Accepted Manuscript

PTAS: Privacy-preserving Thin-client Authentication Scheme in Blockchain-based PKI

Wenbo Jiang, Hongwei Li, Guowen Xu, Mi Wen, Guishan Dong, Xiaodong Lin

PII: S0167-739X(18)31509-7
DOI: https://doi.org/10.1016/j.future.2019.01.026
Reference: FUTURE 4719

To appear in: *Future Generation Computer Systems*

Received date : 7 September 2018
Revised date : 14 December 2018
Accepted date : 15 January 2019

Please cite this article as: W. Jiang, H. Li, G. Xu et al., PTAS: Privacy-preserving Thin-client Authentication Scheme in Blockchain-based PKI, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.01.026

PTAS: Privacy-preserving Thin-client Authentication Scheme in Blockchain-based PKI

Wenbo Jiang[a], Hongwei Li(corresponding author)[a,b,*], Guowei Xu[a], Mi Wen[c], Guishan Dong[d], Xiaodong Lin[e]

[a] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, China*
[b] *Science and Technology on Communication Security Laboratory, Chengdu 610041, China*
[c] *School of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China*
[d] *National Engineering Laboratory of Big Data application to improving the Government governance capacity in China*
[e] *Department of Physics and Computer Science, Faculty of Science, Wilfrid Laurier University, Canada*

Abstract

Recent years have witnessed tremendous academic efforts and industry growth in Internet of Things (IoT). Security issues of IoT have become increasingly prominent. Public Key Infrastructure (PKI) can provide authentication service to IoT devices which is a crucial element to the security of IoT. However, the conventional PKIs are organized as a tree-like centralized structure which has demonstrated serious usability and security shortcomings such as the single point of failure. Blockchain has numerous desirable properties, such as decentralized nature, cryptographic technology and unalterable transaction record, these properties make it a potential tool to build a decentralized blockchain-based PKI. Nevertheless, the latest proposals for blockchain-based PKI didn't take thin-client into consideration where thin-clients indicate those users who can't download the entire blockchain due to the limited storage capacity of their equipment (most IoT devices fall into this category). To settle this problem, we firstly present a Privacy-preserving Thin-client Authentication Scheme (PTAS) employing the idea of private information retrieval (PIR), which enables thin-clients to run normally like full node users and protect their privacy simultaneously. Furthermore, in order to enhance security, we further propose a (m-1)-private PTAS which means thin-client's

*Corresponding author
Email address: `hongwei.uestc@gmail.com` (Hongwei Li(corresponding author))

information can be protected against a collusion of at most (m-1) full node users. Besides, security analysis and functional comparison are performed to demonstrate high security and comprehensive functionality of our schemes. Finally, extensive experiments are conducted to compare computational overhead and communication overhead of PTAS and (m-1)-private PTAS.

Keywords: Public Key Infrastructure, Blockchain, Internet of Things, Privacy-preserving.

---

1. Introduction

Internet of Things (IoT) is an important part of a new generation of information technology. It is widely used in the convergence of networks through intelligent perception, recognition technology, pervasive computing, etc. Therefore, IoT is also called the third information technology revolution after the computer and the Internet. It has shown promising application prospects in many fields such as Internet of Vehicles [1], Vehicle-to-Grid (V2G) [2–4] and so on.

However, IoT devices may suffer numerous malicious attacks. Many devices are vulnerable to hackers and are easy to be infected to form botnets [5] because of lacking security protection. In fact, considerable research efforts have been devoted to security and privacy issues of IoT [6–9]. Among these, one of the biggest challenges to IoT security is authentication. Current IoT systems rely on centralised cloud servers. Specifically, all devices are identified, authenticated and connected through cloud servers. Apparently, this structure remains flawed: the single point of failure can disrupt the entire network.

Public Key Infrastructure (PKI) for IoT is an infrastructure that can secure the communication between IoT devices. To be more specific, PKI distributes certificates to devices to build a correct binding between a public key (PK) and an identity (ID). The traditional architecture of PKI relies on a trusted third party named Certificate Authorities (CAs), for example the Internet X.509 Public Key Infrastructure [10]. Unfortunately, this design has demonstrated serious usability and security shortcomings [11]. The most serious one is the single point of failure which is inevitable under the centralized structure. It is commonly known that traditional PKIs are organized as a tree-like structure and the root of this tree is a Root-CA, which means the whole structure will be affected once a Root-CA is attacked. Several security incidents in recent years are good examples, showing that CAs are vulnerable due to their centralized structure [12].

Much work so far has focused on solving this problem. Among them, decentralized PKI without a certificate authority (CA) is a possible replacement of current PKI. Generally speaking, the goal of decentralized PKI is to eliminate trusted third party in the system. The Web of Trust model in Pretty Good Privacy (PGP) is the first step toward realizing a decentralized PKI which was initially designed for email users to exchange public keys without relying on a CA. Nevertheless, it still faces many barriers in usability and security such as lacking incentive and leaking privacy. Besides, the discovery and construction of certificate chains still rely on centralized keyservers (complete details of these disadvantages will be provided at section 7).

Blockchain has lately received great attention since it was first coined in 2008 [13]. It is a continuously growing list of records (or blocks), which are linked and secured using cryptographic technology. Typically each block contains a timestamp, a hash of the previous block, version information and transaction data. It has proved promising in many application aspects such as energy Internet [14], intelligent transportation systems [15] and IoT applications [16, 17]. In fact, its desirable properties, such as decentralized nature and reliable transaction records, make it a suitable tool to implement a decentralized PKI. A gamut of blockchain-based PKIs have been proposed in the literature [18–25], but none of them considered thin-clients (such as smartphone users) in their protocols. This kind of device has limited memory resources to download the entire blockchain in their devices (most IoT devices fall into this category). In blockchain-based PKIs, how to make these devices run normally and protect their privacy simultaneously still requires further research.

In this paper, we begin by presenting a Privacy-preserving Thin-client Authentication Scheme (PTAS) in blockchain-based PKI. In PTAS, thin-clients have the same functions as full node users and user's privacy will be protected by utilizing the idea of private information retrieval (PIR). After that, in order to gain better security, we propose a (m-1)-private PTAS which means user's privacy can be guaranteed even any (m-1) full node users collude together. Our contributions can be summarized in three aspects as follows:

- We propose a Privacy-preserving Thin-client Authentication Scheme (PTAS) in blockchain-based PKI which enables thin-clients to run normally as full node users. In PTAS, we leverage on the method of PIR so that the identity of the user who is authenticating with the thin-client can be hidden in k indistinguishable identities. To the best of our knowledge ,this is the first work aiming to address the issues of thin-client in blockchain-based PKI.
- For the purpose of improving security, we present a (m-1)-private PTAS. In (m-1)-private PTAS, even if (m-1) full node users colluded together, they still can't get any information about thin-client at all.
- We compare our scheme with latest proposals and find our scheme is the richest in functionalities. Experiments demonstrate that (m-1)-private PTAS sacrifice little efficiency in exchange for safety improvement. Aside from that, extensive experiments confirm that computational overhead of both schemes is in a reasonable range which will not be a burden to smartphone users.

Differ from the preliminary conference version [26], which proposed a privacy-preserving thin-client scheme (PTS) and an efficient privacy-preserving thin-client scheme (EPTS). EPTS can improve efficiency impressively, but user's information may be deduced if several nodes colluded together. In order to gain more security, we presented a (m-1)-private privacy-preserving thin-client authentication scheme ((m-1)-private P-TAS) in this paper. In (m-1)-private PTAS, even if (m-1) full node users colluded together, they still can't get any information about thin-client at all.

In addition, a more comprehensive and detailed performance evaluation is presented in this paper. More concretely, in the conference version, our analysis of computational overhead is limited to the case of m = 4, 8, 16. In this paper, we analyzed computational overhead of our schemes when m = $2d$ and reached a conclusion that the total computational overhead of thin-client in PTAS and (m-1)-private PTAS are in an acceptable range that it will not be a burden to a mobile phone user. Besides, we also compared communication overhead of our schemes when the values of k and m are large. On the basis of these analyses, the conclusion can be obtained that (m-1)-private PTAS can provide higher security but its efficiency is a little inferior to PTAS.

Furthermore, we also revised the conference version to enhance the presentation and readability. More precisely, in background part, we introduced the traditional structure of public key infrastructures (PKI) and analyzed shortcomings of this structure, we also added an introduction to blockchain, mainly describing its four properties: decentralization, non-modifiability, unforgeability and anonymity. After that, we amended the blockchain-based PKI part to make it more detailed and more readable. Compared with the conference version, we added more literature in related work part. Then we divided all literature into two categories: decentralized PKI and blockchain-based PKI.

The rest of this paper is structured as follows. Section 2 introduces the background of blockchain-based PKI. Then we describe the details of PTAS and (m-1)-private PTAS in Section 3 and Section 4, respectively. Section 5 and Section 6 analyse the security and evaluation of our schemes. Related work will be discussed in Section 7. Finally, Section 8 concludes our paper.

2. Background: preliminaries

2.1. Public Key Infrastructure

With the rapid growth and popularization of the Internet, more and more people are communicating through the Internet. Public key infrastructure (PKI) is an important infrastructure that can secure the communication between these participants. More concretely, PKI is built based on asymmetric cryptography, whose function is to build a link between a public key and its owner. Actually, the essential function of PKI is to confirm that a person really owns the public key (and the corresponding private key). In PKI, a certificate authority (CA) issues a digital certificate, which binds the user's identity information and his public key. In the communication process, the certificate relying party obtains the certificate chain of the communication partner and then uses the root-CA certificate stored in the configuration to verify each certificate in the certificate chain one by one. Finally, he can obtain the public key of the communication partner credibly, which is used for various security functions such as confidentiality, data integrity, identity authentication, non-denial, etc.

However, in traditional PKI system, as illustrated in Fig. 1, it requires a trusted third party to act as a certification authority (CA) to issue a digital certificate to confirm the true identity of the public key owner. All CAs are organized into a tree-like structure and the root of this tree is a Root-CA, which means every CA in this system will be affected once a Root-CA is attacked.

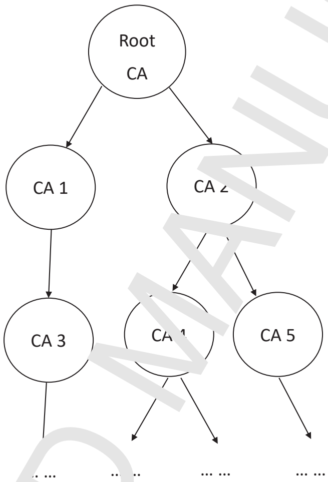


Figure 1: Tree hierarchy of PKI

The hacking of DigiNotar's systems (a Dutch CA) in 2011 [12] provides an example of it. In addition, certification authority is placed in a privileged place to supervise user's communication, which means all the user's information is controlled by CA and users have no privacy at all.

2.2. *Blockchain*

The blockchain was originally introduced by Nakamoto as the technology underlying cryptocurrency Bitcoin [13] in 2008. Blockchain is a decentralized distributed database. The data is stored in blocks and blocks are arranged in a chronological order. Specifically, each block contains several transaction records, a timestamp, a hash of the previous block and version information. The blocks are generated using cryptographic methods which ensure the data in the block can't be altered or forged. In general, blockchain's properties can be summarized in the following four aspects:

- Decentralization: Due to the use of distributed storage, there is no central node or centralized management organization in the system. The rights and obligations of any node are equal and blocks in the system are maintained by all nodes in the entire system. The decentralized structure provides better fault tolerance. Once a centralized system has problems in the center, all the other nodes will collapse easily. On the contrary, this problem would never have arisen in decentralized systems because they rely on all nodes.

- Non-modifiability: Each transaction stored in the blockchain has a corresponding hash and a binary Merkle tree is generated from this hash. The hash value of the Merkle tree is stored in the block header together with a timestamp and the identifier of the previous block. Therefore, if an attacker wants to tamper with a record in the blockchain, he needs not only to modify the hash of the block, but also to modify the hash of all subsequent blocks which are nearly impossible to achieve.

- Unforgeability: The transaction data stored in the blockchain contain not only the hash value, but also the signature of both parties which is unforgeable.

- Anonymity: The anonymity in the blockchain is actually pseudo-anonymity. In the blockchain system, the user performs a series of hash operations on the public key and obtains a fixed-length hash value as the corresponding account in order to cut off the connection between the real identity. In fact, with the use of this account, account's trading behavior can be tracked through the transaction data, such as which accounts are trading with this account, the amount of transaction, and even can be linked to the actual identity in reality.

2.3. Blockchain-Based PKI

In this section, we adapt the model of CertCoin[18] to provide the sketch of blockchain-based PKI. The core idea of CertCoin is to maintain a public ledger of users' identities and their associated public keys. The system comprises six main functionalities: registering an identity with a corresponding public key, updating the public key, searching a public key corresponding to a given identity, revoking the public key corresponding to an identity, recovering the public key corresponding to an identity and mining.

2.3.1. Notations

- σ = *sig*(SK, μ): a digital signature σ on the message μ using the secret key (SK).
- b = *ver*(PK, σ, μ): a verification that evaluates to 0 or 1. b=1 if σ is a valid signature on μ under the secret key corresponding to the public key PK, otherwise b=0.

2.3.2. Registering

(1) Identity owner generates an online public and secret key pair (PK_n, SK_n) and an offline public and secret key pair (PK_f, SK_f). The key pairs must be generated locally (e.g.,via open source client software on user's device) and private key must never be stored or transmitted in an insecure manner.

(2) The identity owner posts (ID, online, register, values= (PK_n, σ_n)) and (ID, offline, register, values= (PK_f, σ_f)) (which will be saved as a transaction in blockchain) to the blockchain where:

- ID is an identity
- PK_n is the online public key
- PK_f is the offline public key
- σ_n = *sig*(SK_n, ID) and σ_f = *sig*(SK_f, ID) are two digital signatures which manifest that the identity owner has the control of SK_n and SK_f.

(3) After getting this information, the block miner preforms the following verifications:

- Traverse through the whole blockchain to check that ID and PK have never been registered before.
- Use the online public key PK_n to check whether *ver*(PK_n, σ_n, ID) = 1.

- Use the offline public key PK_f to check whether *ver*(PK_f, σ_f, ID) = 1.

(4) If any of these verifications fails, the block miner will not put this information into the blockchain. Otherwise, he or she accepts it and includes it in the blockchain. Each recipient of the mined block performs the same verifications as the block miner. If any of these verifications fail, the recipient discards the received block.

2.3.3. Updating

(1) The user posts (ID, update, type of key, values=(PK^{old}, PK^{new}, σ_1, σ_2)) (which will be saved as a transaction in blockchain) onto the blockchain where:

- ID is an identity.
- PK^{old} is the old public key.
- PK^{new} is the new public key which is to replace PK^{old}
- σ_1 = *sig*(SK^{old}, (ID, PK^{new})) is a digital signature of the identity together with the new public key, signed by the old secret key. This proves that the identity owner knows the secret key SK^{old} corresponding to the old public key PK^{old}, and that PK^{new} is the intended new public key for ID.
- σ_2 = *sig*(SK^{new}, ID) is a digital signature of the identity signed by the new secret key. This proves that the identity owner knows the secret key SK^{new} corresponding to the new public key PK^{new}.

(2) The block miner also preforms the following verifications:

- Verify that PK^{old} corresponds to ID (by 2.3.5 searching a public key corresponding to a given identity)
- Use the old public key PK^{old} to check whether *ver*(PK^{old}, σ_1, (ID, PK^{new})) = 1.
- Use the new public key PK^{new} to check whether *ver*(PK^{new}, σ_2, ID) = 1.

(3) is the same as step 4) in 2.3.2.

2.3.4. *Key Recovery and Revocation*

As for key recovery, user's secret key is secretly shared (e.g. using the Shamir secret sharing paradigm[27]) among at least three trusted "friends" and the secret key can be reconstructed with at least two "friends". For ensuring safety, these "friends" should be unaware of each other.

Key revocation is generally handled through Certificate Revocation List (CRL) in traditional centralized PKI [28]. It has a list of certificates that have been revoked. It is well known that maintaining a CRL can be very costly. However, revocation in blockchain-based PKI can be very simple as follows.

(1) An owner of an identity ID can revoke his public key simply by posting (ID, revoke, type of key, PK_n, PK_f, σ_n, σ_f) onto the blockchain (it will be saved as a transaction in blockchain), where σ_n is signature on (ID, revoke, type of key) under the online secret key SK_n and σ_f is signature on (ID, revoke, type of key) under the offline secret key SK_f.

(2) The block miner checks whether *ver*(PK_n, σ_n, (ID, revoke, type of key)) = 1 and *ver*(PK_f, σ_f, (ID, revoke, type of key)) = 1.

(3) is the same as step 4) in 2.3.2.

2.3.5. *Searching a Public Key Corresponding to a Given Identity*

It is important to highlight that the user needn't to traverse the entire blockchain to look up for a public key, all he needs to do is to simply find the most recent transaction posted by the given identity. Then, by retrieving the content of this transaction, he or she will obtain the public key corresponding to the given identity or get a conclusion that the given identity has been revoked.

2.3.6. *Mining*

As the following Fig. 2 shows, registrations, updates and revocations are handled simply by posting the appropriate information to the blockchain and all this information is stored in the block just like transactions in bitcoin.

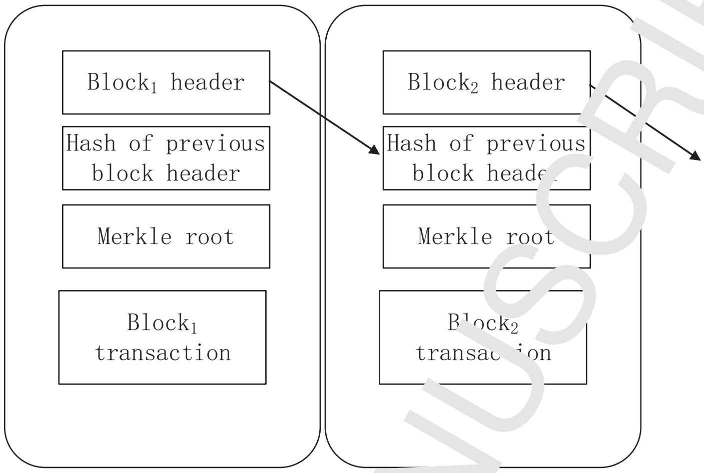


Figure 2: The structure of block in blockchain-based PKI

It must be mentioned that third-parties, who are called miners, still exist in blockchain-based PKI. However, their role is limited to ensuring security and integrity of the blockchain. Each miner in the system will be able to check the correctness of the posted information by using the public key contained in the posted information to verify the digital signature. After that, miners will try to solve the PoW (Proof Of Work) problem to get the chance to put this transaction in the block. The first miner to solve the PoW problem will broadcast the result to the entire network. All other miners will be easily able to check the correctness of the result and move onto solving the next PoW problem. A transaction fee will be paid for the first miner to solve the PoW problem, much like in Bitcoin, as an incentive to block miners.

3. PTAS:Privacy preserving Thin-client Authentication Scheme

According to section 2, users in blockchain-based PKI must download the entire blockchain into their devices in order to perform a series of operations. However, people using portable devices (such as smartphones) can't download the entire blockchain in their devices due to their limited storage space. Taking this kind of user into consideration, as for key registration and key update, they can just post the corresponding information to the blockchain and wait for the miner's confirmation. However, they can't perform searching a public key corresponding to a given identity without the help from full node users.

As we all know, increasingly importance has been attached to user's privacy. Many privacy-preserving technologies are applied in various fields, such as urban traffic systems [29], wireless sensor networks [30, 31], crowd sensing systems [32], machine learning [33], smart grid [34, 35] and cloud server [36–49]. In our system, thin-clients' privacy should also be protected. Specifically, thin-clients want to seek help from full node users but doesn't want full node users to be aware of it. We hence propose a Privacy-preserving Thin-client Authentication Scheme (PTAS) employing the method of private information retrieval [50] (PIR) in this section.

PIR was first proposed by Chor B et al. in 1995 [50]. It is utilized to protect the user's search privacy when retrieving data from the server, meaning that any other user, including the database server itself, can not track the user's search content. Considering such a scenario, a user wants to make a query to a database, but he does not want the database to know the information he is querying. For instance, an investor that queries the stock-market database for the value of a certain stock may wish to keep private the identity of the stock he is interested in.

The most common solution to this problem is that the user downloads all the information from the database and then conducts the query locally, but the communication complexity is very large. If each user goes to download all database data, it is obviously practically unacceptable.

Nowadays, with the rapid development of distributed databases, the same data is usually replicated at several databases, which raises hope to get around the difficulty of achieving privacy in the single database scenario. It may be possible to make queries to several servers and gets the desired information from the responses obtained, while each server (by observing only the query sent to him) gets no information about user's desired item. Specifically, PIR schemes allow a user to retrieve the ith bit of a k-bit database while keeping the value of i private. Fig. 3 describes a standard model of PIR.

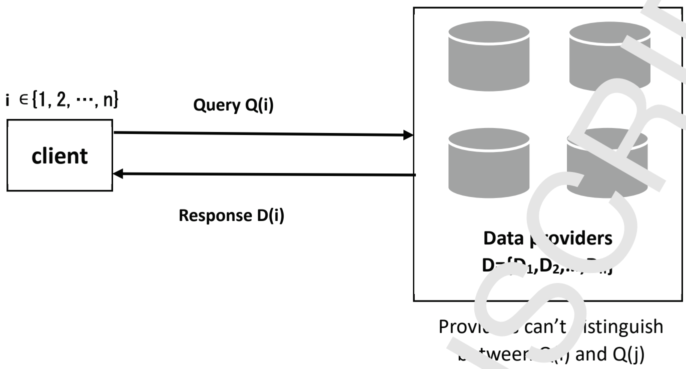


Figure 3: Private information retrieval model

In our system, m full node users are like m servers in PIR. k public keys are like k-bit database in PIR. Our goal is to retrieve the ith public key while keeping the value of i private.

3.1. Threat model

The objective of our PTAS is to protect the privacy of thin-client. The full node users in our system are defined as honest-but-curious. Specifically, the full node users will strictly follow the process of PTAS, after the execution of the protocol, there is no information disclosure except the execution result of the protocol. However, they may record all the information collected during the execution of the protocol and try to infer the thin-client's private data independently. It is worth mentioning that in our (m-1)-private PTAS (which will be proposed in **section 4**), we allow the full node users to collude with any set of less than (m-1) users to infer thin-client's privacy. In (m-1)-private PTAS, they can't get any information about thin-client at all even if (m-1) users collude together.

3.2. Notations and assumption

- C_1, C_2, ..., C_m represent $m = 2^d$ full node users.
- $\alpha(i)$ represents ith component of the vector α.

- For a vector α and a number i, let

$$\alpha \oplus i \triangleq \begin{cases} \alpha(i) = 1, & \text{if } \alpha(i) = 0 \\ \alpha(i) = 0, & \text{if } \alpha(i) = 1 \end{cases}$$

- Bob seeks m full node users for help where m is 2^d.
- These m nodes don't collude together.
- The length of the public key for all users is the same.

3.3. *The process of PTAS*

The detail of the process of authentication between Alice and Bob in PTAS is depicted as in Fig. 4.

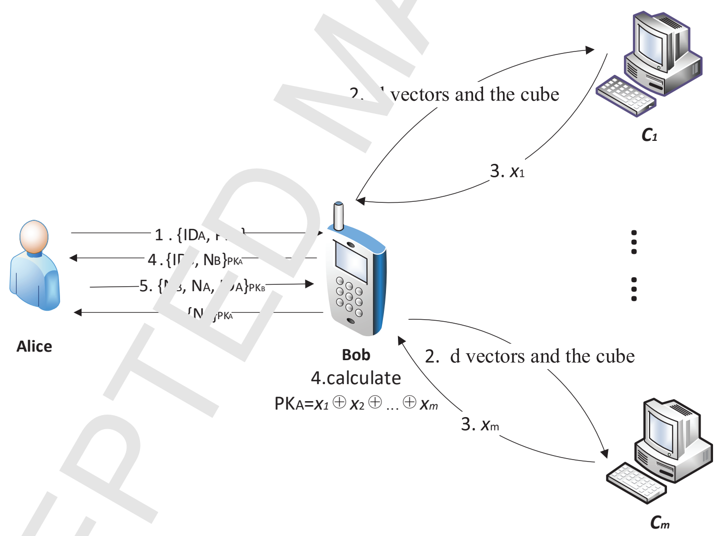


Figure 4: Authentication in PTAS

(1) Alice$\rightarrow$Bob: Alice sends her identity and public key (ID_A, PK_A) to Bob then waits for Bob's authentication.

(2) Bob⟶C_1, C_2, ..., C_m (m full nodes): Firstly, Bob randomly selects n-1 IDs (such as ID_G, ID_H, ID_I ...) and puts ID_A along with ID_G, ID_H, ID_I ... to a d-dimensional cube $[l]^d$ (we assume, without loss of generality that $n = l^d$). The arrangement is random and the position of each ID in the d-dimensional cube can be described as a d-tuple (j_1, j_2, ..., j_d), assume that the desired ID_A is associated with a d-tuple (i_1, i_2, ..., i_d). Secondly, Bob uniformly generates d random vectors α_1^0, α_2^0, ..., $\alpha_d^0 \in \{0,\ 1\}^l$ (the length of each vector is d and each component of the d vectors is set to 1 or 0 with the same probability). After that, Bob calculates α_1^1=$\alpha_1^0 \oplus i_1$, α_2^1=$\alpha_2^0 \oplus i_2$, ..., α_d^1=$\alpha_d^0 \oplus i_d$ and gets another d vectors. These *2d* vectors are paired in a natural way, namely, (α_1^0, α_1^1) (α_2^0, α_2^1), ..., (α_d^0, α_d^1). Finally, Bob sends d vectors and the d-dimensional cube $[l]^d$ to each full node. Specifically, for C_β where $\beta = \sigma_1\sigma_2...\sigma_d \in \{0,1\}^d$, Bob sends $\alpha_1^{\sigma_1}$, $\alpha_2^{\sigma_2}$, ..., $\alpha_d^{\sigma_d}$ and the cube.

(3) C_1, C_2, ..., C_m⟶Bob: Upon receiving $\alpha_1^{\sigma_1}$, $\alpha_2^{\sigma_2}$, ..., $\alpha_d^{\sigma_d}$, C_β finds the ID represented by (j_1, j_2, ..., j_d) where $\alpha_1^{\sigma_1}(j_1) = 1$, $\alpha_2^{\sigma_2}(j_2) = 1$, ..., $\alpha_d^{\sigma_d}(j_d) = 1$, retrieves the public keys corresponding to these IDs. Then performs exclusive-or of the bits on these PKs and the result is denoted as x_β and sent to Bob.

(4) Bob performs exclusive-or of the bits on x_1, x_2, ..., x_m and the final result is PK_A. He then sends Alice a message contained his identity ID_B and a nonce N_B which is encrypted with Alice's public key PK_A.

(5) Alice⟶Bob: Alice uses her secret key SK_A to decrypt the received message to get ID_B and N_B. If Alice is a full node user, she will traverse her own blockchain to find the corresponding PK_B to ID_B. If she is a thin-client, she will do the same thing as Bob to seek m random full node users for help. After finding PK_B, Alice sends Bob a message contained N_A, N_B and ID_A which is encrypted by Bob's public key PK_B.

(6) Bob⟶Alice: Bob uses his secret key SK_B to decrypt the received message to get N_A and checks if N_B is in this message. If so, Bob believes that Alice is, indeed, the owner of ID_A, and Bob will send Alice a message contained N_A encrypted by Alice's public key PK_A. If not, Bob will end the authentication.

(7) Alice uses her secret key SK_A to decrypt the received message and checks if N_A is in this message. If so, Alice believes that Bob is the owner of ID_B, the two sides

authenticated successfully. If not, she will end the authentication.

3.4. *Correctness of PTAS*

The correctness of the above scheme can be proved as follows. Considering the contribution of full node users (x_1, ..., x_m). x_β depends on the number of d vectors that contain the position (j_1, ..., j_d). It is not hard to see that (i_1, ..., i_d) is the only one position that is contained in an odd number of x_1, ..., x_m. This is because, for every $q \in [d]$, the value i_q appears in exactly one of the vectors α_q^0, α_q^1. Each of the other positions (j_1, ..., j_d) which are not equal to (i_1, ..., i_d), appears in an even number of x_1, ..., x_m. Therefore, in the final sum computed by Bob, the contribution of these positions is cancelled and the only value that remains is that of position (i_1, ..., i_d) which refers to PK$_A$.

Apparently, not only each of the vectors α_1^0, α_2^0, ..., α_d^0 is a random vector of $[l]$ but also each of the vectors α_1^1, α_2^1, ..., α_d^1 (since each α_q^1 is obtained by flipping the membership of one element in the random vector α_q^0). Thus, from the point of view of each full node user, it receives d random and independent vectors.

As a further consideration, assume that there are two identical queries in Bob's multiple queries or there are two users Bob and Calor conducting the same query. Since each query's d-dimensional cube is random and the position of ID$_A$ in each d-dimensional cube is different, full node users can't know that they have executed the same query. In summary, the thin-client user's query information has not been disclosed.

3.5. *An example of the process of PTAS*

For ease of understanding, an example is given in case of k = 9 and m = 4 (d = 2). The nine IDs are ID$_A$, ID$_B$ ... and ID$_I$. And they are put into a 3×3 grid, which is shown in Fig. 5.

ID_B	ID_A	ID_F
ID_I	ID_C	ID_E
ID_H	ID_D	ID_G

Figure 5: The 3×3 grid

Firstly, Bob generates two random vectors α_1^0, α_2^0 where α_1^0=(0,0,1) and α_2^0=(1,0,0). The position of ID$_A$ is (1, 2), so Bob calculates $\alpha_1^1 = \alpha_1^0 \oplus 1$ =(1,0,1), $\alpha_2^1 = \alpha_2^0 \oplus 2$ =(1,1,0). Then Bob sends two vectors and the 3×3 grid to each full node, where α_1^0 and α_2^0 are sent to C$_{00}$, α_1^0 and α_2^1 are sent to C$_{01}$, α_1^1 and α_2^0 are sent to C$_{10}$, α_1^1 and α_2^1 are sent to C$_{11}$.

Secondly, C$_{00}$ finds that $\alpha_1^0(3) = 1$ and $\alpha_2^0(1) = 1$, so he retrieves public key corresponding to ID$_H$(3, 1). Then, he performs exclusive-or of the bits on these PKs and the result (called x_{00}=PK$_H$) is sent to Bob; C$_{01}$ finds that $\alpha_1^0(3) = 1$, $\alpha_2^1(1) = 1$ and $\alpha_2^1(2) = 1$, so he retrieves public keys corresponding to ID$_H$(3, 1), ID$_D$(3, 2). Then, he performs exclusive-or of the bits on these PKs and the result (called x_{01}=PK$_H \oplus$PK$_D$) is sent to Bob; C$_{10}$ finds that $\alpha_1^1(1) = 1$, $\alpha_1^1(3) = 1$ and $\alpha_2^0(1) = 1$, so he retrieves public keys corresponding to ID$_B$(1, 1), ID$_H$(3, 1). Then, he performs exclusive-or of the bits on these PKs and the result (called x_{10}=PK$_H \oplus$PK$_B$) is sent to Bob; C$_{11}$ finds that $\alpha_1^1(1) = 1$, $\alpha_1^1(3) = 1$, $\alpha_2^1(1) = 1$ and $\alpha_2^1(2) = 1$, so he retrieves public keys corresponding to ID$_B$(1, 1), ID$_A$(1, 2), ID$_H$(3, 1) and ID$_D$(3, 2). Then, he performs exclusive-or of the bits on these PKs and the result (called x_{11}=PK$_B \oplus$PK$_A \oplus$PK$_H \oplus$PK$_D$) is sent to Bob.

Finally, Bob calculates $x_{00} \oplus x_{01} \oplus x_{10} \oplus x_{11}$ to obtain PK$_A$. Clearly, by doing so, none of the full node user can infer any information regarding which public key is desired by Bob throughout the process.

4. (m-1)-private PTAS

It should be pointed out that the m nodes in PTAS should not collude together. User's information may be deduced if several nodes colluded together. For example, if C_{00} and C_{01} in **section 3.5** collude together, according to the vectors they receive (α_2^0=(1,0,0) and α_2^1=(1,1,0)), they can infer that Bob's desired PK is in the second column of the cube.

In order to gain more security, in this section, we propose a (m-1)-private PTAS which means even (m-1) full node users collude together they still can not determine thin-client's desired information through what they receive. It is worthwhile mentioning that the restriction that the number of full node users must be a power of 2 no longer exists in (m-1)-private PTAS.

4.1. The process of (m-1)-private PTAS

The graphical representation of the process of (m-1)-private PTAS is shown in Fig. 6.

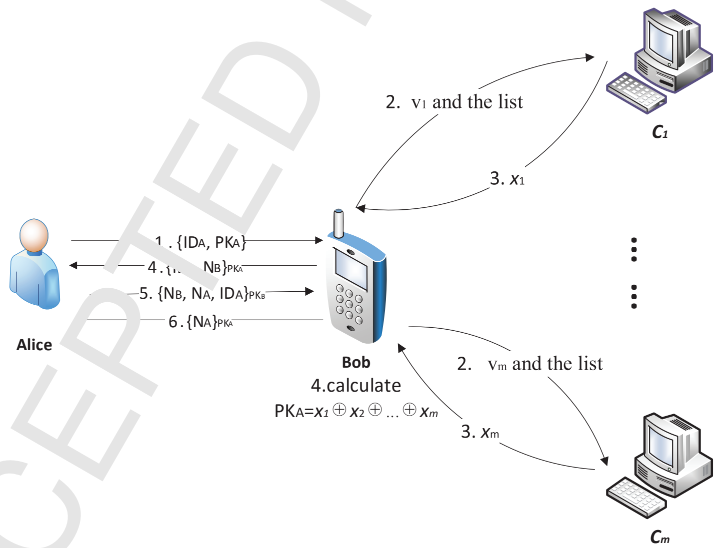


Figure 6: Authentication in (m-1)-private PTAS

(1) Alice⟶Bob: Alice sends her identity and public key (ID_A, PK_A) to Bob then waits for Bob's authentication.

(2) Bob⟶C_1, C_2, ..., C_m (m full nodes): Firstly, Bob randomly selects k-1 IDs (such as ID_G, ID_H, ID_I ...) and puts ID_A along with ID_G, ID_H, ID_I ... to a list. The arrangement is random and the position of each ID in the list can be described as a unique number, assume that the desired ID_A is the ith ID in the list. Secondly, Bob generates a basis vector e_i and uniformly generates m-1 random vectors v_1, v_2, ..., $v_{(m-1)} \in \{0, 1\}^k$ (the length of each vector is k and each component of the m-1 vectors is set to 1 or 0 with the same probability). After that, Bob calculates v_m=e_i⊕(v_1⊕...⊕$v_{(m-1)}$) which will also be a uniformly random vector. Finally, for every full node user C_β (β=1, 2 ... m), Bob sends v_β and the list.

(3) C_1, C_2, ..., C_m⟶Bob: Upon receiving v_β, C_β retrieves the public keys corresponding to *j*th ID where *j*th bit of v_β is 1. Then he performs exclusive-or of the bits on these PKs and the result is denoted as x_β and sent to Bob.

(4) Bob performs exclusive-or of the bits on x_1, x_2, ..., x_m and the final result is PK_A. He then sends Alice a message contained his identity ID_B and a nonce N_B which is encrypted with Alice's public key PK_A.

(5) (6) (7) steps are the same as PTAS.

4.2. Correctness of (m-1)-private PTAS

The correctness of (m-1)-private PTAS can be proved similarly. We assume here the average ID length is y-bit, the list can be described as a $k \times y$ matrix which donated as D. C_β performs v_β·D to find which public key he should retrieve and because (v_1⊕...⊕v_m)·D=e_i·D, the result of exclusive-or of all retrieved public keys is PK_A (as ID_A is the ith ID in the list).

It is important to highlight that this scheme is (m-1)-private because the m vectors are random and independent. Even if m-1 full nodes collude together, they can not obtain the information about the position of the user's desired ID at all.

4.3. An example of the process of (m-1)-private PTAS

To understand the process of (m-1)-private PTAS easily, an example is given in case of k = 5 and m = 5. The nine IDs are ID_A, ID_B ... and ID_E which are put into a list, which is shown in the following Fig. 7

ID_B	ID_C	ID_A	ID_E	ID_D

Figure 7: The list

Firstly, Bob generates a basis vector e_3=(0,0,1,0,0) and uniformly generates 4 random vectors v_1=(0,0,1,1,0), v_2=(1,1,1,1,0), v_3=(0,0,0,0,1), v_4=(1,0,1,0,1). After that, Bob calculates $v_5=e_3\oplus(v_1\oplus...\oplus v_4)$=(0,1,0,0,0). Finally, for every full node user C_β (β=1, 2 ... 5), Bob sends v_β and the list.

Secondly, upon receiving v_1=(0,0,1,1,0), C_1 retrieves PK_A and PK_E. Then he performs $PK_A \oplus PK_E$ and the result (called x_1) is sent to Bob; Upon receiving v_2=(1,1,1,1,0), C_1 retrieves PK_B, PK_C, PK_A and PK_E. Then he performs $PK_B \oplus PK_C \oplus PK_A \oplus PK_E$ and the result (called x_2) is sent to Bob; Upon receiving v_3=(0,0,0,0,1), C_3 retrieves PK_D and sends the result (called x_3) to Bob; Upon receiving v_4=(1,0,1,0,1), C_4 retrieves PK_B, PK_A and PK_D. Then he performs $PK_B \oplus PK_A \oplus PK_D$ and the result (called x_3) is sent to Bob; Upon receiving v_5=(0,1,0,0,0), C_5 retrieves PK_C and sends the result (called x_5) to Bob;

Finally, Bob calculates $x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5$ to obtain PK_A.

5. Security Analysis

5.1. The 51% attack

In terms of the security of the blockchain itself, it has numerous safety concerns [51] [52]. For instance, eclipse attack [53], sybil attack [54] and 51% attack. 51% attack is the most serious one among the various attacks, which may occur when a single miner node has exceptionally more computational resources than the rest of the network nodes, it will manipulate the entire network. More concretely, it can arbitrarily modify the blockchain information, such as inserting fraudulent transactions to blockchain, tampering with the content of transactions and hampering normal mining operations of other miners.

Although no 51% attacks have occurred in the bitcoin network since the first block was created and added to the blockchain, the risk does exist, especially in blockchains with a small number of nodes.

5.2. Dishonest node

Dishonest node means a malicious node attempts to cheat thin-client. It is important to point out that in our schemes, if there is a dishonest node in m full node users who returns incorrect data, because he doesn't know the contents of the data returned by other full nodes, he can't control the final calculation result of exclusive-or to deceive Bob. He can only make the final calculation result incorrect (become meaningless data). Bob will be deceived only if m full nodes are all dishonest and they collude with each other.

Suppose the dishonest nodes in the entire network account for $c\%$ and the selection of full node is completely random, then we can conclude that the probability of being cheated by collusion of the dishonest nodes is $(c\%)^m$. Assume there are 30%,20%,10% dishonest nodes in the entire network and they work together to cheat Bob, as the following TABLE 1 shows, Bob is less likely to be deceived as the value of m increases.

Table 1: Probability of being cheated

Value of m / The proportion of dishonest nodes	1	2	3	4	5
30% dishonest nodes	30%	9%	2.7%	0.81%	0.243%
20% dishonest nodes	20%	4%	0.8%	0.16%	0.032%
10% dishonest nodes	10%	1%	0.1%	0.01%	0.001%

In reality, owning such a large number of dishonest nodes is a costly task that requires a sufficient amount of resources. Additionally, a transaction fee for honest node will be helpful to incentive users to obey the rules. So it's nearly impossible for dishonest nodes to deceive Bob.

6. Performance Evaluation

In this section, we compare our scheme's functionality with latest proposed schemes IPK[23], Authcoin[21], Certcoin[18] and Cecoin[22]. Then we analyze computational overhead and communication overhead of our schemes.

6.1. Functionality

Table 2: Comparison of functionality

	IPK	Authcoin	Certcoin	Cecoin	Our scheme
Registration	✓	✓	✓	✓	✓
Revoking	×	×	✓	✓	✓
Updating	×	×	✓	✓	✓
Validation	✓	✓	✓	✓	✓
Thin-client	×	×	×	×	✓

As shown in TABLE 2, all of the above schemes have the two functions of registration and validation, only Certcoin [18], Cecoin [22] and our scheme have the function of revoking and updating. Besides, among these schemes, only our scheme has the function of thin-client.

6.2. Computational overhead

- Computational overhead of **thin-client**: In fact, computational overhead of thin-client can be measured by operations conduct by thin-client, which are shown in the following TABLE 3 and TABLE 4.

Table 3: Operations conduct by PTAS

Operation \ Scheme	PTAS (m = 4)	PTAS (m = 8)	PTAS (m = 2^d)
Generating a random number	1	1	1
Encryption	1	1	1
Decryption	1	1	1
Generating a random vector	2	3	d
Exclusive-or operation	5	10	$d + 2^d - 1$

Table 4: Operations conduct by (m-1)-private PTAS

Operation \ Scheme	(m-1)-private PTAS (m = 4)	(m-1)-private PTAS (m = 2^d)
Generating a random number	1	1
Encryption	1	1
Decryption	1	1
Generating a random vector	3	$2^d - 1$
Exclusive-or operation	8	$2 \times 2^d - 2$

For the above operations, we use JAVA to program and test the average time of these operations on mobile phone (in the case of *k=64*). The specific phone hardware parameters are as follow, CPU: MSM8996, 2.15Ghz, GPU: Adreno 530, 624MHz, Memory : 3GB RAM. Each operation is performed 10000 times and the average time of each operation is obtained and shown in TABLE 5.

Table 5: Average time of the operations

Operation	Average time of the operation (ms)
Generating a random number	0.00031
Encryption	0.2245
Decryption	1.2547
Generating a random vector	0.00223
Exclusive-or operation	0.0000[illegible]

The following Fig 8 shows the computational overhead of thin-client in both schemes.

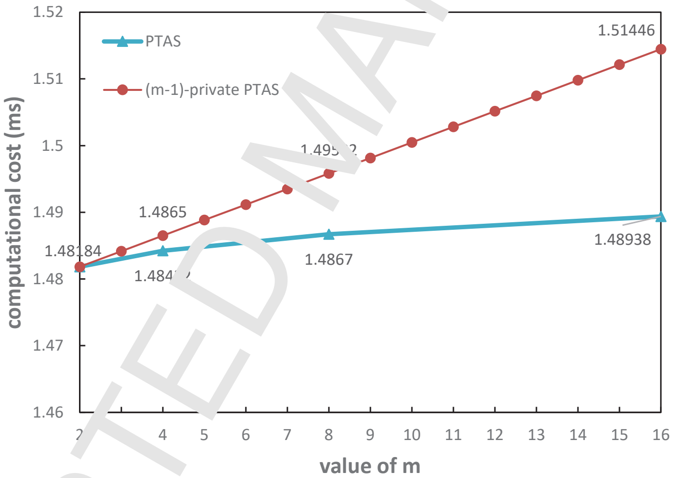


Figure 8: Computational overhead of the two schemes

From the results we have obtained, we can reach a conclusion that the total computational overhead of thin-client in PTAS and $(m$-1)-private PTAS is very close and the cost is in an acceptable range that it will not be a burden to a mobile phone user.

- Computational overhead of **full node users**: In fact, computational overhead of full node users can be measured the number of searches conducted by full node users. In PTAS, every full node user needs to retrieve *k/m* IDs' public keys on average, so the total number of searches is k **which is irrelevant to the value of m**. In (m-1)-private PTAS, every full node user needs to retrieve *k/2* IDs' public keys on average, so the total number of searches is *mk/2*. Fig. 9 shows the number of searches required for the schemes with the increase of k.

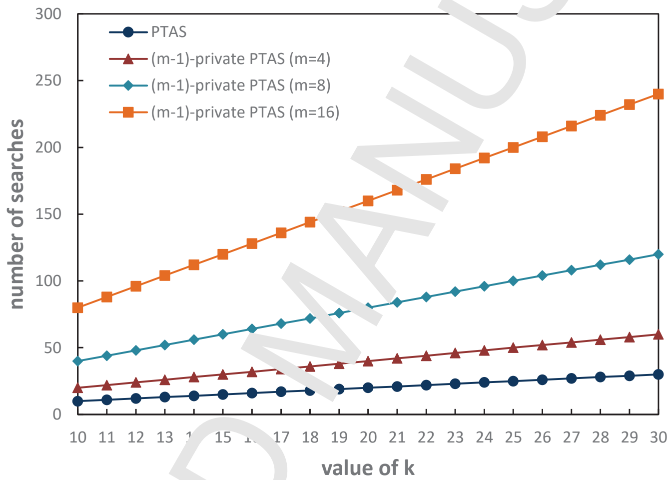


Figure 9: Number of searches

6.3. Communication overhead

The communication overhead between Bob and Alice in the two schemes are negligibly small, so we only analyze the communication overhead between the thin-client Bob and the full node users. We assume here the average ID length is y=64-bit, the length of public key is *t*=1024-bit.

- In the case of $m = 4$: In PTAS, the average communication overhead is $(8\lceil\sqrt{k}\rceil + 4yk)$ bit plus $4t$ bit on average. In (m-1)-private PTAS, the average communication overhead is $(4k + 4yk)$ bit plus $4t$ bit.

- In the case of $m = 2^d$: In PTAS, the average communication overhead is $(2^d d\lceil \sqrt[d]{k}\rceil + 2^d yk)$ bit plus $2^d t$ bit on average. In $(m$-1)-private PTAS, the average communication overhead is $(2^d k + 2^d yk)$ bit plus $2^d t$ bit.

In order to further compare communication overhead of the two schemes when m and k are large. TABLE 6 is developed and significant recommendations are made as follow.

Table 6: Communication overhead of the two schemes

Parameters		**Communication overhead**	
m	k	PTAS	$(m$-1)-private PTAS
32	20	74019	74368
32	40	115022	115968
32	60	156011	157568
32	80	196992	199168
32	100	237970	240768
32	120	278945	282368
64	20	148089	148736
64	40	230086	231936
64	60	312056	315136
64	80	394013	398336
64	100	475963	481536
64	120	557909	564736
128	20	296543	297472
128	40	460626	463872
128	60	624624	630272
128	80	788584	796672
128	100	952523	963072
128	120	1116446	1129472

On the basis of these results, it can be concluded that even if the values of k and m are large, communication overhead in $(m$-1)-private PTAS is almost the same as that in

PTAS. However, computational overhead of full node users and communication overhead in PTAS is a little smaller than that in (m-1)-private PTAS. The conclusion can be reached that compared with PTAS, (m-1)-private PTAS can provide higher security but its efficiency is a little inferior to the former.

7. Related Work

7.1. Decentralized PKI

Research about decentralized PKI began with Pretty Good Privacy (PGP) [55], which was initially designed for email users to exchange public keys without relying on a CA. More precisely, there is no central authority in the PGP trust model [56], each user is an authority itself and ensures many bindings between other users and their public keys. Users publish their self-signed certificates and use their own keys to sign other users' (ID, PK) pairs to confirm that they trust these (ID, PK) pairs. In PGP, this trust is recorded as a form of certificate, for example: $\text{Cert}_A(\text{B}, \text{PK}_B)$. Specifically, this certificate is signed by A's secret key SK_A and means that "A trusts B is binding with PK_B and A trusts B to issue certificates." In fact, if A wants to communicate with C, but he doesn't has a certificate issued by someone directly trusted by C. Then, A can search for a certificate chain from himself to C. If he can construct such a certificate chain, A will surely trust the binding between C and PK_C.

Nevertheless, there are some problematic disadvantages can't be overlooked in usability and security of PGP. Firstly, PGP does not define the method to construct the certificate chain. In practice, the approach is to implement a central keyserver that can store certificates to construct a certificate chain from A to C. Apparently, the method of keyservers, just like CAs in centralized PKI, still remains single point of failure. Besides, the correctness of the certificate information can not be guaranteed because of lacking incentive, some users may forge certificates for benefit. For instance, malicious users can generate a large number of nodes and connect them in such a way, which makes the network look like a group of trustful users. Finally, PGP struggles to preserve user's privacy, but certificate chains are easy to expose the privacy of the user's personal social network. Malicious users can know who you trust (most likely your friends or relatives) through your certificate chain.

KeyChains [57] is a peer-to-peer PKI system built on top of PGP model. In PGP model, the discovery and construction of certificate chains relies on centralized key-servers. However, KeyChains' unique query mechanism allows it to complete the task of generating and retrieving certificate chains in decentralized networks.

[58] and [59] presented an approach for a completely decentralized PKI which can serve as the basis for higher-level security service. In contrast to PGP model, they used a statistical method to provide an analytical model with provable guarantees. As for applications, they provided a layered model for P2P E-commerce, demonstrating the dependencies of various security related issues that can be built on top of a decentralized PKI.

Certificate Transparency (CT) project [60] was proposed by Google, whose goal is to provide an auditing and monitoring system that allows any user to identify whether a certificate was issued incorrectly or used maliciously through auditing the certificate logs, thereby enhancing the security of the system. In previous systems, fraudulent certificates could be overlooked for weeks or months, causing serious damage until discovered. In contrast, Certificate Transparency (CT) project can quickly and effectively identify the certificates that are issued in error. Early detection of suspicious certificates will be helpful for digital certificate authorities to react quickly and withdraw certificates.

7.2. Blockchain-based PKI

A considerable amount of research has been carried out on designing a blockchain-based PKI recently [18–25], Muneeb Ali et al. proposed Blockstack [20], which uses a global system-wide Namecoin blockchain to ensure the high-integrity of data and adapts Bitcoin's proof-of-work consensus mechanism to agree on the latest state of the system. Such approach is feasible as demonstrated by widespread adoption of Bitcoin and Bitcoin-like cryptocurrencies. They also presented various challenges to network reliability and security that they needed to overcome while registering and updating over 33,000 entries and 200,000 transactions on the Namecoin blockchain. Authcoin

[21] was proposed by Benjamin Leading et al. to implement a decentralized PKI, it uses a flexible challengeCresponse method to validate and authenticate when public keys are issued. Besides, they analyzed potential threats (such as sybil attacks) to Authcoin and found methods to mitigate them. CertCoin [18] was introduced by Conner Fromknecht et al. to implement a decentralized authentication scheme. Specifically, it is a public and decentralized authentication scheme which implements the idea of maintaining a public ledger of domains and their associated public keys to ensure identity retention. They proposed the use of cryptographic accumulators [61] to facilitate fast public key verification and applied the Kademlia DHT [62] for fast key lookup. Bo Qin et al. proposed Cecoin [22], which allows an identity to bind multiple public keys. Matsumoto et al.'s model Instant Karma PKI (IKP) [23] aims at achieving an improved PKI, which draws attention to using a blockchain-based mechanism to automatically respond to CA's misbehavior and gives incentives to those who once helped detect CA's misbehavior. Kumar et al. built a blockchain-based in VANET [19]. In these blockchain-based PKI systems, they didn't consider the issue of thin-clients (such as smartphone users), this type of users' storage space is so limited that they can't store the entire blockchain in their devices, their computing power is so weak that they can't afford complex operations. How to make such users run normally still requires further research.

8. Conclusion

In this paper, we have investigated security issues in IoT systems and suggest using PKI to assist the authentication of IoT devices. Then, we summarize the drawbacks of centralized PKI, PGP and latest proposed blockchain-based PKIs. To combat that, we creatively present a Privacy-preserving Thin-client Authentication Scheme (PTAS) using the method of PIR in blockchain-based PKI. For the purpose of improving security, we further propose a (m-1)-private PTAS which means user's privacy can be guaranteed when any (m-1) full node users collude together. Besides, security analysis and functional comparison are performed to demonstrate high security and rich functionalities of our schemes compared with existing schemes. Experiments are conducted to deliver that (m-1)-private PTAS sacrifices little efficiency in exchange for safety improvement. Finally, extensive experiments demonstrate that computational overhead of thin-client in the two schemes is in an acceptable range that will not be a burden to a smartphone user.

Acknowledgment

This work is supported by the National Key R&D Program of China under Grants 2017YFB0802300 and 2017YFB0802000, the National Natural Science Foundation of China under Grants 61802051, 61772121, 61728102, and 61472065, the Fundamental Research Funds for Chinese Central Universities under Grant ZYGX2015J056.

References

[1] L. Guo, K. Ota, M. Dong, Q. Li, T. Ye, J. Wu, J. Li, A secure mechanism for big data collection in large scale internet of vehicle, IEEE Internet of Things Journal PP (99) (2017) 1–1.

[2] H. Liu, H. Ning, Y. Zhang, Q. Xiong, L. T. Yang, Role-dependent privacy preservation for secure v2g networks in the smart grid, IEEE Transactions on Information Forensics & Security 9 (2) (2017) 208–220.

[3] H. Liu, H. Ning, Y. Zhang, L. T. Yang, Aggregated-proofs based privacy-preserving authentication for v2g networks in the smart grid, IEEE Transactions on Smart Grid 3 (4) (2012) 1722–1733.

[4] M. Tao, K. Ota, M. Dong, Z. Qian, Accessauth: Capacity-aware security access authentication in federated-iot-enabled v2g networks, Journal of Parallel and Distributed Computing 118 (2018) 107–117.

[5] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, Ddos in the iot: Mirai and other botnets, Computer 50 (7) (2017) 80–84.

[6] K. Zhao, L. Ge, A survey on the internet of things security, in: International Conference on Computational Intelligence and Security, 2014, pp. 663–667.

[7] R. Mahmoud, T. Yousuf, F. Aloul, I. Zualkernan, Internet of things (iot) security: Current status, challenges and prospective measures, in: Internet Technology and Secured Transactions, 2016, pp. 336–341.

[8] X. Wang, J. Zhang, E. M. Schooler, M. Ion, Performance evaluation of attribute-based encryption: Toward data privacy in the iot, in: IEEE International Conference on Communications, 2014, pp. 725–730.

[9] H. Ren, H. Li, Y. Dai, K. Yang, X. Lin, Querying in internet of things with privacy preserving: Challenges, solutions and opportunities, IEEE Network.

[10] C. Adams, S. Farrell, Internet x.509 public key infrastructure certificate management protocols, RFC Editor, 1999, pp. 82–89.

[11] C. Ellison, B. Schneier, Ten risks of pki: What you're not being told about public key infrastructure, Comput Secur J 16 (1) (2000) 1–7.

[12] H. Adkins, An update on attempted man in the middle attacks, http://googleonlinesecurity.blogspot.com/2011/08/update-on-attemptedman-in-middle.html.

[13] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Consulted.

[14] E. Mengelkamp, B. Notheisen, C. Beer, D. Dauer, C. Weinhardt, A blockchain-based smart grid: towards sustainable local energy markets, Computer Science-Research and Development (2) (2017) 1–8.

[15] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, Z. Sun, Blockchain-based dynamic key management for heterogeneous intelligent transportation systems, IEEE Internet of Things Journal PP (99) (2017) 1–1.

[16] A. Dorri, S. S. Kanhere, R. Jurdak, Towards an optimized blockchain for iot, in: IEEE Second International Conference on Internet-Of-Things Design and Implementation, 2017, pp. 173–178.

[17] M. Samaniego, R. Deters, Hosting virtual iot resources on edge-hosts with blockchain, in: IEEE International Conference on Computer and Information Technology, 2017, pp. 116–119.

[18] C. Fromknecht, D. Velicanu, S. Yakoubov, A decentralized public key infrastructure with identity retention, IACR Cryptology ePrint Archive.

[19] N. Kumar, R. Iqbal, S. Misra, J. J. P. C. Rodrigues, An intelligent approach for building a secure decentralized public key infrastructure in vanet, Journal of Computer and System Sciences 81 (6) (2015) 1042–1058.

[20] M. Ali, J. C. Nelson, R. Shea, M. J. Freedman, Blockstack: A global naming and storage system secured by blockchains, in: USENIX Annual Technical Conference, 2016, pp. 181–194.

[21] B. Leiding, C. H. Cap, T. Mundt, S. Rashidibajgan, Authcoin: Validation and authentication in decentralized networks, arXiv preprint arXiv:1609.04955.

[22] B. Qin, J. Huang, Q. Wang, X. Luo, B. Liang, W. Shi, Cecoin: A decentralized pki mitigating mitm attacks, Future Generation Computer Systemsdoi:10.1016/j.future.2017.08.025.

[23] S. Matsumoto, R. M. Reischuk, Ikp: turning a pki around with decentralized automated incentives, in: IEEE Symposium on Security and Privacy, 2017, pp. 410–426.

[24] M. Al-Bassam, Scpki: A smart contract-based pki and identity system, in: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, ACM, 2017, pp. 35–40.

[25] L. Axon, M. Goldsmith, Pb-pki: A privacy-aware blockchain-based pki, in: International Conference on Security and Cryptography, 2017, pp. 311–318.

[26] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, X. Lin, A privacy-preserving thin-client scheme in blockchain-based pki, in: Proceedings of GLOBECOM 2018, to appear.

[27] A. Shamir, How to share a secret, Communications of The ACM 22 (11) (1979) 612–613.

[28] L. Bassham, W. Polk, R. Housley, Algorithms and identifiers for the internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile, RFC Editor, 2002, pp. 338–342.

[29] S. Chang, H. Zhu, M. Dong, K. Ota, X. Liu, X. Shen, Private and flexible urban message delivery, IEEE Transactions on Vehicular Technology 65 (7) (2016) 4900–4910.

[30] J. Long, A. Liu, M. Dong, Z. Li, An energy-efficient and sink-location privacy enhanced scheme for wsns through ring based routing, Journal of Parallel & Distributed Computing 81-82 (C) (2015) 47–65.

[31] J. Long, M. Dong, K. Ota, A. Liu, Achieving source location privacy and network lifetime maximization through tree-based diversionary routing in wireless sensor networks, IEEE Access 2 (10) (2017) 633–651.

[32] G. Xu, H. Li, C. Tan, D. Liu, Y. Dai, K. Yang, Achieving efficient and privacy-preserving truth discovery in crowd sensing systems, Computers & Security 69 (2017) 114–126.

[33] Q. Zhang, L. Yang, Z. Chen, Privacy preserving deep computation model on cloud for big data feature learning, IEEE Transactions on Computers 65 (5) (2016) 1351–1362.

[34] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, X. Shen, Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid, IEEE Transactions on Parallel and Distributed Systems 25 (8) (2014) 2053–2064.

[35] H. Li, R. Lu, L. Zhou, B. Yang, X. Shen, An efficient merkle-tree-based authentication scheme for smart grid, IEEE Systems Journal 8 (2) (2014) 655–663.

[36] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, X. Lin, Healthdep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems, IEEE Transactions on Industrial Informatics.

[37] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, X. Zhang, Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation, IEEE Transactions on Information Forensics and Security 12 (3) (2017) 676–688.

[38] G. Xu, H. Li, Y. Dai, K. Yang, X. Lin, Enabling efficient and geometric range query with access control over encrypted spatial data, IEEE Transactions on Information Forensics and Security doi:10.1109/TIFS.2018.2868162.

[39] L. Zhang, L. Wei, D. Huang, K. Zhang, M. Dong, K. Ota, Medaps: secure multi-entities delegated authentication protocols for mobile cloud computing, Security & Communication Networks 9 (16) (2016) 3777–3789.

[40] S. Zhang, H. Li, Y. Dai, J. Li, M. He, R. Lu, Verifiable outsourcing computation for matrix multiplication with improved efficiency and applicability, IEEE Internet of Things Journal.

[41] H. Li, Y. Yi, Y. Dai, Y. Xiang, Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data, IEEE Transactions on Cloud Computing.

[42] H. Li, D. Liu, Y. Dai, T. Luan, S. Yu, Personalized search over encrypted data with efficient and secure updates in mobile clouds, IEEE Transactions on Emerging Topics in Computing 6 (1) (2018) 97–109.

[43] X. Chen, J. Li, J. Weng, J. Ma, W. Lou, Verifiable computation over large database with incremental updates, in: European Symposium on Research in Computer Security, 2014, pp. 148–162.

[44] X. Chen, J. Li, X. Huang, J. Ma, W. Lou, New publicly verifiable databases with efficient updates, IEEE Transactions on Dependable & Secure Computing 12 (5) (2015) 546–556.

[45] H. Li, D. Liu, Y. Dai, T. H. Luan, X. S. Shen, Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage, IEEE Transactions on Emerging Topics in Computing 3 (1) (2015) 127–138.

[46] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, X. S. Shen, Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data, IEEE Transactions on Dependable and Secure Computing 13 (3) (2016) 312–325.

[47] H. Li, Y. Dai, L. Tian, H. Yang, Identity-based authentication for cloud computing, in: IEEE International Conference on Cloud Computing, Springer, 2009, pp. 157–166.

[48] H. Li, D. Liu, Y. Dai, T. H. Luan, Engineering searchable encryption of mobile cloud networks: when qoe meets qop, IEEE Wireless Communications 22 (4) (2015) 74–80.

[49] X. Chen, J. Li, J. Ma, Q. Tang, W. Lou, New algorithms for secure outsourcing of modular exponentiations, IEEE Transactions on Parallel and Distributed Systems 25 (9) (2012) 2386–2396.

[50] B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, Private information retrieval, in: Symposium on Foundations of Computer Science, 1995, p. 41.

[51] H. Halpin, M. Piekarska, Introduction to security and privacy on the blockchain, in: IEEE European Symposium on Security and Privacy Workshops, 2017, pp. 1–3.

[52] G. Karame, On the security and scalability of bitcoin's blockchain, in: ACM Sigsac Conference on Computer and Communications Security, 2016, pp. 1861–1862.

[53] E. Heilman, A. Kendler, A. Zohar, S. Goldberg, Eclipse attacks on bitcoin's peer-to-peer network., in: USENIX Security Symposium, 2015, pp. 129–144.

[54] J. R. Douceur, The sybil attack, in: International Workshop on Peer-to-Peer Systems, Springer, 2002, pp. 251–260.

[55] P. R. Zimmermann, The official pgp user's guide, MIT Press, 1995.

[56] A. Abdul-Rahman, The pgp trust model, in: EDI-Forum: the Journal of Electronic Commerce, Vol. 10, 1997, pp. 27–31.

[57] R. Morselli, B. Bhattacharjee, J. Katz, M. Marsh, Keychains: A decentralized public-key infrastructure, Tech. rep., University of Maryland, College Park College Park United States (2006).

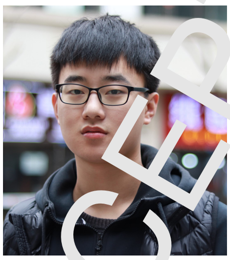
[58] A. Datta, M. Hauswirth, K. Aberer, Beyond" web of trust": Enabling p2p e-commerce, in: E-Commerce, 2003. CEC 2003. IEEE International Conference on, IEEE, 2003, pp. 303–312.

[59] K. Aberer, A. Datta, M. Hauswirth, A decentralized public key infrastructure for customer-to-customer e-commerce, International Journal of Business Process Integration and Management 1 (LSIR-ARTICLE-2005-001) (2005) 26–33.

[60] B. Laurie, Certificate Transparency, ACM, 2014.

[61] L. Reyzin, S. Yakoubov, Efficient asynchronous accumulators for distributed p-ki, in: International Conference on Security and Cryptography for Networks, Springer, 2016, pp. 292–309.

[62] P. Maymounkov, D. Mazieres, Kademlia: A peer-to-peer information system based on the xor metric, in: International Workshop on Peer-to-Peer Systems, Springer, 2002, pp. 53–65.



Wenbo Jiang received his B.S. degree in information security from University of Electronic Science and Technology of China (UESTC) in 2017. Currently, he is a master student at the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), China. His research interests include cryptography, blockchain, and the machine learning.

Hongwei Li is a Professor of University of Electronic Science and Technology of China, China. He received the Ph.D. degree from University of Electronic Science and Technology of China, China in 2008. He has worked as a Post-Doctoral Fellow in Department of Electrical and Computer Engineering at University of Waterloo for one year until Oct.2012.His research interests include network security, applied cryptography, and trusted computing. Dr. Li serves as the Associate Editor of Peer-to-Peer Networking and Applications, the Guest Editor for Peer to-Peer Networking and Applications Special Issue on Security and Privacy of P2P Networks in Emerging Smart City.

Guowen Xu received his B.S. degree in information and computing science from Anhui University of Architecture (AUA) in 2014. Currently, he is a Ph.D. student at the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), China. His research interests include cryptography, Searchable encryption, and the secure big data.

Mi Wen received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008. She is currently a Professor of the College of Computer Science and Technology with Shanghai University of Electric Power, Shanghai, China. Her research interests include privacy preserving in wireless sensor network, smart grid, etc. She serves as an Associate Editor of Peer-to-Peer Networking and Applications (Springer). She acts as the TPC Member of some conferences such as the IEEE INFOCOM, the IEEE ICC, and the IEEE GLOEBECOM, in 2012.

Guishan Dong received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China (UESTC). He is currently an Associate Director with the National Engineering Laboratory for Big Data application to improving the Government governance capacity in China, and a Chief Expert in network security with CETC Group. His main research areas include network security, cloud computing, big data, and network trust system. He was a recipient of the State Council Special Allowance Winner in 2016.

Xiaodong Lin received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2008. He is currently an Associate Professor at Wilfrid Laurier University, Canada. He was a recipient of the prestigious NSERC Canada Graduate Scholarships (CGS) Doctoral, and selected as university nominee for NSERC Doctoral Prize (Engineering and Computer Sciences category). He is a Fellow of the IEEE. His current research interest includes: computer forensics, software security and applied cryptography.

Highlights:

- For the first time, we propose a Privacy-preserving Thin-client Authentication Scheme (PTAS) in blockchain-based PKI.
- We present a (m-1)-private PTAS, in which even if (m-1) dishonest nodes colluded together, they still can't get any information about thin-client at all.
- Our schemes is equipped with high security, comprehensive functionality and desirable performance.