus the storage capacity of blockchain is not cheap. Therefore, some scholars do not store data on blockchain, the blockchain only stores hashes pointing to the data, and the blockchain is treated as a trusted platform for executable smart contracts. For example, a blockchain-based data access control protocol was provided by Rifi et al. [15] to address the issue of private personal data and sensitive medical data being collected. They took advantage of the computing power of blockchain to maintain authentication and communication between different nodes through three different types of smart contracts, and the transaction data are kept in another database. Cruz et al. [16] proposed a platform called RBAC-SC that leverages Ethereum's smart contract technology for the cross-organizational utilization of users. Zhang et al. [17] developed an access control scheme using smart contract to implement access control in IoT scenarios. An attribute-based access control scheme called TrustAccess was provided by Gao et al. [18] to prevent access policy and attributes leakage.

Blockchain has now evolved from a ledger database to a secure and trusted platform. The trusted computing power provided by blockchain is more valuable than the expensive storage capacity. Therefore, when using blockchain storage, users should store access control data, not data generated by IoT devices. In our proposed scheme, the blockchain stores access policies and users' attributes. The smart contract is applied to determine whether an attribute of a data user meets the access structure of the data holder.

3. Background

3.1. Private Set Intersection (PSI)

Private Set Intersection protocol [19–22] can compute the intersection of two parties' sets secretly, and the two parties know only the cardinality or elements of the intersection and no other information.

3.2. Threshold secret sharing scheme

In (S, T)—threshold secret sharing structure [23], let secret A be divided into T pieces, each of which is held by one party, such that A can be reconstructed from pieces held by at least S parties. In addition, these parties cannot obtain any information from pieces less than S.

4. System Model

In this section, as shown in Figure 1, our system model is given. The model consists of four parties, which are blockchain, cloud server, data user, and data holder.

(1) A data holder stores data in a cloud server.

(2) The data holder uploads access policy to blockchain as a transaction.

(3) If a user wants to obtain the data holder's data, the user first sends their attributes set and public key to blockchain as a transaction.

(4) The smart contract of blockchain runs PSI protocol to obtain cardinality of the intersection. When the element number of the intersection reaches the threshold set by the data holder, the user is allowed to access the data holder's data.

(5) The data holder uses the public key that belongs to the selected data user to encrypt the data address or access token.

(6) The data holder sends the ciphertext to the data user.

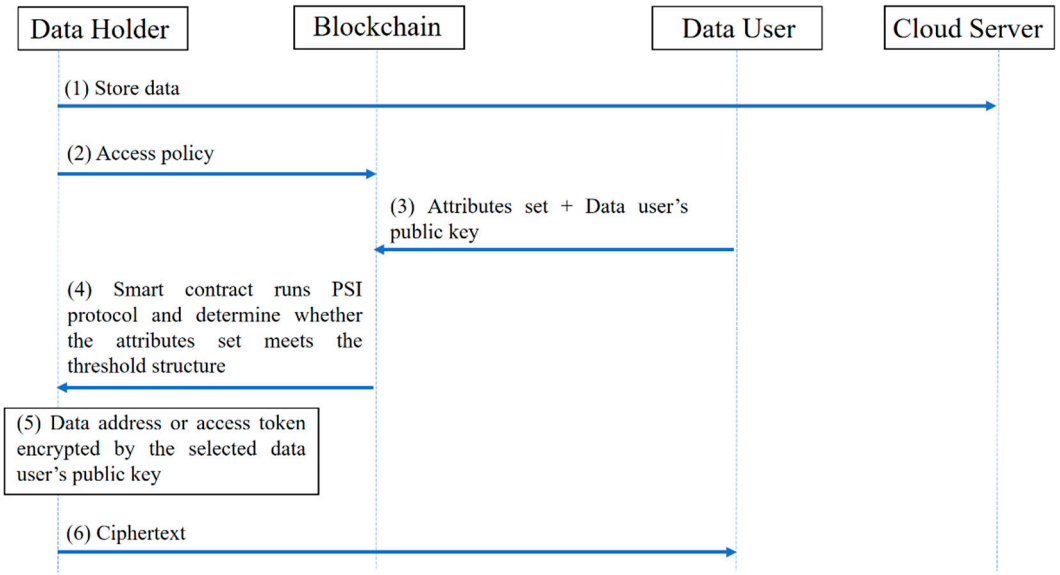


Figure 1. System Model.

5. Security Model

We only assume that adversaries are semi-honest rather than malicious in our security model. This is because if a data holder is malicious in our scenario, they may lie about having some important data to attract users to access. Users will no longer trust the data holder if they find that they have been cheated. The data holder will lose the opportunity to service data users and earn service fees. If a data user is malicious in our scenario, they may fake their own attributes to accommodate the data holder's access structure. Since the PSI protocol is used in our scheme, neither the data user nor the data holder knows which attributes the other has.

In the security model, the adversary corrupts one of the parties. This party abides the protocol directives but may learn more information than allowed after getting transcript of messages. Security of a two-party computing protocol means that both parties do not disclose their input, i.e., security is confidentiality.

Let $R:\{0,1\}^{*}\times\{0,1\}^{*}\to\{0,1\}^{*}\times\{0,1\}^{*}$ be a function, $R_1(p,q)$ and $R_2(p,q)$ are the first element and the second element of $R(p,q)$, respectively. Let TPP be a two-party protocol that computes R. $VIEW_1^{TPP}(p,q)=\{p,d_1,n_1^1,\cdots,n_1^t\}$ represents the view of the data holder, where d_1 is the random number generated by the data holder during the execution of the protocol, and $n_1^i(i=1,\cdots,t)$ represents the i message received by the data holder. Similarly, $VIEW_2^{TPP}(p,q)=\{q,d_2,n_2^1,\cdots,n_2^t\}$ represents the view of the data user. Let $OUTPUT_1^{TPP}(p,q)$ and $OUTPUT_2^{TPP}(p,q)$ be the outputs of the two respective parties.

We say that TPP computes R securely if there exist probabilistic polynomial time algorithms Sim_1 and Sim_2 such that

$$\{(Sim_1(p,R_1(p,q)),R_2(p,q))\}_{p,q\in\{0,1\}^*} \overset{c}{\equiv} \{VIEW_1^{TPP}(p,q),OUTPUT_2^{TPP}(p,q)\}_{p,q\in\{0,1\}^*} \quad (1)$$

$$\{(R_1(p,q),Sim_2(q,R_2(p,q)))\}_{p,q\in\{0,1\}^*} \overset{c}{\equiv} \{OUTPUT_1^{TPP}(p,q),VIEW_2^{TPP}(p,q)\}_{p,q\in\{0,1\}^*} \quad (2)$$

where $|p|=|q|$, Sim_1 and Sim_2 are simulators. The symbol $\overset{c}{\equiv}$ represents computationally indistinguishable.

6. Our Proposed Scheme

Let $\mathbb{G}$ be a group of prime order q, g and h be generators of $\mathbb{G}$, and $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}$ be a bilinear map. Let $X=\{x_1,x_2,\cdots,x_m\}$ be data holder's attributes set; $Y=\{y_1,y_2,\cdots,y_n\}$ be data user's attributes set, where $x_i(i=1,\cdots,m)$; and $y_j(j=1,\cdots,n)$ be elements of $\mathbb{G}$.

(1) A data holder creates a polynomial

$$Q(x)=\prod_{i=1}^{m}(x-x_i)=q_0+q_1x+\cdots+q_mx^m \quad (3)$$

The data holder selects $t_0,t_1,\cdots,t_m$ randomly from $\mathbb{Z}_q^*$ and makes $T_0=g^{t_0},T_1=g^{t_1},\cdots,T_m=g^{t_m}$ public. Then, $S_0=hg^{\frac{q_0}{t_0}},S_1=hg^{\frac{q_1}{t_1}},\cdots,S_m=hg^{\frac{q_m}{t_m}}$ are sent to the smart contract of blockchain by the data holder.

(2) For each $y\in Y$, a data user chooses s randomly from $\mathbb{Z}_q^*$. Then, they compute and send

$$(T_0^{sy^0},T_1^{sy^1},T_2^{sy^2},\cdots,T_m^{sy^m})=(g^{st_0y^0},g^{st_1y^1},g^{st_2y^2},\cdots,g^{st_my^m}) \quad (4)$$

to blockchain.

(3) Then, the smart contract on blockchain computes

$$F=\prod_{i=0}^{m}T_i^{sy^i}=g^{s\sum_{i=0}^{m}t_iy^i} \quad (5)$$

$$\begin{aligned}E&=\prod_{i=0}^{m}e(hg^{\frac{q_i}{t_i}},T_i^{sy^i})\\&=\prod_{i=0}^{m}e(hg^{\frac{q_i}{t_i}},g^{st_iy^i})\\&=\prod_{i=0}^{m}e(g,h)^{st_iy^i}e(g,g)^{sq_iy^i}\\&=e(g,h)^{s\sum_{i=0}^{m}t_iy^i}e(g,g)^{sQ(y)}\\&=e(F,h)e(g,g)^{sQ(y)}\end{aligned} \quad (6)$$

If and only if $y \in X$, $Q(y) = 0$, $E = e(F, h)$, then the smart contract outputs 1, which means the y uploaded by the data user is in the data holder's attributes set. Otherwise, the smart contract outputs 0. In the above process, the smart contract only knows the number of y that is in the data holder's attributes set. Therefore, in our scheme, the privacy of the access policy and the privacy of the attributes of both parties are protected.

• The function of h:

If there is no h in our scheme, $S_0 = g^{\frac{q_0}{t_0}}, S_1 = g^{\frac{q_1}{t_1}}, \cdots, S_m = g^{\frac{q_m}{t_m}}$. In this case, anyone can tell if $y \in X$ by checking that $\prod_{i=0}^{m} e(T_i, S_i)^{y^i} = \prod_{i=0}^{m} e(g^{t_i}, g^{\frac{q_i}{t_i}})^{y^i} = e(g, g)^{Q(y)} = 1$. If the h is introduced into our scheme, we have $\prod_{i=0}^{m} e(T_i, S_i)^{y^i} = e(g, h)^{\sum_{i=0}^{m} t_i y^i} e(g, g)^{Q(y)}$. Since $\sum_{i=0}^{m} t_i y^i$ is a random polynomial, $e(g, g)^{Q(y)}$ is hidden by $e(g, h)^{\sum_{i=0}^{m} t_i y^i}$.

• The function of s:

If there is no s, then for any y, anyone can figure out T_1^y and compare it with the T_1^y in $T_0^{y^0}, T_1^y, T_2^{y^2}, \cdots, T_m^{y^m}$ of the data user uploaded to the blockchain to determine whether $y \in Y$.

7. Security Analysis

(1) A data user is a semi-honest adversary:

The simulator Sim_1, which simulates the data holder, is created as follows: X, $|X \bigcap Y|$, and $|Y|$ are taken as inputs, which means that the simulator Sim_1 can obtain nothing except the data holder's input X and $|X \bigcap Y|$ and $|Y|$ obtained after the end of the protocol.

Moreover, Sim_1 obtains public parameters $T_0 = g^{t_0}$, $T_1 = g^{t_1}, \cdots, T_m = g^{t_m}$, and $S_0 = hg^{\frac{q_0}{t_0}}, S_1 = hg^{\frac{q_1}{t_1}}, \cdots, S_m = hg^{\frac{q_m}{t_m}}$. For $\forall y \in Y$, $(T_0^{sy^0}, T_1^{sy^1}, T_2^{sy^2}, \cdots, T_m^{sy^m}) = (g^{st_0y^0}, g^{st_1y^1}, g^{st_2y^2}, \cdots, g^{st_my^m})$ sent by the data user to blockchain can be simulated by Sim_1. The following procedure is performed $|Y|$ times (X_0 is initially set to be empty):

• If the smart contract outputs 1, $s \leftarrow_R \mathbb{Z}_q^*$, $x \leftarrow_R X \setminus X_0$, computes and outputs $T_0^{sx^0}, T_1^{sx^1}, \cdots, T_m^{sx^m}$, $X_0 = X_0 \bigcup \{x\}$. Because $x \in X \setminus X_0$, $Q(x) = 0$, $T_0^{sx^0}, T_1^{sx^1}, \cdots, T_m^{sx^m}$ satisfies $E = e(F, h)$. Moreover, due to the randomness of s, the tuple $T_0^{sx^0}, T_1^{sx^1}, \cdots, T_m^{sx^m}$ is indistinguishable from the tuple of the data user sent to blockchain in the real experiment.

• If the smart contract outputs 0, $x \in G \setminus X$, compute $T_0^{sx^0}, T_1^{sx^1}, \cdots, T_m^{sx^m}$ so that $E = e(F, h)$ is not true, unless $Q(x) = 0$. However, the probability of this event is negligible.

The data holder can obtain X, $|X \bigcap Y|$, and $|Y|$. Anything else the data holder sees can be simulated by Sim_1. Thus, the data holder cannot obtain any other useful information about the protocol.

(2) A data holder is a semi-honest adversary:

The simulator Sim_2, which simulates the data user, is constructed as follows: Y, $|X \bigcap Y|$, and $|X| = m$ are taken as inputs, and $|X \bigcap Y|$ elements are picked in Y. Moreover, $m - |X \bigcap Y|$ elements are picked in $G \setminus Y$ to form set X. Then, construct polynomial $Q(x) = q_0 + q_1 x + \cdots + q_m x^m$ in the data holder's way. $t_0, t_1, \cdots, t_m \leftarrow_R \mathbb{Z}_q^*$, outputs $T_0 = g^{t_0}, T_1 = g^{t_1}, \cdots, T_m = g^{t_m}$ and $S_0 = hg^{\frac{q_0}{t_0}}, S_1 = hg^{\frac{q_1}{t_1}}, \cdots, S_m = hg^{\frac{q_m}{t_m}}$. These two tuples make $|X \bigcap Y|$ elements in Y satisfy $E = e(F, h)$.

The data user can obtain Y, $|X \bigcap Y|$, and $|X|$. Anything else the data user sees can be simulated by Sim_2. Thus, the data user cannot obtain any other useful information about the protocol.

(3) Access policy privacy

In our scheme, the smart contract runs the private set intersection protocol to determine whether the attributes set of a data user meets the access structure of the data holder. The data user does not know the specific access policy of the data holder.

(4) Attribute privacy

In our scheme, the attributes of the data holder $X = \{x_1, x_2, \cdots, x_m\}$ are converted to a polynomial *Q(x)*, and the coefficients of the polynomial $q_0, ..., q_m$ are then placed on the exponent of $S_0, S_1, \cdots, S_m$. Next, $S_0 = hg^{\frac{q_0}{t_0}}, S_1 = hg^{\frac{q_1}{t_1}}, \cdots, S_m = hg^{\frac{q_m}{t_m}}$ are sent to the smart contract of blockchain by the data holder. Thus, the privacy of data holder's attributes is protected. Moreover, for each $y \in Y$, a data user chooses s randomly from $\mathbb{Z}_q^*$. Then, the data user computes and sends $(T_0^{sy^0}, T_1^{sy^1}, T_2^{sy^2}, \cdots, T_m^{sy^m}) = (g^{st_0y^0}, g^{st_1y^1}, g^{st_2y^2}, \cdots, g^{st_my^m})$ to blockchain. The attributes of the data user are hidden in the exponent of *g*. Therefore, the privacy of data user's attributes is protected.

8. Comparisons and Performance Analysis

As can be seen in Table 1, we first compared our scheme with [8–10,18] in terms of attribute privacy, access policy privacy, and so on. In terms of no intermediary party involved, Zhang et al. [8] and Chen et al. [10] need an intermediary party to distribute keys. However, in our scheme, no intermediate party is required to distribute keys. In terms of access policy privacy, in our scheme, the smart contract runs the private set intersection protocol to determine whether the attributes set of a data user meets the access structure of the data holder. The data user does not know the specific access policy of the data holder. In terms of attribute privacy, in our scheme, the attributes of the data holder $X = \{x_1, x_2, \cdots, x_m\}$ are converted to a polynomial *Q(x)*, and the coefficients of the polynomial $q_0, ..., q_m$ are then placed on the exponent of $S_0, S_1, \cdots, S_m$. Then, $S_0 = hg^{\frac{q_0}{t_0}}, S_1 = hg^{\frac{q_1}{t_1}}, \cdots, S_m = hg^{\frac{q_m}{t_m}}$ are sent to the smart contract of blockchain by the data holder. Thus, the privacy of data holder's attributes is protected. Moreover, for each $y \in Y$, a data user chooses s randomly from $\mathbb{Z}_q^*$. Then, the data user computes and sends $(T_0^{sy^0}, T_1^{sy^1}, T_2^{sy^2}, \cdots, T_m^{sy^m}) = (g^{st_0y^0}, g^{st_1y^1}, g^{st_2y^2}, \cdots, g^{st_my^m})$ to blockchain. The attributes of the data user are hidden in the exponent of *g*. Thus, the privacy of data user's attributes is protected. In terms of fine granularity, since the access control mechanism we have proposed is an attribute-based access control mechanism, we can implement fine-grained access control. In terms of encrypted storage, in our scheme, after selecting a data user, the

data holder uses the public key that belongs to the selected data user to encrypt the data address or access token and sends to the data user. In terms of non-interactivity, in our scheme, the data user and the data holder do not need to interact for access control operations.

Table 1. Comparisons with previous works.

Scheme	No Intermediary Party Involved	Access Policy Privacy	Attribute Privacy	Fine Granularity	Encrypted Storage	Non-Interactive
[8]	×	×	×	✓	✓	×
[9]	✓	×	×	✓	×	✓
[10]	×	×	×	✓	✓	×
[18]	✓	×	✓	✓	✓	×
Ours	✓	✓	✓	✓	✓	✓

Only our proposed scheme can satisfy the above six properties, which are attribute privacy, access policy privacy, fine granularity, encrypted storage, non-interactive, and no intermediary party involved. In addition, as shown in Figure 2, since the scheme in [18] is interactive and our scheme is non-interactive, the efficiency of our scheme is higher than that in [18].

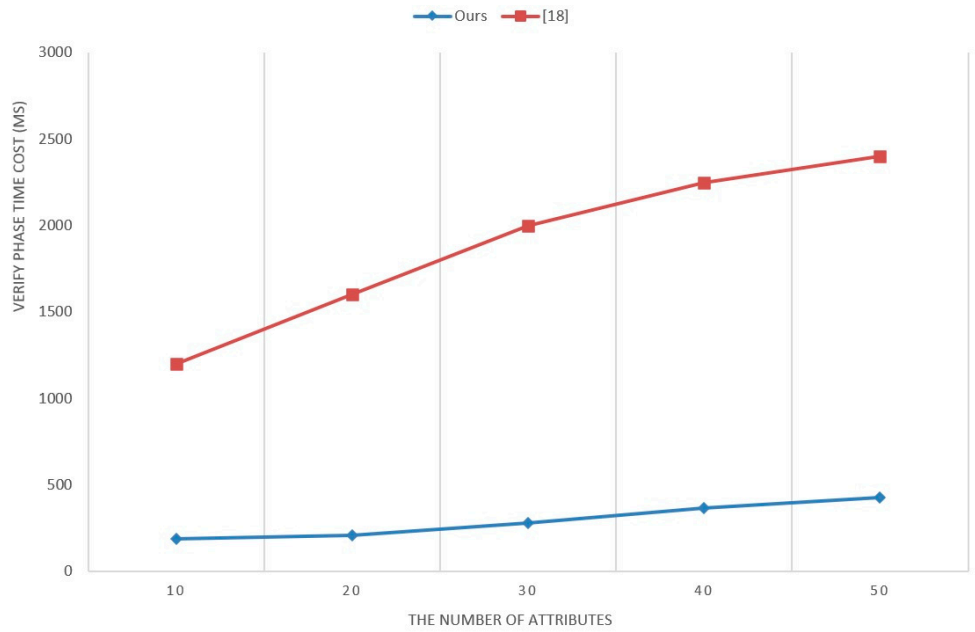


Figure 2. The comparison of verification time of our scheme and [18].

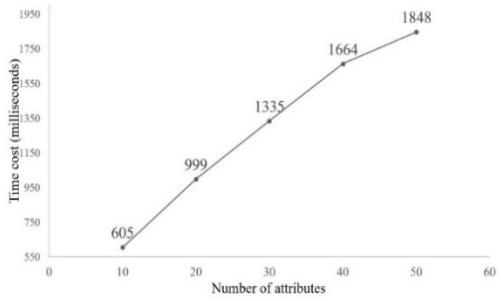
The scheme in [18] is interactive. Because in their scheme, the data user first generates a proof to prove his attributes set satisfies the access policy of the data owner. Then, the data owner generates the decryption key for the data user by the data user's attributes. In addition, the authors of [18] claim that their scheme protects the privacy of access policies. However, in [18], $\vec{x}$ is a part of CT, from which we can know the specific attribute of decrypting a ciphertext. Then, we can derive all the attributes that satisfy the data owner's access policy. That is, the privacy of the access policy is compromised.

Next, we perform an experiment to simulate our proposed scheme. We simulate our proposed scheme on a laptop. The experimental settings are shown in Table 2.

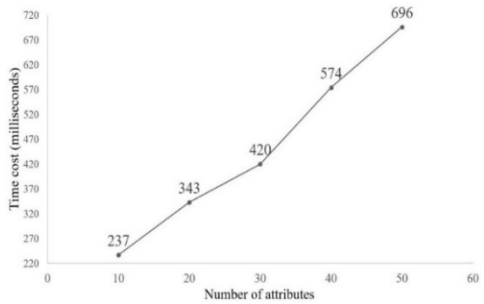
Table 2. Experimental setup.

Language	**Java (Program Version 11.0.3)**
Operating system	Windows 10
Processor	2.60GHz Intel i5-4200H Processor
Memory	8 GB
Cryptography Library	Java Pairing-Based Cryptography Library (JPBC Lib-2.0.0)

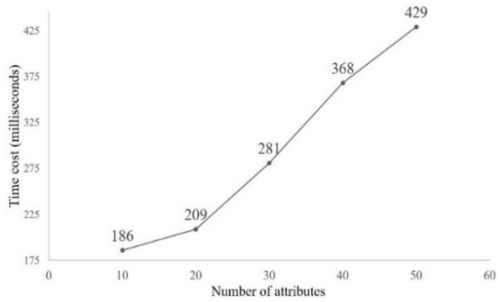
For time measurement, we used Java (11.0.3) as the programming language and Java Pairing-Based Cryptography Library (JPBC Lib-2.0.0) as the Cryptography Library. As shown in Figure 3a–c, we set m = 10, 20, 30, 40, 50 attributes to measure the time cost of data holder, data user, and verification. Moreover, taking the average of five measurements, we found the setup time of our scheme to be 2719 ms.



(**a**)



(**b**)



(**c**)

Figure 3. (**a**) Time cost of data holder; (**b**) time cost of data user; (**c**) verification time cost.

Figure 3a shows the time spent by the data holder in the first step in our scheme to create the polynomial $Q(x) = \prod_{i=1}^{m}(x - x_i) = q_0 + q_1 x + \cdots + q_m x^m$ according to the number of its attributes and calculate $T_0 = g^{t_0}, T_1 = g^{t_1}, \cdots, T_m = g^{t_m}$ and $S_0 = hg^{\frac{q_0}{t_0}}, S_1 = hg^{\frac{q_1}{t_1}}, \cdots, S_m = hg^{\frac{q_m}{t_m}}$.

Figure 3b shows the time spent by the data user in the second step in our scheme to calculate $(T_0^{sy^0}, T_1^{sy^1}, T_2^{sy^2}, \cdots, T_m^{sy^m}) = (g^{st_0y^0}, g^{st_1y^1}, g^{st_2y^2}, \cdots, g^{st_my^m})$.

Figure 3c shows the time spent by the smart contract on blockchain to calculate $F = \prod_{i=0}^{m} T_i^{sy^i} = g^{s\sum_{i=0}^{m} t_i y^i}$ **and** $E = \prod_{i=0}^{m} e(hg^{\frac{q_i}{t_i}}, T_i^{sy^i})$.

The Ethereum transaction price was 1 ETH = USD 339 when this paper was written. Suppose the gas price is 1gas = 1×10^{9} wei. 1wei = 1×10^{-18} ETH, so 1gas = 1×10^{-9} ETH = 3.39×10^{-7} USD. We measured the smart contract gas consumption of storing attribute elements. As shown in Table 3, we set 10, 20, 30, 40, and 50 attributes to perform gas consumption computations.

Table 3. The smart contract cost of storing attribute elements.

Number of Attributes	Gas Used	USD
10	78089	0.0264
20	98139	0.0332
30	133196	0.0451
40	168258	0.0570
50	203327	0.0689

9. Conclusions

We have developed a non-interactive access control scheme by blockchain for IoT by using PSI technology. A data holder uploads data to a cloud server. If a user wants to access the data, the data user first writes attributes to blockchain as a transaction. Next, the PSI protocol is run by a smart contract to determine whether the attributes set meets the threshold structure. If the condition is met, the data user is allowed to access the data holder's data. Then, the data holder uses the selected user's public key to encrypt the data address and sends it to the user. Our scheme is able to protect both the privacy of access policy and the privacy of attributes while ensuring trusted access control. In addition, a complete security proof is given. On the basis of the Ethereum Truffle development framework, we simulated the scheme in the Windows 10 system, and the experimental results indicate that our scheme has high efficiency.

Author Contributions: Conceptualization, Q.Y., Y.Z., T.W., Z.X., and B.Y.; methodology, Q.Y.; software, Q.Y. and M.Z.; validation, Q.Y., M.Z., Y.Z., T.W., Z.X., and B.Y.; formal analysis, Q.Y.; investigation, Q.Y.; resources, B.Y.; data curation, Q.Y.; writing—original draft preparation, Q.Y.; writing—review and editing, Q.Y.; visualization, Q.Y.; supervision, Q.Y.; project administration, B.Y.; funding acquisition, B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number U2001205.

Data Availability Statement: The experimental data of this paper is true and reliable. The relevant code link in this paper is https://github.com/QiliangYang/An-Access-Control-Scheme-by-Using-Blockchain-in-Cloud-Storage-Environment (accessed on 1 August 2021).

Acknowledgments: The authors would like to thank the anonymous reviewers for your helpful comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ashton, K. That 'internet of things' thing. *RFID J.* **2009**, *22*, 97–114.
2. Chen, Y.W.; Meng, L.H.; Zhou, H.; Xue, G.T. A Blockchain-Based Medical Data Sharing Mechanism with Attribute-Based Access Control and Privacy Protection. Wireless Communications and Mobile Computing. 2021, 6685762:1-6685762:12.
3. Yin, H.; Xiong, Y.Q.; Zhang, J.X.; Ou, L.; Liao, S.L.; Qin, Z. (2019). A Key-Policy Searchable Attribute-Based Encryption Scheme for Efficient Keyword Search and Fine-Grained Access Control over Encrypted Data. *Electronics* **2019**, *8*, 265–285, doi:10.3390/electronics8030265.
4. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2009. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 1 March 2009).
5. Yuan, E.; Tong, J. Attributed based access control (ABAC) for Web services. In Proceedings of the 2005 IEEE International Conference on Web Services (ICWS 2005), Orlando, FL, USA, 11–15 July 2005; pp. 561–569.
6. Hemdi, M.; Deters, R. Using REST based protocol to enable ABAC within IoT systems. In Proceedings of the 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference, Vancouver, Canada, 13–15 August 2016; pp. 1–7.
7. Ouechtati, H.; Azzouna, N.B. Trust-ABAC towards an access control system for the Internet of things. In Proceedings of the 12th Green, Pervasive, and Cloud Computing International Conference (GPC 2017), Cetara, Italy, 11–14 May 2017; pp. 75–89.
8. Zhang, Y.; Li, B.; Liu, B.; Wu, J.; Wang, Y.; Yang, X. An Attribute-Based Collaborative Access Control Scheme Using Blockchain for IoT Devices. *Electronics* **2020**, *9*, 285, doi:10.3390/electronics9020285.
9. Song, L.; Ju, X.; Zhu, Z.; Li, M. An access control model for the Internet of Things based on zero-knowledge token and blockchain. *J. Wirel. Commun. Netw.* **2021**, *2021*, 1–20.
10. Chen, H.; Wan, W.; Xia, J.; Zhang, S.; Zhang, J.; Peng, X.; Fan, X. Task-attribute-based access control scheme for iot via blockchain, *Comput. Mater. Contin.* **2020**, *65*, 2441–2453.
11. Ouaddah, A.; Kalam, A.A.E.; Ouahman, A.A. FairAccess: A new Blockchain-based access control framework for the Internet of Things. *Secur. Commun. Netw.* **2016**, *9*, 5943–5964.
12. Alphand, O.; Amoretti, M.; Claeys, T.; Dall'Asta, S.; Duda, A.; Ferrari, G.; Rousseau, F.; Tourancheau, B.; Veltri, L.; Zanichelli, F. IoTChain: A blockchain security architecture for the Internet of Things. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC 2018), Barcelona, Spain, 15–18 April 2018; pp. 1–6.
13. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT security and privacy: The case study of a smart home. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops 2017). Kona, Big Island, HI, USA, 13–17 March 2017; pp. 618–623.
14. Alansari, S.; Paci, F.; Sassone, V. A distributed access control system for cloud federations. In Proceedings of the 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017), Atlanta, GA, USA, 5–8 June 2017; pp. 2131–2136.
15. Rifi, N.; Rachkidi, E.; Agoulmine, N.; Taher, N.C. Towards using blockchain technology for IoT data access protection. In Proceedings of the 17th IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB 2017), Salamanca, Spain, 12–15 September 2017; pp. 1–5.
16. Cruz, J.P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-based access control using smart contract. *IEEE Access* **2018**,*6*, 12240–12251.
17. Zhang, Y.; Kasahara, S.; Shen, Y.; Jiang, X.; Wan, J. Smart contract-based access control for the Internet of things. *IEEE Internet Things J.* **2019**, *6*, 1594–1605.
18. Gao, S.; Piao, G.R.; Zhu, J.M.; Ma, X.D.; Ma, J.F. TrustAccess: A Trustworthy Secure Ciphertext-Policy and Attribute Hiding Access Control Scheme Based on Blockchain. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5784–5798.
19. Pinkas, B.; Rosulek, M.; Trieu, N.; Yanai, A. PSI from PaXoS: Fast, Malicious Private Set Intersection. In Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 10–14 May 2020; pp. 739–767.
20. Ruan, O.; Wang, Z.; Mi, J.; Zhang, M.W. New Approach to Set Representation and Practical Private Set-Intersection Protocols. *IEEE Access* **2019**, *7*, 64897–64906.
21. Lv, S.Y.; Ye, J.H.; Yin, S.J.; Cheng, X.C.; Feng, C.; Liu, X.Y.; Li, R.; Li, Z.H.; Liu, Z.L.; Zhou, L. Unbalanced private set intersection cardinality protocol with low communication cost. *Future Gener. Comput. Syst.* **2020**, *102*, 1054–1061.
22. Cristofaro, E.D.; Tsudik, G. Practical Private Set Intersection Protocols with Linear Complexity. In Proceedings of the 14th Financial Cryptography, Tenerife, Canary Islands, Spain, 25–28 January 2010; pp. 143–159.
23. Shima, K.; Doi, H. New Proof Techniques Using the Properties of Circulant Matrices for XOR-based (k, n) Threshold Secret Sharing Schemes. *J. Inf. Process.* **2021**, *29*, 266–274.