TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 13/14 pp 1128–1147
DOI: 10.26599/TST.2023.9010053
Volume 28, Number 6, December 2023

tsrCert: Traceable Self-Randomization Certificate and Its Application to Blockchain Supervision

Yan Zhu, Haibin Zheng*, Bo Qin*, Wanting Fu, Zhenwei Guo, Yujue Wang,
Qianhong Wu, Bingyu Li, and Xuan Ding

Abstract: Traditional public key infrastructure (PKI) only provides authentication for network communication, and the standard X.509 certificate used in this architecture reveals the user's identity. This lack of privacy protection no longer satisfies the increasing demands for personal privacy. Though an optimized anonymous PKI certificate realizes anonymity, it has the potential to be abused due to the lack of identity tracking. Therefore, maintaining a balance between user anonymity and traceability has become an increasing requirement for current PKI. This paper introduces a novel traceable self-randomization certificate authentication scheme based on PKI architecture that achieves both anonymity and traceability. We propose a traceable self-randomization certificate authentication scheme based on the short randomizable signature. Specifically, certificate users can randomize the initial certificate and public key into multiple anonymous certificates and public keys by themselves under the premise of traceability, which possesses lower computational complexity and fewer interactive operations. Users can exhibit different attributes of themselves in different scenarios, randomizing the attributes that do not necessarily need to be displayed. Through security and performance analysis, we demonstrate the suitability of the improved PKI architecture for practical applications. Additionally, we provide an application of the proposed scheme to the permissioned blockchain for supervision.

Key words: public key infrastructure; traceable self-randomization certificate; randomizable signature; anonymity and traceability; blockchain supervision

1 Introduction

With the rapid development of information and computer technology, dependence on network communication is becoming increasingly pervasive. Public key infrastructure (PKI) is an internationally standardized security mechanism and the main technical means for solving identity authentication issues when accessing online systems. The traditional PKI can implement the binding between users and keys through digital certificates, to verify users' identities and further ensure system security. However, according to the X.509

• Yan Zhu, Wanting Fu, Qianhong Wu, and Bingyu Li are with the School of Cyber Science and Technology, Beihang University, Beijing 100191, China. E-mail: {zhuyan; fuwanting; qianhong.wu; libingyu}@buaa.edu.cn.
• Haibin Zheng, Zhenwei Guo, and Yujue Wang are with Hangzhou Innovation Institute, Beihang University, Hangzhou 310051, China. E-mail: zhenghaibin29@buaa.edu.cn; zhenweiguo0724@163.com; wyujue2-c@my.cityu.edu.hk.
• Haibin Zheng is also with the Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China. E-mail: zhenghaibin29@buaa.edu.cn.
• Bo Qin is with the School of Information, Renmin University of China, Beijing 100872, China. E-mail: bo.qin@ruc.edu.cn.
• Xuan Ding is with the School of Software and BNRist, Tsinghua University, Beijing 100084, China. E-mail: dingxuan@tsinghua.edu.cn.
∗ To whom correspondence should be addressed.

Manuscript received: 2023-04-28; revised: 2023-05-25; accepted: 2023-05-26

standard[1], the subject domain of the digital certificate will be marked with the certificate holders' real name and other personal information, which is easy to be attacked to result in the disclosure of personal identity information. Moreover, if the same certificate is used in multiple authentication services, once these data are identified and accumulated, the possibility of privacy invasion will increase.

Given this context, the concept of anonymous credentials has emerged as a solution to users' privacy concerns. Anonymous credentials enable users to be authenticated without disclosing any information except for the fact that they hold a valid certificate. Two common strategies for implementing anonymous credentials are pseudonym systems and anonymous signatures. The former was first introduced by Chaum[2] in 1985, allowing users to interact with multiple organizations anonymously. The pseudonym certificate is similar to the traditional PKI certificate, except that the real name of user is replaced with an anonymous one in the subject domain. But this approach cannot provide unlinkability because a user's authentication information may be linked through pseudonyms with the same certificate. To avoid this problem, a user can be issued with multiple certificates with different pseudonyms, yet the management of multiple certificates is extremely inconvenient. Depending on the attributes, the user will need to manage different certificates. As the number of attributes contained in the certificate increases, the number of different certificates will increase. This approach substantially increases the number of certificates that the user needs to manage. The second technology, anonymous signature, such as ring signature[3], blind signature[4], etc., is very suitable for strong anonymity practical applications. Many anonymity mechanisms may not provide traceability due to their inherent system structure, which could lead to anonymity abuse and illegal activities such as fraud. The ring signature hides the user in a group. Thus the larger the signature group, the better the anonymity provided, but more signatures and verification of signature calculations are required. Therefore, these mechanisms often require complex calculations, which can be a significant obstacle to adoption and implementation.

Compared to ring signature, group signature[5] could further provide a tracing function to user's identity when illegal action occurs because of the existence of group manager. But group signature has a fatal flaw, that is, users in the group do not have their own public keys, only a common group public key instead. This fact makes group signature not applicable in many practical scenarios. For example, in the blockchain system, each user's public key is needed to verify the validity of a transaction.

At present, the conventional PKI only provides authentication function. The anonymous credential could provide authentication and anonymity but lacks tracking function. Group signature could achieve all the above functions but cannot provide user's own public key. In view of the current status, when designing a traceable anonymous certificate system based on PKI, a balance between the properties of anonymity, traceability, and system performance, is an important aspect to be considered.

1.1 Contribution of this work

To achieve the functions of anonymity, traceability, and high computational efficiency simultaneously for traditional PKI, we refine and present a traceable PKI authentication scheme based on the PKI system. We also propose an architecture of supervised blockchain by using the unique property of the traceable self-randomization certificate (tsrCert) scheme. The main contributions of our work are as follows:

- **Traceable PKI authentication.** We propose a traceable PKI authentication system architecture that achieves both anonymity and traceability simultaneously. Traceable PKI authentication scheme is a revolution to the traditional PKI, which perfectly solves the defects in traditional PKI and group signature, and enjoys their advantages. It achieves the authentication, anonymity, traceability of the certificate, and the randomization of the user's public key at the same time. The randomized public key can also be applied in many practical scenarios, such as anonymous encryption and anonymous signature.
- **Supervised with privacy-friendly blockchain.** Effective supervision has always been a problem to be solved in the blockchain. We propose an architecture of supervised blockchain based on the PKI system. It adds a traceable self-randomization certificate to the underlying architecture of the current permissioned blockchain, providing a balance between user privacy protection and identity supervision.

1.2 Techniques of this work

For specific implementation, we propose a tsrCert scheme based on the short randomizable signature scheme. This tsrCert scheme allows users to generate multiple anonymous certificates in view of an initial certificate and still maintain traceability. The implementation steps in supervised blockchain are also based on the tsrCert scheme. The main techniques of our work are as follows.

(1) Traceable self-randomization certificate. We propose a traceable self-randomization certificate scheme called tsrCert, which contains three entities and six algorithms. The proposed scheme involves three entities: the certificate user, the certificate authority (CA), and the certificate verification device. It differs from existing traceable anonymous certificate (TAC) systems in that it does not divide the CA into multiple roles. This different scheme requires fewer interaction calculations with the CA and has a shorter signature length, making it more practical for real-life applications. Additionally, we formally define two core security requirements of anonymity and traceability for the traceable anonymous certificate scheme.

(2) Short randomizable signature scheme. Short randomizable signature scheme is the core of our proposed traceable self-randomization certificate. With this approach, users can randomize the initial signature to multiple randomized signatures themselves by using one private key and can guarantee the validity of the randomized signatures at the same time.

1.3 Related work

PKI. The inherent information and data format in an X.509 certificate often cause privacy protection problems in traditional PKI. Anonymous credentials, introduced by Chaum[2], and first fully realized by Camenisch[6], is a centrally important building block under the PKI system. There has been a lot of research on anonymous credentials aimed at providing privacy protection to certificate holders, such as pseudonym system[7], the application of group signature to attribute-based anonymous credential systems[8], anonymous attribute certificates based on traceable signatures[9], and anonymous credential techniques using blind signature for privacy-preserving[10]. At the same time, there also exists some corresponding practical realization with systems, such as IBM's Identity Mixer[11] and Cinderella[12]. The Identity Mixer mechanism relies on the Camenisch–Lysyanskaya (CL) signature[13], zero-knowledge proof, and verifiable encryption to transform a credential into a presentation token. Cinderella mechanism[12] turns a shabby X.509 certificate into an elegant anonymous credential with the magic of verifiable computation. It provides a new format compiler to generate C code by composing X.509 templates for validating certificates. In general, multiple solutions related with privacy protection PKI have been researched, but unfortunately, none of the above mentioned schemes provides tracing capability for anonymous certificates. This status makes anonymous abusing being a potential threat[14]. If a CA is unable to map anonymous certificates to the actual users to whom they were issued, it creates a risk of anonymity abuse. Users may exploit their anonymity to engage in activities such as cybercrime and cyberattacks, as there are no resources available for the CA to track such behavior.

Traceable anonymous certificate. The formal notion of the traceable anonymous certificate was defined by Park[15] in the Request For Comment (RFC) 5636, which is published for recording standards, recommendations, and informational along with experimental documents. Some technical implementation details are not given within RFC 5636, though it defines a practical architecture of traceable anonymous certificates within the X.509 public key infrastructure. Afterward, Heijden[15] gave a specific discussion of procedural details, which divides CA into multiple sub-organizations, such as A and B, to achieve different functions. Azurmendi[16] constructed a coercion-resistant and easy-to-use Internet e-voting protocol based on a traceable anonymous certificate. But none of them has given specific instances. Apparently, some anonymous credential schemes based on group signature are also suitable for the application environment of the traceable anonymous certificate because of the nature of group administrators[17, 18]. It should be noted that these techniques are primarily designed for centralized organizational scenarios and often involve significant computational complexity. For example, in a privacy-preserving PKI design based on group signature[19], the user has proved the secret value s which is certified and valid every time after obtaining a signature σ on this secret value s, which is a rather complex statement to prove. Furthermore, the concept of PKI 2.0 was first proposed by Bouzefrane[20] in 2011, which is used to guarantee secure access to electronic services at a low cost. Then a report in 2013 proposed to improve the transparency of PKI 2.0, firmly anchoring PKI 2.0 in the Health Profession Card (HPC) and Medical ID Card of the German healthcare system[21]. After that, Boyen[22] decentralized PKI transparency, which is a decentralized client-based approach to enforcing transparency in certificate issuance and revocation. Li[23] proposed a

TAC scheme called EOLTAA based on zk-SNARK and utilized it in electronic voting. However, using zk-SNARK application authentication in a blockchain environment will incur significant overhead. The PTAP[24] protocol utilizes two dynamic anonymous and one static anonymous to generate a hash value that corresponds to an account.

Blockchain supervision. Blockchain is a decentralized infrastructure that emerged with the increasing popularity of Nakamoto[25]. Bitcoin system uses the public key as a pseudonym to provide anonymity and privacy protection for transaction users. While this approach provides anonymity, it also bypasses the supervision of relevant institutions, which could create opportunities for illegal or criminal activities. At present, the research techniques are relatively few. Some common techniques are information traceability techniques that use propagation rules to infer the transaction origination server node from the network level[26, 27], and transaction data analysis techniques for clustering anonymous users by using the association relationship between different transaction addresses from the data level[28, 29]. However, these are regulatory approaches suitable for the public chain scenario. For permissioned blockchain, the most famous one is the Membership Service Provider (MSP) in Fabric, which generates digital certificates to identify and manage the identities of members. Through a hierarchical structure consisting of root certificates, intermediate certificates, and signature certificates, MSP transforms user identities into organizational members of the blockchain, performing effective verification to achieve authentication and management of user identities. But this scheme does not have a supervisory function.

At the same time, there are other schemes available for supervising blockchain activities. In 2018, Zheng[30] proposed a linkable group signature scheme, which realizes the anonymous privacy protection of transaction senders. In 2019, Zheng[31] defined the cryptographic primitive of threshold indicative commitment, and proposed a zero-knowledge proof scheme to realize supervision of user identity. In 2020, Ma[32] proposed a traceable blockchain system SkyEye, which utilizes cryptographic techniques such as chameleon hash functions and zk-SNARKs zero-knowledge proofs. In 2021, Wang[33] divided the key technologies of blockchain supervision into the network layer, transaction layer, and application layer for analysis. In the same year, Bogatov[34] implemented transaction authorization based on a proxyable anonymous authentication scheme, which supports revocation and auditing of user identities. In 2022, Zhang[35] proposed a reliable blockchain traceability system supported by blockchain and CP-ABE encryption technology, which can set flexible access control policies for data.

In addition, a limitation of traditional PKI and anonymous credential based PKI is the requirement of a trusted credential issuer, which can raise the issue of excessive CA rights and become another potential threat to user privacy. Certainly, this problem can be resulted by distributing the authority of CA to other institutions, in which situation that all other roles should cooperate to complete the membership service. For example, the LocalPKI[36], decentralized anonymous credentials[37] and delegatable credentials[38]. These approaches can solve the single point problem of CA but with the cost of large calculations. In this paper, we do not pay much attention on this issue and assume CA is a trustworthy authority, where the operations of original certificate issue and anonymous certificate trace are both performed by the CA. This assumption allows us to avoid using expensive cryptographic blocks.

2 Preliminary

This section gives the basic building blocks to construct traceable self-randomization certificate and supervised blockchain system.

2.1 Bilinear pairings

Let G_1, G_2, and G_T be three cyclic groups of prime order p. A bilinear pairing is an efficient bilinear map $e : G_1 \times G_2 \rightarrow G_T$ with the following properties:

- **Bilinearity:** For all $u \in G_1, v \in G_2$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$;
- **Non-degeneracy:** For $u \neq 1_{G_1}$ and $v \neq 1_{G_2}$, $e(u, v) \neq 1_{G_T}$;
- **Computability:** e can be efficiently computed.

Bilinear pairings are generally classified into three basic types:

- **Type 1:** $G_1 = G_2$;
- **Type 2:** $G_1 \neq G_2$, but there is an efficiently computable homomorphism $\phi : G_2 \rightarrow G_1$;
- **Type 3:** $G_1 \neq G_2$, and there are no efficiently computable homomorphisms between G_1 and G_2.

2.2 BLS short signature scheme

Boneh–Lynn–Shacham (BLS) signature scheme was the first short signature constructed using bilinear pairings[39]. Its signature length is half of the traditional ECDSA signature and its construction is based on Gap Diffie-Hellman (GDH) groups. Let G_1, G_2 be GDH groups where $|G_1| = |G_2| = p, g_2$ be a generator of G_2, and $\mathrm{H} : \{0,1\}^* \rightarrow G_1$ be a hash function. A BLS signature scheme consists of three algorithms: Key generation, Signing, and Verification.

- **Key generation:** Randomly selects $x \leftarrow \mathbb{Z}_p$, computes $v \leftarrow g_2^x$. The public key is $v \in G_2$, the private key is x;
- **Signing:** Given private key $x \in \mathbb{Z}_p$, and message $m \in \{0,1\}^*$, computes $h \leftarrow \mathrm{H}(m) \in G_1$, and generates signature $\sigma \leftarrow h^x$;
- **Verification:** Given signature σ, message $m \in \{0,1\}^*$ and public key $v \in G_2$, computes $h \leftarrow \mathrm{H}(m)$ and verifies whether $e(\sigma, g_2) = e(h, v)$. If the verification succeeds, outputs 1.

Correctness analysis. If $\sigma = h^x$, then

$$e(\sigma, g_2) = e(h^x, g_2) = e(h, g_2^x) = e(h, v).$$

Security analysis. The security of BLS signature scheme is against existential forgery under adaptive chosen-message attacks in the random oracle model. The detailed proof process is given in Ref. [39].

2.3 A short randomizable signature scheme

We review the short randomizable signature scheme proposed by Pointcheval[40] based on Camenisch–Lysyanskaya (CL) signature.

- $\boldsymbol{pp \leftarrow Setup(1^k)}$**:** Let G_1, G_2, and G_T be Type 3 bilinear groups, $G_1^* = G_1/\{1_{G_1}\}$. The system public parameter $pp = (p, G_1, G_2, G_T, e)$;
- $\boldsymbol{(pk, sk) \leftarrow Keygen(pp)}$**:** Randomly chooses $\bar{g} \leftarrow G_2$, $(x, y) \leftarrow \mathbb{Z}_p^2$, computes $(\bar{X}, \bar{Y}) \leftarrow (\bar{g}^x, \bar{g}^y)$. The private key $sk = (x, y)$, public key $pk = (\bar{g}, \bar{X}, \bar{Y})$;
- $\boldsymbol{\sigma \leftarrow Sign(sk, m)}$**:** Randomly chooses $h \leftarrow G_1^*$, generates signature $\sigma = (\sigma_1, \sigma_2) = (h, h^{x+y\cdot m})$;
- $\boldsymbol{1/0 \leftarrow Verify(pp, pk, m, \sigma)}$**:** Given σ, checks whether $\sigma_1 \neq 1_{G_1}$ and $e(\sigma_1, \bar{X} \cdot \bar{Y}^m) = e(\sigma_2, \bar{g})$. If the verification succeeds, outputs 1, otherwise, outputs 0.

Correctness analysis. If $\sigma = (\sigma_1, \sigma_2) = (h, h^{x+y\cdot m})$, then

$$e(\sigma_1, \bar{X} \cdot \bar{Y}^m) = e(h^{x+y\cdot m}, \bar{g}) = e(\sigma_2, \bar{g}).$$

Randomization analysis. Given signature $\sigma = (\sigma_1, \sigma_2)$, the initial signature can be randomized by selecting a random $t \leftarrow \mathbb{Z}_p^*$, and generating randomized signature $\sigma' = (\sigma_1', \sigma_2') = (\sigma_1^t, \sigma_2^t)$. We can get the fact that the randomized signature is still a valid signature because it is equivalent to replacing $h \in G_1^*$ with $h' = h^t \in G_1^*$.

Security analysis. The simple signature scheme satisfies existential unforgeability under chosen message attack (EUF-CMA). The detailed proof process is given in Ref. [40].

2.4 Non-interactive zero-knowledge proof

The non-interactive zero-knowledge (NIZK) proof system is a protocol with no interaction between prover and verifier. For statement $s \in L$, witness w and relation R, $(s, w) \in R$, an NIZK proof protocol $NIZK\{s|(s, w) \in R\}$ consists of key generation Algorithm KGen, a prover P and a verifier V. The system model is as follows:

- $c \leftarrow KGen(1^k)$: Outputs the common reference string c;
- $\pi \leftarrow P(c, s, w)$: Computes the proof;
- $1/0 \leftarrow V(c, s, \pi)$: If accepts the proof, outputs 1, otherwise, outputs 0.

Property analysis. An NIZK proof protocol should satisfy the completeness, soundness, and zero-knowledge properties.

- **Completeness.** For all $s \in L$, $(s, w) \in R$, $Pr[c \leftarrow KGen(1^k), \pi \leftarrow P(c, s, w) : V(c, s, \pi) = 1] = 1$;
- **Soundness.** For malicious prover P' with $s \notin L$, $Pr[c \leftarrow KGen(1^k), (s, \pi) \leftarrow P'(c) : V(c, s, \pi) = 1] = neg(k)$;
- **Zero-Knowledge.** There exists a simulator $S = (S_1, S_2)$, for all polynomial time adversaries A, $Pr[c \leftarrow KGen(1^k), (s, w) \leftarrow A(c), \pi \leftarrow P(c, s, w) : V(c, s, \pi) = 1] \equiv Pr[(c, \tau) \leftarrow S_1(1^k), (s, w) \leftarrow A(c), \pi \leftarrow S_2(c, \tau, s, w) : V(c, s, \pi) = 1]$.

3 tsrCert Mechanism

This section presents an overview of the application of the tsrCert scheme, which includes a description of the system architecture and the security requirements that it addresses.

3.1 System architecture

Compared to the traditional PKI system, the upgraded PKI system has more comprehensive features. This design principle comes from Pointcheval's short randomizable signature approach[40] as we mentioned in Section 2.3, which is shown in Fig. 1. In Fig. 1, the conventional PKI system usually issues a certificate to a user, where the certificate corresponds to the user's public key and identity information. In the upgraded PKI system that includes tsrCert, each user can randomize their initial certificate using their private key to generate multiple randomizable certificates

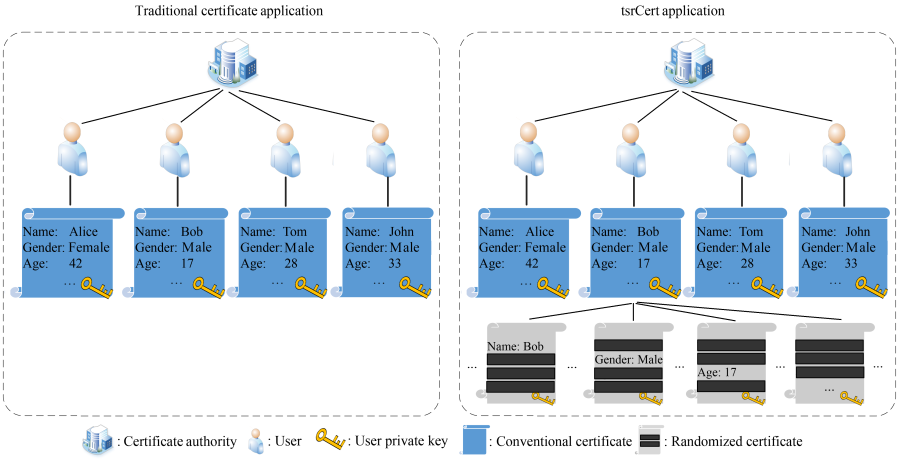


Fig. 1 Traditional certificate application vs tsrCert.

for anonymous authentication in different service environments. Users can expose different attributes in the certificate depending on the scenario in which they are employed. Additionally, this system ensures that the certificate authority can track the user's identity based on these corresponding randomizable public keys. This unique feature can offer greater anonymity for users while providing stronger supervision for the certificate authority.

The application of the tsrCert system mainly involves three types of entities: certificate user, certificate authority, and certificate verification device. The protocol flow is depicted in Fig. 2, and the responsibility and operations of entities are described as follows.

- **Certificate user.** The certificate user, denoted as U in our system, is the executor of certificate application and randomization. It first registers certificate with identities to the certificate authority (Step 1). After obtaining the certificate, it randomizes the initial certificate with its private key to multiple certificates to be authenticated anonymously by the remote certificate verification device (Steps 3 and 4).

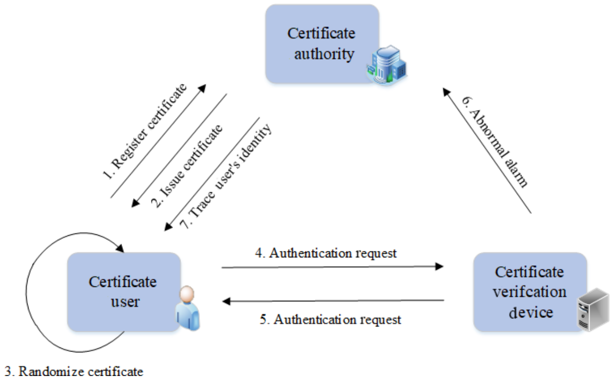


Fig. 2 Protocol flow of the tsrCert application.

- **Certificate authority.** The certificate authority, denoted as CA in our system, is the authority with certificate issuing and identity tracking for the user, which is the core of the PKI system. Upon receiving the U's registration application, it generates a signature based on its private key and U's public key and sends the signature as a certificate to U (Step 2). Finally, the CA traces and opens U's identity based on the user's randomized public key and associated parameters upon receiving a user's abnormal behavior alarm from the certificate verification device (Step 7).
- **Certificate verification device.** The certificate verification device, denoted as V in our system, is the remote verifier of randomized certificates. It verifies the validity of the randomized certificates (Step 5). If the verification succeeds, it indicates that the randomized certificate is still valid, and the anonymity feature can be achieved. Besides, if needed, this entity will report the abnormal alarm of the anonymous user to the certificate authority CA (Step 6).

3.2 Security requirements

To ensure the security of communication within the upgraded PKI, the system must meet the following security requirements.

- **Anonymity.** For PKI environment, the anonymity of the certificate holder is an essential security requirement. Such anonymity indicates the user's real-world identity is not revealed in the subject domain, and no information would be released from this anonymous certificate.
- **Randomization.** The PKI authentication process should preserve the acquisition requirement of randomizable anonymous certificates by registered certificate users. That is, a registered user can generate

any anonymous certificate by randomizing operations without the authorized authority.

- **Unlinkability.** To ensure user privacy, PKI-based authentication processes should prevent linkability between different randomizable certificates. This prevents malicious attackers or third parties from analyzing a user's identity by observing their various communications. The act of observing and recording a user's anonymous communication is considered an invasion of their privacy.
- **Traceability.** Traceability is a critical feature of PKI-based authentication processes as it enables the linkage between a user's real-world identity and their randomizable certificate. Without traceability, anonymous users may engage in illegal activities such as cybercrime, online fraud, and electronic theft. A trusted institution should be responsible for tracing a user's identity from their anonymous certificate.
- **Accountability.** To prevent the abuse of anonymous certificates, PKI-based authentication processes should incorporate accountability measures. This means that certificate users should be held responsible for their actions, even when using an anonymous certificate. To achieve this, there could be an authority agency, such as a supervision authority, responsible for collecting, collating, and interpreting evidence of users' abnormal behaviors. This approach motivates users to comply with the system since any deviation may be detected.
- **Non-repudiation.** The PKI authentication process must ensure non-repudiation for the actions of all parties involved. Non-repudiation means that no party can falsely deny their action or claim that their actions were performed by another when presented with evidence of their actions.

The above-stated security requirements are the common requirements of upgraded PKI. Since the pseudonym systems introduced by Chaum[2], various security requirements for anonymous certificates have been proposed, including anonymity, unforgeability, traceability, unlinkability, accountability, non-repudiation, customer profiling, and notification. Many of these requirements are interdependent and overlapping. To achieve anonymity, the features of unlinkability and randomization must be met. To achieve accountability, the feature of non-repudiation must be met. To achieve traceability, both unforgeability and accountability features must be met. Referring to the BMW security model which Bellare[41] defined for group signature, these above overlapping security requirements can also be extracted into fewer core security definitions.

4 Traceable Self-Randomization Certificate Model

The tsrCert scheme is the essential component of upgraded PKI-based authentication processes. In this section, we provide formal definitions for both the system model and the security model.

4.1 System definition

Definition 1 The tsrCert scheme mainly consists of the following six algorithms: setup algorithm Setup, key generation algorithm Keygen, issue algorithm Issue, randomization algorithm Randomize, verify algorithm Verify and trace algorithm Trace. The corresponding function of each algorithm is as follows:

- $\boldsymbol{pp, (cpk, csk) \leftarrow Setup(1^k)}$**:** On input a security parameter k, outputs the system public parameter pp and CA's key pair (cpk, csk).
- $\boldsymbol{(upk, usk) \leftarrow Keygen(pp)}$**:** On input system public parameter pp, outputs U's key pair (upk, usk).
- $\boldsymbol{(Cert, T) \leftarrow Issue(U(upk, usk), CA(cpk, csk))}$**:** The *Issue* algorithm is an interactive protocol which user U and certificate authority CA engaged in.

(1) Given system public parameters pp and CA's public key cpk, user U generates tracing parameter T, then sends U's public key upk and T to certificate authority CA;

(2) Given system public parameters pp and U's public key upk, certificate authority CA generates user's certificate $Cert$ using its private key csk, then sends $Cert$ to user U. At the same time, it records the user's identity $(upk, Cert)$ and tracing parameter T to certificate library.

- $\boldsymbol{(upk', Cert') \leftarrow Randomize(pp, upk, usk, Cert)}$**:** The user U makes random operation towards public key upk, and certificate $Cert$ using its private key usk. On the input system public parameter pp, public key upk, and certificate $Cert$, outputs randomized public key upk' and randomized certificate $Cert'$. Simultaneously, the user runs zero-knowledge proof $NIZK\{usk|(usk, w) \in R\}$ to prove the correct random operation was performed.
- $\boldsymbol{1/0 \leftarrow Verify(pp, cpk, Cert')}$**:** The certificate verification device V makes verify operation. On the input system public parameters pp, CA's public key cpk, and U's randomized certificate $Cert'$, outputs 1 if and only if the certificate is valid, otherwise outputs 0.
- $\boldsymbol{(upk, Cert) \leftarrow Trace(pp, upk', T)}$**:** The certificate authority CA makes trace operation using tracing parameter T. On the input system public parameters pp,

U's randomized public key upk', and tracing parameter T, outputs the registered user's identity, including initial public key *upk* and certificate *Cert*.

4.2 Security definitions

From the interdependent security features mentioned above, we can derive some core security requirements for a tsrCert scheme. We comply with the formal security definitions of accountable anonymous certificate proposed by Critchlow[42] and modify it to the model of traceable self-randomization certificate. A secure tsrCert scheme should satisfy the following core properties: correctness, anonymity, and traceability.

Correctness. A tsrCert scheme is correct if for all $k \in \mathbb{N}$, $(pp, (cpk,\ csk)) \leftarrow Setup(1^k)$, $(upk, usk) \leftarrow Keygen(pp)$, $(Cert, T) \leftarrow Issue(U(upk,\ usk), CA(cpk,\ csk))$, and $Cert' \leftarrow Randomize(pp, upk,\ usk, Cert)$, $upk' \leftarrow Randomize(pp,\ upk, usk, Cert)$. There exists $Verify(pp, cpk, Cert') = 1$ and $Trace(pp, upk', T) = upk$, the first shows that the randomized certificate is still a valid certificate. The second shows that the trace algorithm correctly tracks the real identity of the certificate user.

Anonymity. Anonymity is the basic security property for traceable self-randomization certificate system. In the tsrCert system environment, we define anonymity as the notion that an adversary cannot distinguish the identity of the user after given certificates nor judge whether these certificates are from the same user. The formal definition is given by experiment $\boldsymbol{Exp}_A^x(b, k)$.

Adversary A operates in two stages: choose and guess. In the choose phase, it is given some parameters produced by $Setup(1^k)$, outputs two identities upk_0, upk_1, a message m, and some auxiliary information *aux* to be used in the second stage. In the guess phase, it is given a challenge randomized certificate $Cert'_b$ formed by upk_b on message m where $b \leftarrow \{0, 1\}$, and says which identity was chosen. During the two stages, adversary A is allowed to query Trace oracle.

We define the advantage of the adversary as

$$Adv_{\text{tsrCert},A}^{\text{anony}}(k) =$$
$$Pr[\boldsymbol{Exp}_A^{\text{anony}}(1, k) = 1] - Pr[\boldsymbol{Exp}_A^{\text{anony}}(0, k) = 1].$$

Definition 2 Anonymity. A tsrCert scheme satisfies anonymity if for any polynomial-time adversary A, its advantage $Adv_{\text{tsrCert},A}^{\text{anony}}$ is negligible in the above anonymity attack experiment.

```
Exp_A^anony (b, k):
  $pp,\ (cpk, csk) \leftarrow Setup(1^k)$
  $(upk, usk) \leftarrow Keygen(pp)$
  $((upk_0, usk_0),\ (upk_1,\ usk_1),\ m,\ aux)$
  $\leftarrow A^{Keygen(\cdot), Trace(csk,\cdot)}(choose, pp)$
  $b \leftarrow \{0,\ 1\}$
  $Cert_b \leftarrow Issue(U(upk_b, usk_b), CA(cpk, csk))$
  $Cert'_b \leftarrow Randomize(pp, upk_b, usk_b, Cert_b)$
  $d \leftarrow A^{Trace(csk,\cdot)}(guess, Cert_b{}', aux)$
  return $d$
If $A$ did not query trace oracle with
$m, Cert'_b$ in the guess stage, return $d$, else
return 0.
```

Note: Anonymity security experiment.

Traceability. Traceability is another fundamental security property of a traceable self-randomization certificate system. In the tsrCert system, we define traceability as the ability of the CA to trace any valid certificate, even in the event of an adversary attempting to forge a certificate or corrupt users. The formal definition is given by experiment $\boldsymbol{Exp}_A^{trace}(k)$. Adversary A operates in two stages: corrupt and forge. In the corrupt phase, it is given some parameters produced by $Setup(1^k)$, outputs the identity list of corruption users L, the private key list of corruption users *usk*, corruption judgment Cort and some auxiliary information aux to be used in the second stage. In the forge phase, it outputs a forged certificate. If the corresponding identity information cannot be traced by certificate authority CA or the traced identity information is not in list L, then we say the adversary A succeeds in this attack game. During the two stages, A is allowed to query Issue and Randomize oracle.

Definition 3 Traceability. A tsrCert scheme satisfies traceability if for any polynomial-time adversary A, its advantage $Adv_{\text{tsrCert},A}^{trace}$ is negligible in the above traceability attack experiment.

We define the advantage of the adversary as

$$Adv_{\text{tsrCert},A}^{trace}(k) = Pr[\boldsymbol{Exp}_A^{trace}(k) = 1].$$

5 Traceable Self-Randomization Certificate Construction

In this section, we construct a concrete traceable self-randomization certificate scheme, and give the correctness, security and performance analysis of this construction.

5.1 Concrete scheme

According to the given traceable self-randomization certificate system model, we specifically present our

```
$\boldsymbol{Exp}_A^{trace}(k)$:
  $pp$, $(cpk, csk) \leftarrow Setup(1^k)$
  $(upk, usk) \leftarrow Keygen(pp)$
  $L \leftarrow \varnothing, USK \leftarrow \varnothing, Cort \leftarrow true$,
  $aux \leftarrow (L, USK, Cort)$,
  While ($Cort = true$) do
  $(Cort, upk_j, aux)$
  $\leftarrow A^{Issue(\cdot)}(corrupt, USK, aux)$
If $Cort = true$,
  then $L \leftarrow L \cup \{upk_j\}$,
  $USK \leftarrow USK \cup \{usp_j\}$,
End If
End While
  $(m, Cert') \leftarrow A^{Issue(\cdot), Randomize(\cdot)}(forge, aux)$
If $Verify(m, Cert', cpk) = 0$, then return 0
If $Trace(m, Cert', csk) = \perp$, then return 1
If there exists $upk_i$ such that the
following are true, then return 1, else
return 0
  1.  $Trace(m, Cert', csk) = upk_i$;
  2.  $upk_i \notin L$;
  3.  $(upk_i, m)$ was not queried to $Issue(\cdot)$
oracle by $A$.
```

Note: Traceability security experiment.

tsrCert construction in this part.

***pp*, (*cpk*, *csk*) ← *Setup*(1^k):** Initializes the system public parameter *pp*. Certificate authority *CA* generates key pairs (*upk*, *usk*).

(1) Let G_1, G_2, and G_T be cyclic groups with the same large prime p. $e: G_1 \times G_2 \rightarrow G_T$ is the type 3 pairing, where $G_1 \neq G_2$, and there is no valid homomorphic mapping between G_1 and G_2. Let $G_1^* = G_1/\{1_{G_1}\}$. Finally, outputs the system public parameter $pp = (p, G_1, G_2, G_T, e)$.

(2) Certificate authority CA randomly chooses $\bar{g} \leftarrow G_2$, $(x, y) \leftarrow \mathbb{Z}_p^2$, computes $(\bar{X}, \bar{Y}) \leftarrow (\bar{g}^x, \bar{g}^y)$, generates the private key $csk = (x, y)$, public key $cpk = (\bar{g}, \bar{X}, \bar{Y})$. CA keeps the private key csk and publishes the public key cpk.

(*upk*, *usk*) ← *Keygen*(*pp*): User U generates key pairs (*cpk*, *csk*) respectively.

User U randomly chooses $g \leftarrow G_1$, $\alpha \leftarrow \mathbb{Z}_p$, generates the private key $usk = \alpha$, public key $upk = (g, g^\alpha) \leftarrow G_1^2$.

(*Cert*, *T*) ← *Issue*(*U*(*upk*, *usk*), *CA*(*cpk*, *csk*)): User U and certificate authority CA make interaction to complete certificate issue.

(1) User U computes $T = \bar{g}^\alpha$, then sends public key *upk*, attribute m and tracing parameter T to certificate authority CA.

(2) After receiving $upk = (g, g^\alpha)$ and attribute m, CA randomly chooses $r \leftarrow \mathbb{Z}_p$, computes $X = g^r, Y = g^{mr}$, then generates a signature $\sigma = (X, X^x \cdot Y^y) = (g^r, g^{r(x+ym)}) = (\sigma_1, \sigma_2)$. The signature is regarded as user's certificate *Cert*, which means $Cert = (\sigma, m) = ((\sigma_1, \sigma_2), m)$.

(3) After sending certificate *Cert* to U, *CA* records the user's identity (*upk*, *Cert*) and tracing parameter T to certificate library.

(*upk′*, *Cert′*) ← *Randomize*(*pp*, *upk*, *usk*, *Cert*): The user U makes random operation.

(1) Given $Cert = ((\sigma_1, \sigma_2), m)$, user U first verifies the validity of given certificate. That is, checking whether $\sigma_1 \neq 1_{G_1}$ and $e(\sigma_1, \bar{X} \cdot \bar{Y}^m) = e(\sigma_2, \bar{g})$. If verification fails, it terminates.

(2) User U randomly chooses $\mu \leftarrow \mathbb{Z}_p$, computes $g_1 = g^\mu$, $X_1 = (g^\alpha)^\mu = g_1^\alpha$, and generates randomized public key $upk' = (g_1, X_1)$.

(3) User U randomly chooses $\nu \leftarrow \mathbb{Z}_p$, computes $\tilde{\sigma_1} = \sigma_1^\nu$, $\tilde{\sigma_2} = \sigma_2^\nu$, $\tilde{m} = m^\nu$, and generates part randomized certificate $Cert_0 = ((\tilde{\sigma_1}, \tilde{\sigma_2}), \tilde{m})$. Simultaneously, user U computes non-interactive zero-knowledge proof $NIZK\{usk|(usk, w) \in R\}$ to prove the correct random operation was performed (equivalent to prove it did perform effective randomization with the correct private key). It first computes $\sigma_3 = \tilde{\sigma_1}^m$, then computes $\pi = NIZK\{(m, \alpha)|\sigma_3 = \tilde{\sigma_1}^m \wedge X_1 = g_1^\alpha\}$. Finally, user U generates the randomized certificate $Cert' = (Cert_0, \sigma_3, \pi) = (\tilde{\sigma_1}, \tilde{\sigma_2}, \tilde{m}, \sigma_3, \pi)$, and sends $Cert'$ to the certificate verification device.

1/0 ← *Verify* (*pp*, *cpk*, *Cert′*): The certificate verification device V makes verify operation.

Given $Cert' = (\tilde{\sigma_1}, \tilde{\sigma_2}, \sigma_3, \pi)$, certificate verification device V verifies the validity of randomized certificate. That is, checking whether $\tilde{\sigma_1} \neq 1_{G_1}$ and $e(\tilde{\sigma_1}, \bar{X}) \cdot e(\sigma_3, \bar{Y}) = e(\tilde{\sigma_2}, \bar{g})$. If the verification succeeds, accept the certificate, otherwise, reject it.

(*upk*, *Cert*) ← *Trace*(*pp*, *upk′*, *T*): The certificate authority CA makes trace operation.

(1) After receiving tracing request, CA retrieves all tracing parameters $\hat{T} = \{T_1, T_2, \ldots, \infty\}$ from certificate library, where T_n is the tracking parameter of the n-th registered user.

(2) CA verifies the equation $e(X_1, \bar{g}) = e(g_1, T_i)$, for $i = 1, 2, \ldots, \infty$, successively by using the tracking parameters $\hat{T}$ and randomized public key upk'. If the equation holds for some T_i, then the registered user corresponding to this T_i is the user to be tracked.

5.2 Correctness analysis

The correctness of this tsrCert scheme is justified by the following equation.

- If user U has correctly randomized the original certificate, the randomized certificate is still a valid certificate. Because, if $\sigma_3 = \tilde{\sigma_1}^m$, then

$$e(\tilde{\sigma_1}, \bar{X}) \cdot e(\sigma_3, \bar{Y}) = e(\tilde{\sigma_2}, \bar{g})$$
$$e(\tilde{\sigma_1}, \bar{X}) \cdot e(\tilde{\sigma_1}^m, \bar{Y}) = e(\tilde{\sigma_2}, \bar{g})$$
$$e(\tilde{\sigma_1}, \bar{X} \cdot \bar{Y}^m) = e(\tilde{\sigma_2}, \bar{g})$$

- Certificate authority CA can certainly track the corresponding users by using tracing parameters $\hat{T}$ from certificate library. Because, for the n-th registered user, whose private key is α_i, and tracing parameter is $T_i = \bar{g}^{\alpha_i}$, we have

$$e(X_1, \bar{g}) = e(g_1^{\alpha_i}, \bar{g}) = e(g_1, \bar{g}^{\alpha_i}) = e(g_1, T_i).$$

5.3 Security analysis

We now prove the construction presented satisfies the security definition of anonymity and traceability. These securities are based on the Decision Linear (DLIN) assumption introduced by Boneh[43] and a modified Lysyanskaya, Rivest, Sahai, and Wolf (LRSW) assumption for Type 3 pairing groups put forth by Pointcheval[40].

Theorem 1 The proposed tsrCert construction satisfies anonymity under the DLIN assumption.

Proof Let A be an adversary against the anonymity of proposed tsrCert scheme with an advantage ε, then we construct an Algorithm B to solve the DLIN problem. Based on the existence of DLIN assumption (Lemma 1), it's easy to get the anonymity security of tsrCert construction. We first briefly introduce the DLIN assumption.

DLIN assumption. Suppose that (p, G_1, e) is defined the same as Setup algorithm in traceable self-randomization certificate scheme. The decision linear assumption states that, given $u, v, w, u^a, v^b, w^c \in G_1$, where $u, v, w \in G_1$, $a, b, c \in \mathbb{Z}_p$, it is hard to distinguish w^{a+b} from random w^c. More precisely, for all probabilistic polynomial time adversary A, the probability $|Pr[A(u, v, w, u^a, v^b, w^{a+b}) = 1] - Pr[A(u, v, w, u^a, v^b, w^c) = 1]|$ is negligible.

Lemma 1 The DLIN assumption holds for the generic bilinear group model that no adversary can solve the problem with probability greater than $8(q+9)^2/p$ after q queries to group oracle.

Proof The specific proof details of Lemma 1 can be found in Ref. [43]. ■

Now, we continue to prove Theorem 1 based on the existence of Lemma 1. Assume adversary A has broken the anonymity attack game, we will build Algorithm B to solve DLIN problem, which means, given public parameters $u, v, w, u^a, v^b, w^{a+b}$ and u, v, w, u^a, v^b, w^c, Algorithm B could distinguish w^{a+b} from random w^c. The interaction between Algorithm B and adversary A is as follows.

- After obtaining (p, G_1, e), $u, v, w, u^a, v^b, w^{a+b}$, and u, v, w, u^a, v^b, w^c, Algorithm B generates public parameters $pp = (p, G_1, G_2, G_T, e)$ to A;
- Adversary A makes queries of Keygen and Trace oracles. When querying for Keygen phase, B randomly chooses $\alpha \leftarrow \mathbb{Z}_p$, sets $g \leftarrow w$, sends $((w, w^\alpha), \alpha)$ to A as user's public key upk and private key usk; When querying for Trace phase, B gives upk to A;
- After adversary A selecting challenge bit $\hat{b} \leftarrow \{0, 1\}$, it makes queries of Issue and Randomize oracles. When querying for Issue phase, B generates challenge certificate $Cert \leftarrow (w, w^{a+b})$ to A when $\hat{b} = 1$ and generates challenge certificate $Cert \leftarrow (w, w^c)$ to A when $\hat{b} = 0$. When querying for Randomize phase, B randomly chooses $\nu \leftarrow \mathbb{Z}_p$, and gives $Cert^\nu$ as randomized certificate to A;
- A outputs a bit $\hat{b} \in \{0, 1\}$ as guess result.

From the above query-answer interaction, since for adversary A, w^{a+b} and w^c are valid certificates, if adversary A guesses the value of b with non-negligible probability greater than 1/2, then B could distinguish w^{a+b} from random w^c, further to solve the DLIN problem. By Lemma 1, we get the proposed traceable self-randomization certificate construction satisfies anonymity. ■

Theorem 2 The proposed tsrCert construction satisfies traceability under modified LRSW assumption.

Proof Let A be an adversary against the traceability of proposed tsrCert scheme with an advantage ε, then we construct an Algorithm C to solve the modified LRSW problem. Based on the existence of discrete-logarithm-based LRSW assumption (Lemma 2), it is easy to get the traceability security of tsrCert construction. We first briefly introduce the modified LRSW assumption.

Modified LRSW assumption. Suppose that (p, G_1, G_2, G_T, e) is a Type 3 pairing defined by the Setup algorithm. Let g be a generator of G_1, $\bar{g}$ be a generator of G_2. For $X = g^x, Y = g^y, (X, Y) \in G_1$ and $\bar{X} = \bar{g}^x, \bar{Y} = \bar{g}^y, (\bar{X}, \bar{Y}) \in G_2$, where x, y are randomly chosen from $\mathbb{Z}_p$, we define $O_{X,Y}(m)$ as an oracle that, on input $m \in \mathbb{Z}_p$, outputs a tuple

$T_t = (h, h^{x+my})$ for random $h \in G_1$. Then, given $(g, X, Y, \bar{g}, \bar{X}, \bar{Y})$ and unlimited access to oracle $O_{X,Y}(m)$, there is no probabilistic polynomial time adversary can generate a valid tuple T_t^* for m^*, with $h \neq 1_{G_1}$.

Lemma 2 The modified LRSW assumption holds for the generic bilinear group model that no adversary can generate a valid tuple with probability greater than $6(q_O + q_G)^2/p$ after q_O oracle queries and q_G group oracle queries.

Proof The specific proof details of Lemma 2 can be found in Ref. [40]. ■

Now, we continue to prove Theorem 2 based on the existence of Lemma 2. Assume adversary A has broken the traceability attack game, we will build Algorithm C to solve LRSW problem, which means, given public parameters (p, G_1, G_2, G_T, e) and $(g, X, Y, \bar{g}, \bar{X}, \bar{Y})$, Algorithm C could forge a valid tuple $T_t = (h, h^{x+my})$ for random $h \in G_1$. The interaction between Algorithm C and adversary A is as follows.

- After obtaining public parameters (p, G_1, G_2, G_T, e), Algorithm C forwards it to A as *pp*. Besides, C sends the received parameters $(g, X, Y, \bar{g}, \bar{X}, \bar{Y})$ to A as challenge parameters, where $X = g^x, Y = g^y$, $(X, Y) \in G_1$.
- Adversary A makes queries of Issue and Randomize oracles. When querying for Issue phase, C operates $Cert \leftarrow T_t$, and sends it to A; When querying for Randomize phase, C randomly chooses $\nu \leftarrow \mathbb{Z}_p$, and sends T_t^{ν} as randomized certificate to A.
- Finally, A outputs a forged randomized certificate based on the challenge parameters and query oracle.

From the above query-answer interaction, an Issue oracle query is perfectly equivalent to the oracle $O_{X,Y}(m)$. Then, if adversary A successfully attacks the traceability experiment, Algorithm C could further solve the LRSW problem. By Lemma 2, we get the proposed traceable self-randomization certificate construction satisfies traceability. ■

5.4 Performance analysis

In this section, we analyze the performance of our tsrCert construction based on short randomizable signature and compare it with some other well known pairing-based anonymous credential schemes based on short group signature[43], Camenisch–Lysyanskaya signature[13], and randomized blind signature[44]. The comparison mainly focuses on the size of public key, secret key and signature, the cost of signature and verification, and some properties these schemes supported including anonymity, traceability, randomizable and constant-size. The results are given in Table 1 and Table 2.

From Table 1, we can see that, our tsrCert scheme possesses higher efficiency regardless of the signature size or the computational cost. Table 2 shows that our tsrCert scheme possesses more security properties, which is better than others. Here, $|pk|$, $|sk|$, and $|Sig|$ are the size of public key, secret key, and signature, respectively; *Sig.Cost*, *Verify.Cost* are the cost of signature and verification; $|G_1|$ and $|G_2|$ are the size of group G_1 and G_2; $|\mathbb{Z}_p|$ and $|\mathbb{Z}_q|$ are the size of $\mathbb{Z}_p$ and $\mathbb{Z}_q$; r is the number of messages; R_{G_1} and R_{Z_p} are the cost of generating a random element of G_1 and $\mathbb{Z}_p$; E_{G_1} and E_{G_2} are the cost of an exponentiation in G_1, G_2; P and H are the cost of a pairing and hash computation.

We continue to give the data structure of certificates constructed in the above scheme. Figure 3 shows the specific format of initial certificate and randomizable certificate. Among them, the initial certificate has the same data format as the X.509 certificate in traditional PKI, and the traceable self-randomization certificate proposed in our upgraded PKI is further modified based on it, adding zero-knowledge proof to hide the user subject information. During the implementation process, user first registers with the CA to obtain the initial certificate, and then calculates the randomizable certificates for different application scenarios. In upgraded PKI, the randomizable certificate simultaneously ensures the anonymity of user identity and the traceability of CA to user.

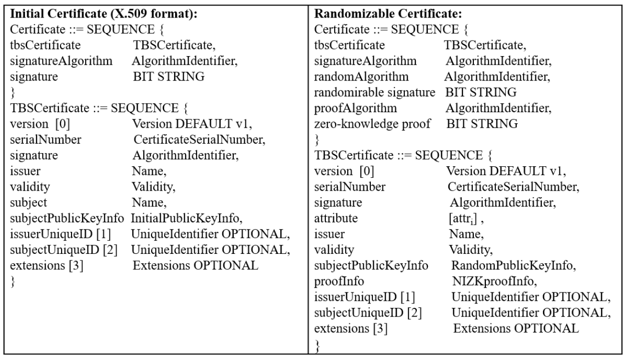
From Fig. 3, we notice that the data formats of these two certificates are basically identical and are both optimized on the basis of X.509 certificate, which indicates that the transfer from PKI to upgraded PKI can be implemented simply and quickly. In addition,

Table 1 Efficiency comparison with related works.

Scheme	$\lvert pk\rvert$	$\lvert sk\rvert$	$\lvert Sig\rvert$	$Sig.Cost$	$Verify.Cost$
SGS[43]	$4\lvert G_1\rvert + 2\lvert G_2\rvert$	$\lvert \mathbb{Z}_p\rvert + \lvert G_1\rvert$	$6\lvert \mathbb{Z}_p\rvert + 3\lvert G_1\rvert$	$6R_{Z_p} + 3E_{G_1}$	$5P + 4E_{G_1} + rH$
CLS[13]	$2\lvert G_1\rvert$	$2\lvert \mathbb{Z}_q\rvert$	$(2+r)\lvert G_1\rvert$	$1R_{G_1} + (1+r)E_{G_1}$	$4rP + rE_{G_2}$
RBS[44]	$2\lvert G_1\rvert$	$2\lvert \mathbb{Z}_q\rvert$	$(1+r)\lvert G_1\rvert$	$rR_{G_1} + 1E_{G_1}$	$3P + 4E_{G_1} + rH$
Coconut[45]	$3\lvert G_2\rvert$	$2\lvert \mathbb{Z}_p\rvert$	$2\lvert G_1\rvert$	$n(E\lvert G_1\rvert + E\lvert G_2\rvert)$	$2P + 2E_{G_1} + 2E_{G_2}$
Ours	$3\lvert G_2\rvert$	$2\lvert \mathbb{Z}_p\rvert$	$5\lvert G_1\rvert$	$1R_{G_1} + 4E_{G_1}$	$2P + rE_{G_2}$

Table 2 Functionality comparison with related works.

Scheme	Anonymity	Traceability	Randomizable	Constant-size
SGS[43]	✓	✓	×	✓
CLS[13]	✓	×	✓	×
RBS[44]	✓	×	✓	×
Coconut[45]	✓	×	✓	✓
Ours	✓	✓	✓	✓



Initial Certificate (X.509 format):

```
Certificate ::= SEQUENCE {
tbsCertificate          TBSCertificate,
signatureAlgorithm      AlgorithmIdentifier,
signature               BIT STRING
}
TBSCertificate ::= SEQUENCE {
version  [0]            Version DEFAULT v1,
serialNumber            CertificateSerialNumber,
signature               AlgorithmIdentifier,
issuer                  Name,
validity                Validity,
subject                 Name,
subjectPublicKeyInfo    InitialPublicKeyInfo,
issuerUniqueID [1]      UniqueIdentifier OPTIONAL,
subjectUniqueID [2]     UniqueIdentifier OPTIONAL,
extensions [3]          Extensions OPTIONAL
}
```

Randomizable Certificate:

```
Certificate ::= SEQUENCE {
tbsCertificate          TBSCertificate,
signatureAlgorithm      AlgorithmIdentifier,
randomAlgorithm         AlgorithmIdentifier,
randomirable signature  BIT STRING
proofAlgorithm          AlgorithmIdentifier,
zero-knowledge proof    BIT STRING
}
TBSCertificate ::= SEQUENCE {
version  [0]            Version DEFAULT v1,
serialNumber            CertificateSerialNumber,
signature               AlgorithmIdentifier,
attribute               [attr_i] ,
issuer                  Name,
validity                Validity,
subjectPublicKeyInfo    RandomPublicKeyInfo,
proofInfo               NIZKproofInfo,
issuerUniqueID [1]      UniqueIdentifier OPTIONAL,
subjectUniqueID [2]     UniqueIdentifier OPTIONAL,
extensions [3]          Extensions OPTIONAL
}
```

Fig. 3 Format of initial certificate and randomizable certificate.

for the certificate revocation operation, in our upgraded PKI mechanism, the CA organization can be managed hierarchically. Specifically, the CA is divided into the core certificate registration center and the randomirable certificate record center. The certificate registration center is used for issuing initial certificate and tracing user identity. The randomirable certificate archive center is used for users to record and revoke various random certificates. Only when user applies to revoke the initial certificate, the certificate registration center conducts the audit operation.

6 Application to Permissioned Blockchain for Supervision

In addition to anonymizing the user's certificate, the tsrCert mechanism also supports the randomization of the user's public key. This feature makes it suitable for use in blockchain, as it enables a balance between privacy protection and the supervision of transaction users, particularly in permissioned blockchain.

As mentioned above, for permissioned blockchains, the most well-known method is MSP in Fabric, but it lacks a strict supervisory function. In this section, we present an extended practical application of traceable self-randomization certificate to a membership service in a permissioned blockchain system. Specifically, we propose a blockchain supervision mechanism based on the traceable self-randomization certificate (tsrCert-BS). We begin by introducing the system architecture, followed by a description of the concrete system implementation. Finally, we summarize our analyses of the system's security, user privacy, and identity supervision.

6.1 System model

The design principle of supervised blockchain is based on widely recognized permissioned blockchain architecture. Figure 4 demonstrates the system model of proposed supervised blockchain.

In Fig. 4, compared to conventional blockchain, supervised blockchain adds a certificate authority which is independent of the original blockchain architecture to achieve supervision function through user's identity registration and certificate acquisition. At the same time, this system also supports stronger anonymity character by user's randomization operation after obtaining initial certificate.

Supervised blockchain system mainly contains three basic modules: transaction user (includes sender and receiver), certificate authority, and modified blockchain structure. Here, we briefly describe the responsibility and operations of these modules shown as follow.

The format of the original blockchain transactions and the format of our supervised blockchain transactions are depicted in Fig. 4. As shown in Fig. 4, compared to traditional blockchain transaction order, the difference lies in the signature algorithm and the randomization certificate. Our scheme maintains both anonymity and traceability while preserving the transaction format by introducing a novel cryptographic algorithm.

- **Transaction user.** The transaction user (includes sender and receiver), denoted as $\hat{U}$ in our system, is the transaction party on blockchain, and the executor of certificate application and certificate randomization at the same time. It first registers with identities to certificate authority. After obtaining the initial certificate, it randomizes the certificate with its private key to make the randomized certificate to be authenticated anonymously and adds it to the blockchain transaction.
- **Certificate authority.** The CA is the authority with certificate issuing and identity tracking for user, which is the core of supervised blockchain system. Upon receiving the $\hat{U}$'s registration application, it generates a signature based on its private key and $\hat{U}$'s public key, then sends the signature as certificate to $\hat{U}$. Finally, the CA opens $\hat{U}$'s identities based on the user's randomized public key and associated parameters after receiving this user's abnormal behavior alarm.

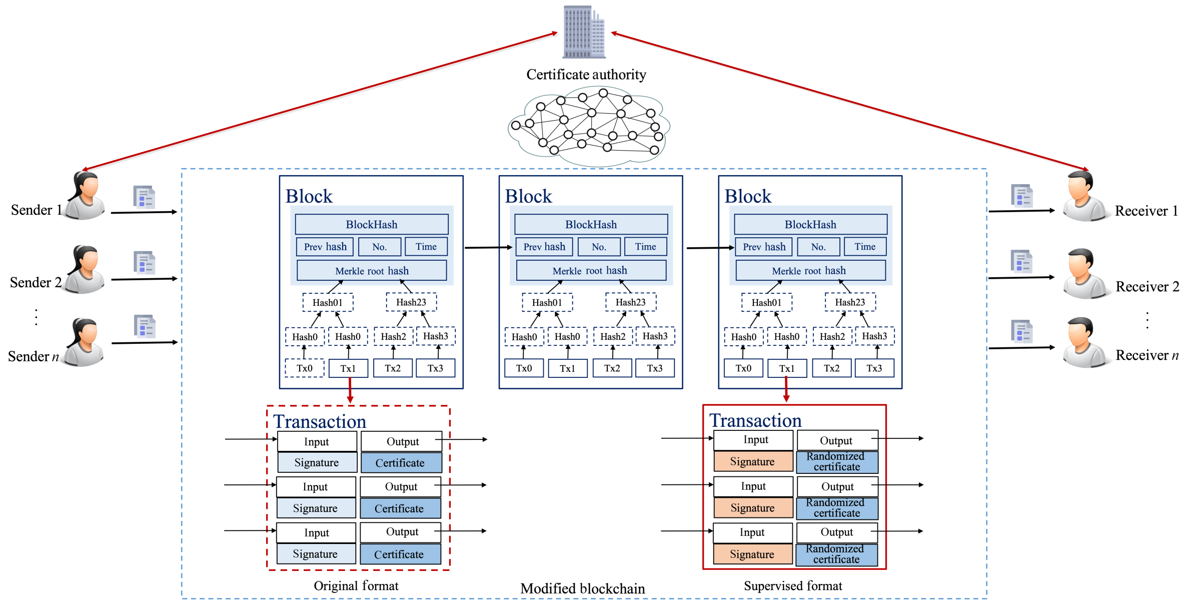


Fig. 4 System architecture of supervised blockchain.

- **Modified blockchain.** The modified blockchain, denoted as ModBlockchain in our system, is similar to the existing permissioned blockchain architecture. For example, each transaction chain can decide to adopt any consensus protocol, such as PoW, PoS, PBFT, that does not contradict the supervision operation for certificate authority CA. The only two differences are: (1) User's randomized certificate is added to each transaction; (2) In order to uniformly configure the operating environment of bilinear pairing, the traditional ECDSA scheme is replaced by a bilinear pair-based BLS signature for the underlying signature algorithm. Simultaneously, the miner nodes in blockchain are the verifiers of both transaction users' signatures and randomized certificates.

6.2 System implementation

According to the above system architecture of supervised blockchain and the concrete traceable self-randomization certificate scheme, this section introduces the scheme implementation of supervisioned blockchain based on traceable self-randomization certificate. This scheme constructs an identity management mechanism similar to Idemix in MSP, including eight algorithm steps of generating certificate authority key pairs, user registration application, initial certificate request, certificate request verification, generating initial certificate, generating anonymous certificate, anonymous certificate verification, and user identity tracing.

(1) Generate system parameters and CA key pairs

Let G_1, G_2, G_T be cyclic groups with prime p, Z_p be an integer group of order p. g_1 is a generator of G_1, g_2 is a generator of G_2, $e : G_1 \times G_2 \rightarrow G_T$ is the Type 3 pairing, $\mathrm{H} : \{0,1\}^* \rightarrow \{0,1\}^*$ is a hash function. The sysytem public parameter is $pp = (p, g_1, g_2, G_1, G_2, G_T, e, \mathrm{H})$.

CA randomly chooses integers $x \in Z_p$, $y \in Z_p$, the private key of CA is $ISK = (x, y)$. CA computes $(\bar{X}, \bar{Y}) = (g_2^x, g_2^y)$. CA sets the user attribute list $AttrName = [\mathrm{name}_1, \mathrm{name}_2, \ldots, \mathrm{name}_k]$, defining the attribute structure through user attribute list, where $len(AttrName) = k$. CA randomly chooses $r_1, r_2, r_3 \in Z_p$, computes $HSK = g_1^{r_1}$, $Hrand = g_1^{r_2}$, $\bar{g}_1 = g_1^{r_3}$, $\bar{g}_2 = \bar{g}_1^x$, $\bar{g}_3 = \bar{g}_1^y$.

Then CA makes a zero-knowledge proof of its private key, $NIZK\{ISK | \bar{X} = g_2^x \wedge \bar{g}_2 = \bar{g}_1^x, \bar{Y} = g_2^y \wedge \bar{g}_3 = \bar{g}_1^y\}$. The process is as follows.

- Randomly chooses $\hat{x} \in Z_p$, $\hat{y} \in Z_p$, computes $\bar{t}_{11} = g_2^{\hat{x}}, \bar{t}_{12} = \bar{g}_1^{\hat{x}}, \bar{t}_{21} = g_2^{\hat{y}}, \bar{t}_{22} = \bar{g}_1^{\hat{y}}$.
- Computes the challenge

$$c_x = \mathrm{H}(\bar{t}_{11}, \bar{t}_{12}, g_2, \bar{g}_1, \bar{X}, \bar{g}_2)$$

$$c_y = \mathrm{H}(\bar{t}_{21}, \bar{t}_{22}, g_2, \bar{g}_1, \bar{Y}, \bar{g}_3)$$

- Computes $s_x = \hat{x} + c_x \cdot x$, $s_y = \hat{y} + c_y \cdot y$.

Finally, CA outputs the key pairs of CA.

$ISK = (x,\ y)$

$IPK = (\bar{X}, \bar{Y}, AttrName, HSK, Hrand, \bar{g}_1, \bar{g}_2, \bar{g}_3, c_x, c_y, s_x, s_y)$.

Anyone can verify the proof. The process is as follows.

- Computes $\bar{t}'_{11} = g_2^{s_x} \cdot \bar{X}^{-c_x}$, $\bar{t}'_{12} = \bar{g}_1^{s_x} \cdot \bar{g}_2^{-c_x}$, $\bar{t}'_{21} = g_2^{s_y} \cdot \bar{Y}^{-c_y}$, $\bar{t}'_{22} = \bar{g}_1^{s_y} \cdot \bar{g}_3^{-c_y}$.

- Computes ${c_x}' = \mathrm{H}(\bar{t}'_{11}, \bar{t}'_{12}, g_2, \bar{g}_1, \bar{X}, \bar{g}_2)$, ${c_y}' = \mathrm{H}(\bar{t}'_{21}, \bar{t}'_{22}, g_2, \bar{g}_1, \bar{Y}, \bar{g}_3)$.
- Checks whether ${c_x}' = c_x$, ${c_y}' = c_y$. If both equations hold, the key pair of CA is correct.

(2) User registration application

User randomly selects $\alpha \in Z_p$, the user private key is $USK = \alpha$, user public key is $UPK = g_1^{\alpha}$. User calculates the tracing parameter $T = g_2^{\alpha}$, and sends the public key and tracing parameter to CA for registration.

(3) Initial certificate request

CA randomly selects an integer $IssuerNonce \in Z_p$ and sends it to user. User uses its private key α and a randomly selected integer $cerds \in Z_p$ to calculate the pseudonym $Nym = HSK^{\alpha} \cdot Hrand^{creds}$, then calculates the zero-knowledge proof about the private key α and the random number $creds$, denoted as $NIZK\{\alpha, creds | Nym = HSK^{\alpha} \cdot Hrand^{creds}\}$. The process is as follows.

- Randomly chooses $r_s, r_d \in Z_p$, computes
$t = HSK^{r_s} \cdot Hrand^{r_d}$.
- Computes the challenge $c_{sk} = \mathrm{H}(t, HSK, Nym, IssuerNonce)$.
- Computes $s_1 = r_s + c_{sk} \cdot \alpha$, $s_2 = r_d + c_{sk} \cdot creds$.

Finally, User outputs the certificate request $CertQst = (Nym, IssuerNonce, c_{sk}, s_1, s_2)$.

(4) Certificate request verification

After CA receives the user's certificate request, it first performs verification. The process is as follows.

- Computes $t'' = HSK^{s_1} \cdot Hrand^{s_2}$, $t' = t''/Nym^{c_{sk}}$.
- Computes ${c_{sk}}' = \mathrm{H}(t', HSK, Nym, IssuerNonce)$.
- Checks whether ${c_{sk}}' = c_{sk}$. If the equation holds, the user's certificate request is valid. Otherwise, CA refuses to issue a certificate for the user.

At the same time, CA stores the user's registered public key UPK and the tracing parameter T in the list *list*. They are used in user identity tracing phase to realize the supervision of CA.

(5) Generate initial certificate

Suppose the attribute submitted by the user is $attr = [attr_1, attr_2, \ldots, attr_k] \in Z_p^k$. For any $attr_j, (j = 1, 2, \ldots, k)$, CA selects a random integer $r_j \in Z_p$, computes $X_j = g_1^{r_j}, Y_j = UPK^{r_j} = g_1^{attr_j \cdot r_j}$. Then CA calculates the signature using the following formula for $j = 1, 2, \ldots, k$ and output $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_k)$.

$$\sigma_j = (\sigma_{j1}, \sigma_{j2}) = (X_j, {X_j}^x \cdot {Y_j}^y) = (g_1^{r_j}, g_1^{r_j(x + y \cdot attr_j)}),$$

Finally, CA outputs the certificate $Cert = (\sigma, attr_j)$.

After the user receives the certificate from CA, it first verifies the correctness of the certificate using the following formula. If the verification succeeds, the user accepts the valid certificate and stores the certificate locally.

$$e(\sigma_{j1}, \bar{X} \cdot \bar{Y}_j^{attr}) = e(\sigma_{j2}, g_2), \ j = 1, 2, \ldots, k$$

(6) Generate anonymous certificate

The user needs to show the certificate when transacting, it can specify the attributes to be shown according to different requirements. The user marks the subscript of the attribute to be hidden. For example, there are l attributes to be hidden, they can be marked as $HiddenIndices = [I_1, I_2, \ldots, I_l]$.

Specifically, user first randomizes the registration public key. User randomly selects an integer $u \in Z_p$, and calculates the randomized public key UPK'. It calculates $\xi = g_1^u$, $\eta = UPK^u = (g_1^{\alpha})^u = (g_1^u)^{\alpha} = \xi^{\alpha}$, $UPK' = (\xi, \eta)$.

Then User randomizes the signature of the attributes that need to be hidden in the certificate. User randomly selects an integer $v \in Z_p$, computes

$$\tilde{\sigma}_{j1} = \sigma_{j1}^{v}, \tilde{\sigma}_{j2} = \sigma_{j2}^{-v}, \tilde{\sigma}_{j3} = \tilde{\sigma}_{j1}^{attr_j},$$

$$\tilde{\pi}_{j4} = NIZK\{(attr_j, \alpha) | \tilde{\sigma}_{j3} = \tilde{\sigma}_{j1}^{attr_j} \wedge \eta = \xi^{\alpha}\},$$

$$j = 1, 2, \ldots, l,$$

$$\sigma'_j = (\tilde{\sigma}_{j1}, \tilde{\sigma}_{j2}, \tilde{\sigma}_{j3}, \tilde{\pi}_{j4}), \ j = 1, 2, \ldots, l,$$

where $\tilde{\pi}_{j4} = NIZK\{(attr_j, \alpha) \,|\, \tilde{\sigma}_{j3} = \tilde{\sigma}_{j1}^{attr_j} \wedge \eta = \xi^{\alpha}\}$. The process of proving the correctness of the $attr_j$ is as follows. The proof of the α is similar.

- User randomly selects integer $\hat{\alpha} \in Z_p$, and calculates auxiliary value $t_1 = \tilde{\sigma}_{j1}^{\hat{\alpha}}$.
- User uses the following formula to calculate the challenge value $c_{j\alpha} = \mathrm{H}(t_1, \tilde{\sigma}_{j1}, \xi, \tilde{\sigma}_{j3}, \eta, attr_j)$.
- Uer computes $s_{\alpha, j} = \hat{\alpha} + c_{j\alpha} \cdot attr_j$. Then $\tilde{\pi}_{j4} = s_{\alpha, j}$, for $j = 1, 2, \ldots, l$.

Finally, user outputs anonymous certificate $t' = (\sigma, attr_m) = (\sigma'_j, \sigma_m, attr_m)$ where $j = 1, 2, \ldots, l, m = l + 1, l + 2, \ldots, k$.

(7) Anonymous certificate verification

For the anonymous certificate presented by user, the verification process is as follows.

- Verifies equations

$$e(\tilde{\sigma}_{j1}, \bar{X}) \cdot e(\tilde{\sigma}_{j3}, \bar{Y}) = e(\tilde{\sigma}_{j2}, g_2), \ j = 1, 2, \ldots, l.$$

If the equations are true, the certificate presented by the user is valid. Otherwise certificate is invalid.

- Calculates $\tilde{\sigma}_{j1}^{s_{\alpha,j}} = \tilde{\sigma}_{j1}^{\hat{\alpha}} \cdot \tilde{\sigma}_{j1}^{c_{j\alpha}}$. If the equation holds, the certificate presented by the user is valid. Otherwise, the certificate is invalid.

(8) User identity tracing

When a dispute occurs, the verifier can send the anonymous certificate $Cert'$ to CA for arbitration. CA utilizes the tracing parameter list $\hat{T} = \{T_1, T_2, \ldots, T_n, \ldots\}$ to verify the equations $e(\eta, g_2) = e(\xi, T_i)$, for $i = 1, 2, \ldots, \infty$. If there is a T_i that makes the equation succeded, then the registered user corresponding to the T_i is the user to be traced. Then CA further determines the real identity of user.

6.3 System analysis

Our proposed supervised blockchain system can effectively solve the security, privacy and supervision issues that existed in the current blockchain and achieve a balance between the three properties.

- **Security.** For supervised blockchain, we only made a corresponding modification in the underlying data structure of existing blockchain, so the security of this system mainly depends on the security of existing blockchain and the security of the BLS signature scheme and traceable self-randomization certificate. As well known, the BLS signature scheme is strongly existentially unforgeable under an adaptive chosen message attack. Our proposed traceable self-randomization certificate scheme achieves the anonymity required by certificate security model. Hence, the security of our supervised blockchain system is equivalent to the security of current blockchain architecture.
- **Privacy.** Compared to the current blockchain, the supervised blockchain system can provide stronger privacy protection for user identities. This is because our traceable self-randomization certificate scheme ensures that no personal information is released, thanks to the randomizable certificate feature.
- **Supervision.** The main highlight of our proposed supervised blockchain is the traceability feature of the traceable self-randomization certificate scheme, which enables the system to achieve a supervision function that is not available in the existing blockchain system.

The differences between tsrCert-BS and Idemix. Our blockchain supervision mechanism based on traceable self-randomization certificate is similar to the Idemix mechanism in Fabric blockchain. But they also have the following differences.

(1) The Idemix mechanism does not provide a supervision function and generates unlinkable pseudonyms. However, tsrCert-BS mechanism combines anonymity and supervision with the PKI. The pseudonyms generated by tsrCert-BS are also unlinkable, ensuring user privacy while enabling monitoring and supervision.

(2) The Idemix mechanism can anonymize multiple messages simultaneously, which can be advantageous in scenarios where a large number of messages need to be processed. This approach can help to improve performance and reduce processing times. In contrast, the tsrCert-BS anonymizes a single message separately, which offers better operation atomicity. Additionally, certificate users can issue different anonymous certificates depending on the scenario, providing flexibility and adaptability to different use cases.

Furthermore, we give a comparison of the efficiency and performance of our tsrCert-BS scheme and the Idemix mechanism. The results are given in Table 3 and Figs. 5–7.

As shown in Table 3, in terms of computational cost, compared with Idemix mechanism, our tsrCert-BS scheme has lower computational complexity in the key generation phase and the certificate verification phase. But the exponential calculation of the initial certificate and anonymous certificate generation is too large. In terms of storage cost, compared with Idemix mechanism, the key length and certificate length of our tsrCert-BS scheme are slightly longer. But this scheme does not

Table 3 Efficiency comparison between tsrCert-BS and Idemix.

Phase	tsrCert-BS	Idemix
Key generation	$19E + 4H$	$(k + 11)E + 2H$
Initial certificate request	$4E + H$	$4E + H$
Request verification	$3E + H$	$3E + H$
Initial certificate generation	$2kE$	$(k + 2)E$
Anonymous certificate	$(4l + 2)E + 2H$	$(l + 14)E + 2H$
Anonymous certificate verification	$4E + 2H + 3P$	$(k + 10)E + 2H + 2P$
CA private key length	$2\|\mathbb{Z}_p\|$	$\|\mathbb{Z}_p\|$
User private key length	$\|\mathbb{Z}_p\|$	$\|\mathbb{Z}_p\|$
System public key length	$(k + 7)\|\mathbb{Q}\| + 4\|\mathbb{Z}_p\|$	$(k + 5)\|\mathbb{Q}\| + 2\|\mathbb{Z}_p\|$
Initial certificate length	$2k\|\mathbb{Q}\|$	$2\|\mathbb{Q}\| + 2\|\mathbb{Z}_p\|$
Anonymous certificate length	$4l\|\mathbb{Q}\| + \|\mathbb{Z}_p\|$	$(8 + l)\|\mathbb{Q}\| + 4\|\mathbb{Z}_p\|$

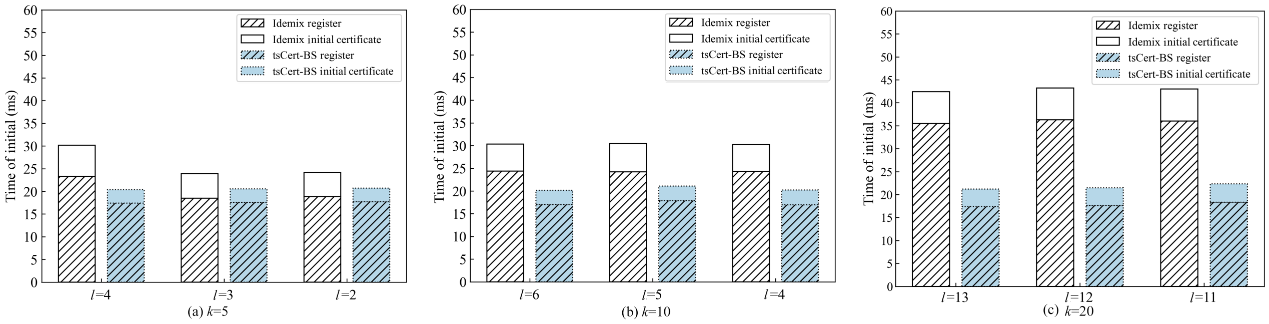


Fig. 5 Initial certificate.

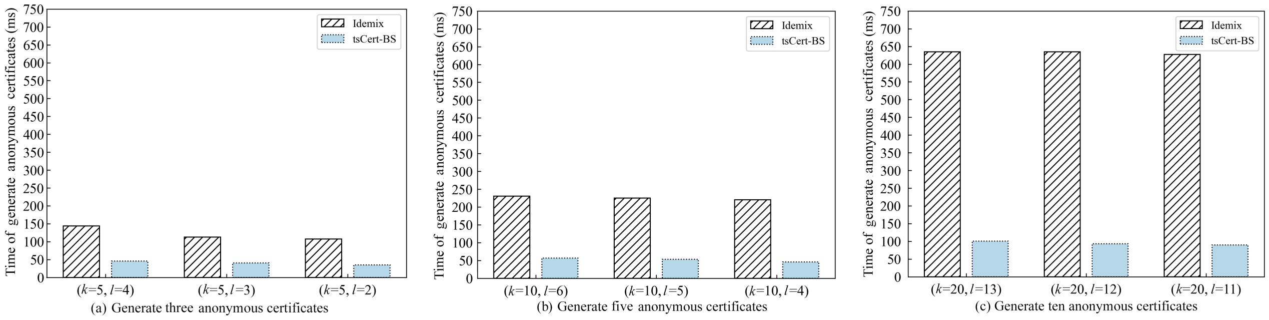


Fig. 6 Generate different anonymous certificates.

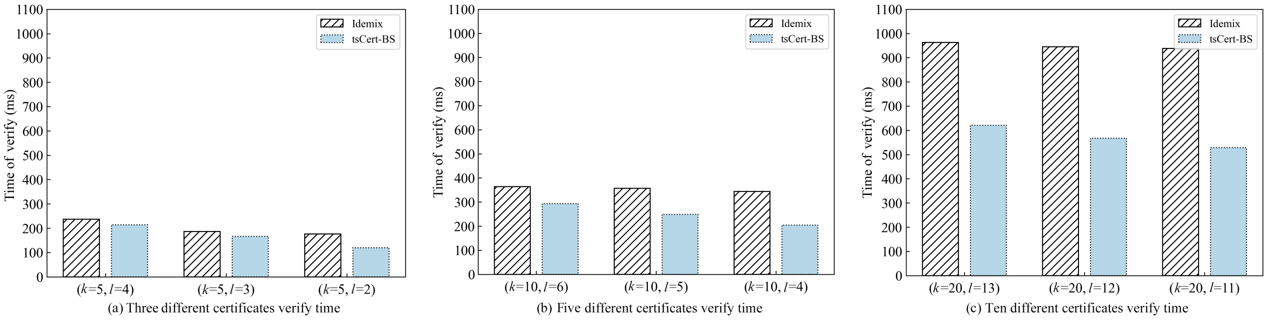


Fig. 7 Verify different anonymous certificates.

need to interact with the CA for each transaction, so that the total storage space in the same time is not much more than the Idemix mechanism. Here, E, H, P represent exponentiation operation, hash operation, and bilinear pair operation, respectively, $|\mathbb{Q}|$ is the size of the group $\mathbb{Q}$, $|\mathbb{Z}_p|$ is the size of the group $\mathbb{Z}_p$, k is the upper limit of all attributes, l is the upper limit of anonymous attributes.

The runtime of different stages in the tsrCert-BS scheme were compared to those in the Fabric blockchain Idemix mechanism. The simulation was implemented in Go by using a computer of 64 bits macOS 11.1 with 1 Core Intel i7 2.60 GHz and 32 GB RAM. Data are obtained by averaging 100 times. The unit of time is milliseconds.

First, we compared the time it takes to initialize certificates, including the time to register and generate initial certificates. To facilitate a comparison of anonymity and verification, certificates with different attribute scales were generated for analysis. Due to the fact that this process does not involve attribute blinding, the overall generation time increases as the number of attributes increases. As shown in Fig. 5, tsrCert-BS is faster than Idemix, which is due to the use of different signature algorithms.

Generating anonymous certificates is one of the important tasks of the scheme. We compared the time to generate different numbers of anonymous certificates. During the experiment, we calculated the time it takes to generate three different anonymous certificates with five attributes, five different anonymous certificates with ten attributes, and ten different anonymous certificates with twenty attributes.

From Fig. 6, it can be seen that tsrCert-BS has a lower time overhead than Idemix in generating different anonymous certificates. Notably, the time required for tsrCert-BS to generate anonymous certificates remains relatively stable, even as the number of certificates generated increases. On the other hand, the time it takes for Idemix to generate anonymous certificates increases significantly as the number of anonymous certificates generated increases.

These results are inherent in the design of the schemes. In Idemix, each generation of anonymous certificates requires authentication from the CA, and users need to interact with the CA to generate different signatures. In contrast, the anonymity process in the tsrCert-BS scheme is user-driven, allowing users to generate anonymous certificates independently of the CA without interaction. After CA authentication in the tsrCert-BS scheme, users can generate different anonymous certificates independently according to the application scenario.

While verifying a single certificate, tsrCert-BS does not offer an advantage over Idemix, as Idemix signs all the attributes in the certificate, whereas tsrCert-BS signs each attribute separately. This results in a larger number of signatures that need to be verified during the certificate verification process. However, in practical scenarios, tsrCert-BS has an advantage since certificates are typically used multiple times and are presented in different environments. In such cases, tsrCert-BS is advantageous due to its ability to handle repeated authentications among users.

According to the scenario of multiple communication between users, we conducted experimental analysis on certificate verification with 3 different certificates for 5 attributes, 5 different certificates for 10 attributes, and 10 different certificates for 20 attributes. Since tsrCert-BS verifies each attribute signature individually after each authentication, while Idemix verifies the entire signature, tsrCert-BS outperforms Idemix in this scenario, as demonstrated in Fig. 7.

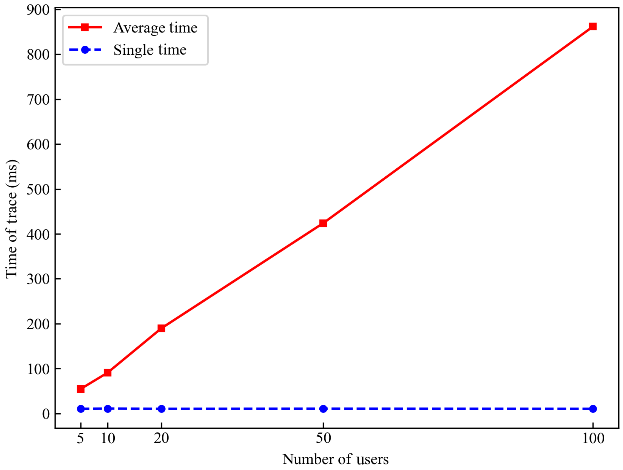


Fig. 8 tsrCert-BS trace time.

The tracking of anonymous certificates is a distinct feature of our scheme that is not available in Idemix. Anonymity revocation in tsrCert-BS is tied to the order in which users are registered. Therefore, in our experiment, we simulated the mathematical expectation value by designating a user in the middle of the user set to be traced. From Fig. 8, it is clear that the tracking time of our scheme increases significantly with the number of registered members, due to the traversal of tracking parameters. It may be possible to develop a plan in which the tracking time is linearly related to the number of members. The tracking time for each single user is shown in the Fig. 8. As the number of users changes, the scheme is constant for the tracking time.

7 Conclusion

We defined an upgrade PKI framework and proposed a specific traceable self-randomization certificate construction (tsrCert) based on short randomizable signature, then we applied the unique feature of the certification to the blockchain for realizing supervision. Simultaneously, we extracted and formally defined the security model for tsrCert scheme according to the general security requirements. This proposed tsrCert scheme can effectively achieve the anonymity and traceability functions for certificate user in current PKI, and improve the system efficiency by users' own randomization of certificates and reduced interactions with CA. In future work, we plan to develop more feasible schemes that balance user privacy and supervision while addressing the bottleneck in tracing efficiency.

Acknowledgment

This work was supported by the National Key R&D Program of China (No. 2020YFB1005600), Beijing Natural Science Foundation (No. M21031), the Natural Science Foundation of China (Nos. U21A20467, 61932011, 62002011, and 61972019), the Populus Euphratica Foundation (No. CCF-HuaweiBC2021009), the Open Research Fund of Key Laboratory of Cryptography of Zhejiang Province (No. ZCL21007), Zhejiang Soft Science Research Program (No. 2023C35081), and the Youth Top Talent Support Program of Beihang University (No. YWF-22-L-1272).

References

[1] D. Cooper, S. Santesson, S. Farrell, et al., Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, RFC 5280, https://www.rfc-editor.org/rfc/rfc5280, 2008.

[2] D. Chaum, Security without identification: Transaction systems to make big brother obsolete, *Commun. ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[3] R. L. Rivest, A. Shamir, and Y. Tauman, How to leak a secret, in *Proc. Int. Conf. Theory and Application of Cryptology and Information Security* (*ASIACRYPT*), Gold Coast, Australia, 2001, pp. 552–565.

[4] D. Chaum, Blind signatures for untraceable payments, in *Advances in cryptology*, Boston, MA, USA: Springer, 1983, pp. 199–203.

[5] D. Chaum and E. Heyst, Group signatures, in *Proc. Advances in Cryptology – Int. Conf. Theory and Application of Cryptographic Techniques* (*EUROCRYPT*), Brighton, UK, 1991, pp. 257–265.

[6] J. Camenisch and A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation, B. Pfitzmann, Ed. in *Lecture Notes in Computer Science*, Berlin, Germany: Springer, 2001, pp. 93–118.

[7] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, Pseudonym systems, in *Selected Areas in Cryptography*, Berlin, Germany: Springer, 2000, pp. 184–199.

[8] M. H. Au, P. P. Tsang, W. Susilo, and Y. Mu, Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems, in *Proc. Cryptographers' Track at the RSA Conference*, San Francisco, CA, USA, 2009, pp. 295–308.

[9] V. Benjumea, J. Lopez and J. M. Troya, Anonymous attribute certificates based on traceable signatures, *Inter. Res.*, vol. 16, no.2, pp. 120–139, 2006.

[10] F. Baldimtsi and A. Lysyanskaya, Anonymous credentials light, in *Proc. ACM SIGSAC Conf. Computer and Communications Security* (*CCS*), Berlin, Germany, 2013, pp. 1087–1098.

[11] J. Camenisch and E. Van Herreweghen, Design and implementation of the idemix anonymous credential system, in *Proc. 9th ACM Conf. Computer and communications security*, New York, NY, USA, 2002, pp. 21–30.

[12] A. Delignat-Lavaud, C. Fournet, M. Kohlweiss, and B. Parno, Cinderella: turning shabby X.509 certificates into elegant anonymous credentials with the magic of verifiable computation, in *Proc. 2016 IEEE Symp. on Security and Privacy (SP)*, San Jose, CA, USA, 2016, pp. 235–254.

[13] J. Camenisch and A. Lysyanskaya, Signature schemes and anonymous credentials from bilinear maps, in *Proc. 42nd Annu. Int. Cryptology Conf.* (*Crypto*), Santa Barbara, CA, USA, 2004, pp. 56–72.

[14] M. Mondal, Double-edged swords: The good and the bad of privacy and anonymity in social media, in *Proc. 28th ACM Conf. Hypertext and Social Media*, Prague, Czech Republic, 2017, pp. 1–2.

[15] S. Park, H. Park, Y. Won, J. Lee, KISA, and S. Kent, Traceable anonymous certificate (rfc5636), https://www.rfc-editor.org/rfc/rfc5636.html, 2011.

[16] I. Q. Azurmendi, J. L. Hern'andez-Ardieta, V. G. Martınez, L. H. Encinas, and D. A. Guardeno, A coercion-resistant and easy-to-use internet e-voting protocol based on traceable anonymous certificates, in *Proc. 3th Jornadas Nacionales de Investigación en Ciberseguridad*, Madrid, Spain, 2017.

[17] A. Miyaji and K. Umeda, A fully-functional group signature scheme over only known-order group, in *Proc. Applied Cryptography and Network Securit*y (*ACNS*), Huangshan, China, 2004, pp. 164–179.

[18] N. Kaaniche and M. Laurent, Attribute-based signatures for supporting anonymous certification, in *Proc. Computer Security - European Symp. on Research in Computer Security* (*ESORICS*), Heraklion, Greece, 2016, pp. 279–300.

[19] S. Lee, H. C. Kwon, and D. Seo, Privacy-preserving pki design based on group signature, https://ro.ecu.edu.au/ism/122/, 2011.

[20] S. Bouzefrane, K. Garri, and P. Thoniel, A user-centric pki based-protocol to manage fc2 digital identities, *Int. J. Comput. Sci.*, vol. 8, no. 1, pp. 1694–0814, 2011.

[21] H. J. Bickenbach, Common PKI 2.0, https://www.bundesnetzagentur.de/EVD/SharedDocuments/Downloads/QES/Common PKI v2.0 02.html, 2013.

[22] X. Boyen, U. Herath, M. McKague, and D. Stebila. Associative blockchain for decentralized PKI transparency, *Cryptogr.*, vol. 5, no. 2, pp. 1–14, 2021.

[23] P. Li, J. Lai, and Y. Wu, Event-oriented linkable and traceable anonymous authentication and its application to voting, *J . Inf. Secur. Appl.*, vol. 60, p. 102865, 2021.

[24] X. Liu, Y. Wang, Y. Li, and H. Cao, PTAP: A novel secure privacy-preserving & traceable authentication protocol in VANETs, *Comput. Netw.*, vol. 226, p. 109643, 2023.

[25] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org/bitcoin.pdf, 2008.

[26] P. Koshy, D. Koshy, and P. McDaniel, An analysis of anonymity in bitcoin using P2P network traffic, in *Financial cryptography and data security*, N. Christin and R. Safavi-Naini, Eds. vol. 8437, Berlin, Germany: Springer, 2014.

[27] A. Biryukov, D. Khovratovich, and I. Pustogarov, Deanonymisation of clients in bitcoin P2P network, in *Proc. 2014 ACM SIGSAC Conf. Computer and Communications Security*, Scottsdale, AZ, USA, 2014, pp. 15–29.

[28] B. Huang, Z. Liu, J. Chen, A. Liu, Q. Liu, and Q. He, Behavior pattern clustering in blockchain networks, *Multimed. Tools Appl.*, vol. 76, no. 19, pp. 20099–20110, 2017.

[29] D. Di Francesco Maesa, A. Marino, and L. Ricci, Data-driven analysis of Bitcoin properties: Exploiting the users graph, *Int. J . Data Sci. Anal.*, vol. 6, no. 1, pp. 63–80, 2018.

[30] H. Zheng, Q. Wu, B. Qin, L. Zhong, S. He and J. Liu, Linkable group signature for auditing anonymous communication, in *Proc. 23rd Australasian Conf.*

Information Security and Privacy (*ACISP*), Wollongong, Australia, 2018, pp. 304–321.

[31] H. Zheng, Q. Wu, Z. Guan, B. Qin, S. He, and J. Liu, Achieving liability in anonymous communication: Auditing and tracing, *Comput*. *Commun*., vol. 145, pp. 1–13, 2019.

[32] T. Ma, H. Xu, and P. Li, Skyeye: A traceable scheme for blockchain, https://eprint.iacr.org/2020/034, 2020.

[33] Y. Wang, G. Gou, C. Liu, M. Cui, Z. Li, and G. Xiong, Survey of security supervision on blockchain from the perspective of technology, *J*. *Inf. Secur*. *Appl*., vol. 60, p. 102859, 2021.

[34] D. Bogatov, AD. Caro, K. Elkhiyaoui and B. Tackmann. Anonymous transactions with revocation and auditing in hyperledger fabric, in *Proc. Cryptology and Network Security: 20th Int. Conf.* (*CNS*), Vienna, Austria, 2021, pp. 435–459.

[35] G. Zhang, X. Chen, B. Feng, X. Guo, X. Hao, H. Ren, C. Dong, and Y. Zhang, BCST-APTS: Blockchain and CP-ABE empowered data supervision, sharing, and privacy protection scheme for secure and trusted agricultural product traceability system, *Secur*. *Commun*. *Netw*., vol. 2022, pp. 1–11, 2022.

[36] J. G. Dumas, P. Lafourcade, F. Melemedjian, J. B. Orfila, and P. Thoniel, Localpki: A user-centric formally proven alternative to pkix, in *Proc.* 14th *Int. Conf. on Security and Cryptography* (*SECRYPT*), Madrid, Spain, 2017, pp. 1–18.

[37] C. Garman, M. Green, and I, Miers, Decentralized anonymous credentials, in *Proc. Annu. Network and Distributed System Security Symp.* (*NDSS*), San Diego, CA, USA, 2014, pp. 1–21.

[38] J. Camenisch, M. Drijvers, and M. Dubovitskaya, Practical uc-secure delegatable credentials with attributes and their application to blockchain, in *Proc. ACM SIGSAC Conf. Computer and Communications Security* (*CCS*), Dallas, TX, USA, 2017, pp. 683–699.

[39] D. Boneh, B. Lynn, and H. Shacham, Short signatures from the Weil pairing, *J*. *Cryptol*., vol. 17, no. 4, pp. 297–319, 2004.

[40] D. Pointcheval and O. Sanders, Short randomizable signatures, in *Proc. Cryptographers Track at the RSA Conf.* (*CT-RSA*), San Francisco, CA, USA, 2016, pp. 111–126.

[41] M. Bellare, D. Micciancio, and B. Warinschi, Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. in *Proc. Advances in Cryptology-Int. Conf. Theory and Applications of Cryptographic Techniques* (*EUROCRYPT*), Warsaw, Poland, 2003, pp. 614–629.

[42] D. Critchlow and N. Zhang, Security enhanced accountable anonymous PKI certificates for mobile e-commerce, *Comput*. *Netw*., vol. 45, no. 4, pp. 483–503, 2004.

[43] D. Boneh and X. Boyen, Short signatures without random oracles and the SDH assumption in bilinear groups, *J*. *Cryptol*., vol. 21, no. 2, pp. 149–177, 2008.

[44] C. I. Fan, W. Z. Sun, and V. S. M. Huang, Provably secure randomized blind signature scheme based on bilinear pairing, *Comput*. *Math*. *Appl*., vol. 60, no. 2, pp. 285–293, 2010.

[45] A. Sonnino, M. Al-Bassam, S. Bano , S. Meiklejohn, and G. Danezis, Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers, in *Proc. Ann. Network and Distributed System Security Symp.* (*NDSS*), San Diego, CA, USA, 2019, pp. 1–15.

Yan Zhu received the MS degree from China Agricultural University, China in 2019. He is currently pursuing the PhD degree in cyber science and technology at the School of Cyber Science and Technology, Beihang University, China. His research interests include applied cryptography and blockchain.

Haibin Zheng received the PhD degree from Beihang University, China in 2020, where she is currently an associate researcher in the Research Center of Cyber Science and Technology, Hangzhou Innovation Institute, and also a project leader at the Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, China. Her research interests include blockchain and public-key cryptography, especially privacy protection and supervision in blockchain.

Bo Qin received the PhD degree in cryptography from Xidian University, China in 2008. She is currently an associate professor in Renmin University of China. Her research interests include pairing-based cryptography, data security and privacy, and blockchain and cryptocurriency.

Wanting Fu received the BS degree in information security from Chongqing University, China in 2020. She is currently pursuing the MS degree in cyber science and technology at the School of Cyber Science and Technology, Beihang University, China. Her research interests include privacy protection and supervision in blockchain.

Zhenwei Guo received the BEng and PhD degrees in automation from Zhejiang University, Hangzhou, China in 2016 and 2021, respectively. He is currently a postdoctoral researcher with the Research Center of Cyber Science and Technology, Hangzhou Innovation Institute, Beihang University, China. His research interests include smart grid, blockchain technology, distributed optimization, and P2P electricity market.

Yujue Wang received the PhD degrees from Wuhan University, Wuhan, China in 2015, and City University of Hong Kong, China, under the joint PhD program. He is currently with the Hangzhou Innovation Institute, Beihang University, China. His research interests include applied cryptography and blockchain.

Qianhong Wu received the PhD degree in cryptography from Xidian University, China in 2004. He is currently a professor in Beihang University, China. His research interests include applied cryptography, information security and privacy, VANET security, cloud computing security, cryptocurrency, and blockchain.

Bingyu Li received the BS degree from Jilin University, China in 2013 and the PhD degree from the University of Chinese Academy of Sciences, China in 2020. He is an associate professor with the School of Cyber Science and Technology, Beihang University, China. His research interests include applied cryptography, blockchain, trust management, and network security.

Xuan Ding received the BS degree from Tsinghua University, China in 2008, and the PhD degree from Tsinghua University, China in 2014. He is currently a research assistant professor at the School of Software and BNRist, Tsinghua University. His research interests include privacy-preserving computing, blockchain, RFID, and wireless sensing.