Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.019589

ARTICLE

Unidirectional Identity-Based Proxy Re-Signature with Key Insulation in EHR Sharing System

Yanan Chen[1,2,3,4], Ting Yao[1,4,*], Haiping Ren[2] and Zehao Gan[1]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China

[2]Basic Course Teaching Department, Jiangxi University of Science and Technology, Nanchang, 330013, China

[3]Institute of Electronic and Information Engineering, University of Electronic Science and Technology of China, Dongguan, 523808, China

[4]Network and Data Security Key Laboratory of Sichuan Province, Unversity of Electronic Science and Technology of China, Chengdu, 610054, China

[*]Corresponding Author: Ting Yao. Email: yaoting.uestc@gmail.com

Received: 30 September 2021 Accepted: 30 November 2021

ABSTRACT

The introduction of the electronic medical record (EHR) sharing system has made a great contribution to the management and sharing of healthcare data. Considering referral treatment for patients, the original signature needs to be converted into a re-signature that can be verified by the new organization. Proxy re-signature (PRS) can be applied to this scenario so that authenticity and nonrepudiation can still be insured for data. Unfortunately, the existing PRS schemes cannot realize forward and backward security. Therefore, this paper proposes the first PRS scheme that can provide key-insulated property, which can guarantee both the forward and backward security of the key. Although the leakage of the private key occurs at a certain moment, the forward and backward key will not be attacked. Thus, the purpose of key insulation is implemented. What's more, it can update different corresponding private keys in infinite time periods without changing the identity information of the user as the public key. Besides, the unforgeability of our scheme is proved based on the extended Computational Diffie-Hellman assumption in the random oracle model. Finally, the experimental simulation demonstrates that our scheme is feasible and in possession of promising properties.

KEYWORDS

Proxy re-signature; key insulation; electronic medical record (EHR); random oracle model

1 Introduction

With the improvement of living standards, the healthcare field has attracted more awareness and is playing an increasingly crucial role [1,2]. At the same time, with the aging of the population, the demand for online healthcare treatment is rising. Therefore, the data that the hospital needs to manage is enormous, resulting in a mass of health data that needs to be stored and

maintained. The introduction of electronic medical records (EHRs) can solve this problem well, that is, storing and using patient medical records including personal information through the integration of the Internet of Things (IoT), deep learning, blockchain, and other technologies [3–6]. Compared with the conventional method of using paper to preserve health records, it is more advantageous and convenient to analyze the condition and manage the data in EHR sharing system. Nevertheless, putting health records on the Internet in electronic form will inevitably cause security problems. For the confidentiality of information in the IoT environment, there are already various effective methods to encrypt and protect them [7–11], but it is not enough to have only encryption means, and the authenticity of the message needs to be determined. Any unauthorized changes in data will affect the diagnosis and timely treatment of the disease, thus the integrity of the data must be guaranteed. Generally, digital signatures are used to ensure whether the data has been tampered with or not [12–15]. However, when a patient needs to be referred for treatment, for example, a physician in the hospital needs to give the patient's EHR to the researcher in the institution for further study. In this case, the signature generated by the hospital demands to be converted into the signature under the institution. Traditionally, the institution is required to verify the legality of the signature first and then recompute its own signature. To make matters worse, there are plenty of data to be processed in the EHR sharing system, which is difficult for the institution with limited resources or inconvenient situations. Therefore, the task of converting signatures can be entrusted to the semi-trusted proxy—an insurance company. The technology of proxy re-signature (PRS) can implement the transform requirement, while the proxy can not completely replace the hospital or institution and create any other signatures belonging to them without receiving the authorization and delegation. Through the interaction with the delegator, the signature belonging to the delegatee can be converted into the delegator by proxy. What's more, the conversion process can be completed without the private key of the delegatee, that is, the delegatee does not need actual interactive participation. Accordingly, the concept of PRS can be applied to transfer electronic medical records for management scenarios in EHRs.

However, in a complex environment, the problem of key exposure arises. Once the user is compromised, the attacker can completely pretend to be the user and do whatever he wants, which is definitely fatal. In 2009, Yang et al. [16] combined the two primitives of PRS and forward-secure threshold signature to construct the first forward-secure threshold proxy re-signature scheme. The re-signature key will be updated in different time periods, thus if the re-signature key is leaked in a certain period, it will not alter the previous re-signature key. However, the user's private key does not change, so Sunitha et al. [17] constructed a multi-use PRS scheme with forward security in the e-banking application. Their private key and re-signing key will be updated after a period of time, but their time slice is limited and needs to be set in advance. Although forward-secure schemes have been proposed by some researchers, forward security can only guarantee the security of the key before the leakage, but not after the leakage. Therefore, a key-insulated method in PRS is proposed, which can guarantee both forward and backward security. In addition, while updating the user key, it can also refresh the re-signature key of the proxy, which ensures the security of both user's signature and re-signature. In general, in order to construct a promising PRS scheme to ensure the unforgeability and non-repudiation in the environment of the EHR, the following properties are generally desired.

1) Unidirectional: When converting the signature, the proxy can only convert in the specified direction. Without authorization, the proxy cannot obtain the reverse conversion key through calculation.

2) Single-use: The Resign algorithm can determine whether the input signature is original or converted. For schemes with single-use property, re-signature cannot be used as the input of this algorithm.
3) Transparent: Nobody could judge the existence of the proxy, and the form of the re-signature generated by the proxy is indistinguishable from the signature generated by the user.
4) Non-interactive: In the whole delegating process, there is no need for the delegatee to actually participate, so as to realize the non-interactivity between the delegatee and the delegator.
5) ID-based: The user directly views the public parameter of identity information (ID) as his public key, so there is no need for certificate authority (CA) to specifically produce a certificate to bind the public key with the user.
6) Forward-Secure: The keys generated before the leakage occurred are not correlated with the leaked key, and their security will not be affected.
7) Backward-Secure: Conversely, if after key exposure, the key generated later is still secure, it is called backward security.

1.1 Related Work

The primitive proxy re-signature was first proposed by Blaze et al. [18], but it did not give a specific formal definition so that it did not attract people's attention for a period of time. Until 2005, Ateniese et al. gave a formal definition and security model for PRS [19]. The delegator can authorize the proxy to generate a re-signature key, and then the proxy can utilize this key to convert the designated signature from the delegatee to delegator. Shao et al. [20] eliminated the random model, constructed a PRS scheme under the standard model. Moreover, the PRS scheme in the identity-based cryptosystem was proposed for the first time. Then, Libert et al. [21] solved the open problem left by [19] and proposed the first multi-use unidirectional PRS scheme. The signature can be re-signed by multiple users in sequence, and the direction of conversion is nonreversing. Due to the semi-trusted nature of the proxy, Yang et al. [22] formally proposed the threshold proxy re-signature scheme. Multiple proxies are utilized to jointly perform the re-signature process, only when the number of proxies reaches the threshold. In 2011, Shao et al. [23] first combined the unidirectional PRS scheme with the identity-based cryptosystem. In the random oracle, the unforgeability of the signature can be proved based on the extended Computational Diffie-Hellman assumption. In order to save the Computational cost of the verification algorithm, Wang et al. [24] proposed a PRS scheme with server-assisted verification. In this scheme, the proxy can be used to verify the validity of the signature besides resigning, so that the user does not need to undertake the heavy burden of calculation. Then, Patonico et al. [25] put forward an efficient proxy re-signcryption scheme using the arithmetic operation in the elliptic curve, so as to realize the safe ownership transfer in the cloud. Under the attention of quantum computers, in order to resist this kind of attack, some researchers [26,27] turned their direction to construct PRS schemes from the perspective of lattices. Recently, more and more scenarios need to take advantage of the demand for re-signing. The PRS scheme is used in authentication [28–32], auditing [33–37], secure automated valet parking [38] and data sharing [39].

1.2 Contribution

In this paper, we propose the first unidirectional identity-based proxy re-signature scheme with key insulation (KI-IDPRS), which can satisfy all the properties mentioned above. First, an ID-based scheme reduces the overhead of managing public key certificates by taking the user ID as

the public key. In addition, considering identity information is fixed for the user, KI-IDPRS can update the private key without changing the public key to achieve the key-insulated property. The proxy can be authorized by the delegator to convert the signature from delegatee into delegator, but not vice versa. What's more, the secure status of the key at current moment will not affect the keys at other moments, thus realizing key insulation. Although the proxy is semi-trusted, even if both the user and the proxy are compromised at one certain moment, the security is not broken at other times. Security of this scheme can be reduced to extended Computational Diffie-Hellman (eCDH) assumption in the random oracle model. Finally, it can be concluded that our scheme is feasible and has nice properties from the theoretical analysis including experimental results.

1.3 Organization

The following parts of the paper will discuss our proposed scheme as follows: the second part introduces related basic knowledge including system model, the formal definition and security model of the proposed scheme. The third part constructs the first unidirectional identity-based PRS scheme with key insulation and proves the unforgeable security of the proposed scheme. The fourth part puts together other relevant schemes with our scheme from the three dimensions of properties, computing cost and communication overhead, then compares them through experiments.

2 Preliminaries

2.1 Bilinear Map

Given two cyclic groups $\mathbb{G}$, $\mathbb{G}_T$ of prime order p and g be generator of $\mathbb{G}$. A map that meets the following requirements can form a bilinear pairing $e: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

1) Bilinearity: $\forall x, y \in Z_p^*, e(g^x, g^y) = e(g, g)^{xy}$.
2) Non-degeneracy: $e(g, g) \neq 1$.
3) Computability: $e(g^x, g^y)$ can be computed.

2.2 eCDH Assumption

In the case of known a tuple $< g, g^a, g^b > \in \mathbb{G}$, compute the value of a pair (A, A^{ab}), for $a, b \in \mathbb{Z}_p^*$, $A \in \mathbb{G}$.

2.3 System Model

The proposed scheme requires six entities to complete together: Private key generator (PKG), helper A, helper B, user A in hospital (delegatee), user B in the institution (delegator), and an insurance company (proxy). The system model is depicted in Fig. 1, and the description of the notations involved in the model is introduced in Table 1.

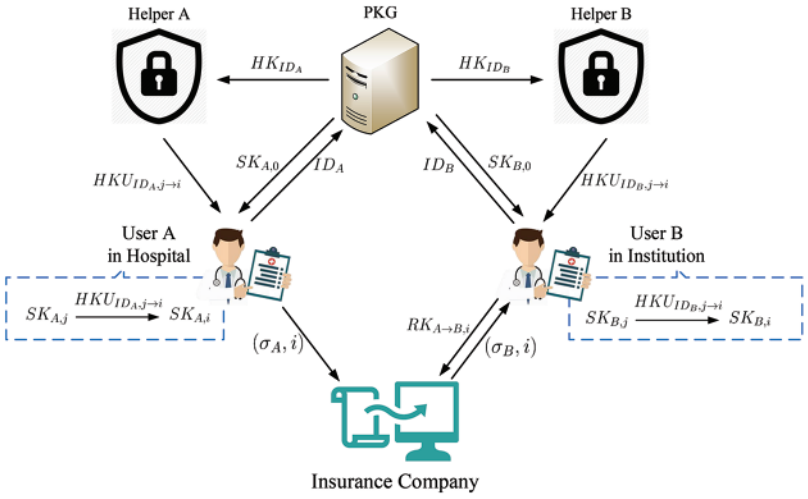

Figure 1: System model

Table 1: The list of symbolic representations

Symbol	Description
ID_A, ID_B	Identity information of the user A and B
$SK_{A,0}, SK_{B,0}$	Initial private key of the user A and B
$SK_{A,j}, SK_{B,j}$	Temporary private key of the user A and B in the time period j
$SK_{A,i}, SK_{B,i}$	Temporary private key of the user A and B in the time period i
HK_{ID_A}, HK_{ID_B}	Helper key with ID_A and ID_B
$HKU_{ID_A,j\to i}$, $HKU_{ID_B,j\to i}$	Update key from the time period j to i with ID_A and ID_B
$RK_{A\to B,i}$	Re-signing key in the time period i
$(\boldsymbol{\sigma_A}, i), (\boldsymbol{\sigma_B}, i)$	Signature of the user A and B in the time period i

PKG: PKG is a trusted authority used to generate secret keys. When a user is supposed to extract its secret key, the identity information will be given to the PKG. The corresponding initial key and helper key will be generated by PKG, and they will be returned to the user and helper, respectively.

Helper A/B: The helper is an auxiliary device with absolute physical security but limited computing power, which is used to store the helper key. It only needs to interact with the user at the beginning of each time period, generate the update key and send it to the user to help the user update the temporary private key, but does not participate in any other cryptographic operations.

User A in hospital: The hospital can act as the delegatee to receive the delegation from the institution, and he sends the signature that needs to be re-signed to the insurance company.

User B in institution: The institution is viewed as the delegator to re-sign user A's signature into his own signature through the insurance company, rather than directly signing the message.

Insurance company: The insurance company is used to implement the re-signature process through the re-signature key. After receiving the signature of user A and the re-signing key entrusted by user B, he converts the specified signature from A to B.

2.4 Syntax of the KI-IDPRS

A KI-IDPRS scheme consists of eight different algorithms, which are as follows:

1) *Setup:* This is a system establishment algorithm executed by key generation center (KGC). Security parameter 1^k is given to KGC and the master public key mpk is returned back for public use, the master secret key msk is returned for PKG.
2) *Extract:* This is an initial private key extraction algorithm executed by PKG. The master secret key msk and a user's identity ID are the input of PKG. Then, PKG outputs the initial private key $SK_{ID,0}$ stored by the user and the helper key HK_{ID} stored by the helper for the identity ID.
3) *HUpdate:* This is an update key generation algorithm executed by the helper. The helper key HK_{ID} for the identity ID and the time period j, i are as the input of the helper. Then, the helper outputs the update key $HKU_{ID,j\to i}$ used to update time period j to i.
4) *UUpdate:* This is a temporary private key update algorithm executed by the user. The private key $SK_{ID,j}$ in the old time period j for the identity ID and the update key $HKU_{ID,j\to i}$ are as the input of the user. Then, the user outputs the private key $SK_{ID,i}$ in the new time period i for the identity ID.
5) *RKGen:* This is a re-signature key generation algorithm executed by the delegator. The delegatee's identity ID_A and the delegator's private key $SK_{ID_B,i}$ in the time period i are as the input of the delegator. Then, the delegator outputs the re-signature key $RK_{A\to B,i}$ to the proxy, which is used to implement signature conversion from identity ID_A to ID_B under the same message m in the time period i.
6) *Sign:* This is a signature generation algorithm executed by the user/delegatee. The private key $SK_{ID_A,i}$ in the time period i for the identity ID_A, message m and the signature's level L are as the input of the delegatee. Then, the delegatee outputs signature $(\boldsymbol{\sigma}_A^{[L]}, i)$ at level L in the time period i, where $L = 1, 2$.
7) *ReSign:* This is a re-signature conversion algorithm executed by the proxy. The original signature $(\boldsymbol{\sigma}_A^{[1]}, i)$ at level 1, message m, the delegatee's identity ID_A and the re-signature key $RK_{A\to B,i}$ are as the input of the proxy. Then, the proxy outputs the signature $(\boldsymbol{\sigma}_B^{[2]}, i)$ at level 2 for identity ID_B in the time period i, if $(\boldsymbol{\sigma}_A^{[1]}, i)$ can pass the *Verify* algorithm.
8) *Verify:* This is a verification algorithm. The alleged signature $(\boldsymbol{\sigma}_{ID}^{(L)}, i)$ at level L, message m and the identity ID are as the input. Then, 1 can be returned if the signature is valid.

2.5 Security Model of the KI-IDPRS

To assess the security of the key-insulated proxy re-signature in a formal manner, the following interactive game between the challenger $\mathscr{C}$ and the adversary $\mathscr{A}$ is defined by incorporating the security model for the key insulated signature [40] and the one for the proxy re-signature [23].

Setup: $\mathscr{C}$ performs the algorithm Setup to generate the master public key mpk and the master secret key msk, and then returns mpk to $\mathscr{A}$ while keeps msk secret.

Query: Before the adversary $\mathscr{A}$ attempts to forge a signature, the challenger $\mathscr{C}$ permits him to adaptively make a number of different queries.

$O(Extract)$: $\mathscr{C}$ performs the algorithm Extract to produce the user $ID's$ initial secret key $SK_{ID,0}$ for $\mathscr{A}$'s request (ID).

$O(UUpdate)$: $\mathscr{C}$ performs the algorithm UUpdate to produce the user $ID's$ temporary secret key $SK_{ID,i}$ in time period i for $\mathscr{A}$'s request (ID, i).

$O(RKGen)$: $\mathscr{C}$ performs the algorithm RKGen to produce the re-signature key $RK_{A\to B,i}$ in time period i for $\mathscr{A}$'s request (ID_A, ID_B, i).

$O(Sign)$: $\mathscr{C}$ performs the algorithm Sign to produce the signature $(\boldsymbol{\sigma}, L, i)$ in time period i for $\mathscr{A}$'s request (ID, L, i, m).

$O(ReSign)$: $\mathscr{C}$ performs the algorithm ReSign to produce the re-signature $(\boldsymbol{\sigma}', L+1, i)$ in time period i for $\mathscr{A}$'s request $(\boldsymbol{\sigma}, L, i, m, ID_A, ID_B)$.

Forgery Output: Suppose $(ID^*, m^*, \boldsymbol{\sigma}^*, L^*, i^*)$ is $\mathscr{A}$'s counterfeit result after queries. If the adversary meets the following restrictions at the same time, and can still generate a valid signature that makes the equation $\text{Verify}(\boldsymbol{\sigma}, i, m, ID) = 1$ hold, then the adversary is considered to be the winner in this game.

1) $\mathscr{A}$ can't query ID^* in $O(Extract)$
2) $\mathscr{A}$ can't query (ID^*, i^*) in $O(UUpdate)$
3) $\mathscr{A}$ can't query ID^* in $O(RKGen)$
4) $\mathscr{A}$ can't query (ID^*, m^*, i^*, L^*) in $O(Sign)$
5) $\mathscr{A}$ can't query $(\boldsymbol{\sigma}^*_{ID_j,i^*}, m^*, ID_j, ID^*)$ in $O(ReSign)$

3 Our Proposed KI-IDPRS Scheme

3.1 Construction

By combining identity-based key-insulated signature [41] in the unidirectional IDPRS [23], the concrete construction of the unidirectional identity-based PRS scheme with insulated key has been given as follows:

• $\text{Setup}(1^k) \to (mpk, msk)$: After giving the security parameter 1^k, the following operations are performed by KGC to generate the public key mpk and the master secret key msk.

1) Choose two finite cyclic groups $\mathbb{G}$, $\mathbb{G}_T$ with prime order $p = \Theta\left(2^k\right)$, and choose a generator g of $\mathbb{G}$.
2) Pick bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$;
3) Generate the master key: $x \xleftarrow{R} \mathbb{Z}_p^*$ and compute $X = g^x$;
4) Define three hash functions: $H_1(\cdot), H_2(\cdot) : \{0,1\}^* \to Z_p^*$, $H_3(\cdot) : \{0,1\}^* \to \mathbb{G}$;
5) Generate the public key $(mpk = (g, p, \mathbb{G}, \mathbb{G}_T, e, X, H_1, H_2), msk = x)$.

• $\text{Extract}(ID, msk) \to (HK_{ID}, SK_{ID,0})$: After inputting the master secret key msk and a user's identity ID, PKG firstiy randomly chooses one element $HK_{ID} \xleftarrow{R} \mathbb{Z}_p^*$ as the helper key. Then, it computes the initial secret key $SK^{(1)}_{ID,0} = g^{HK_{ID}}$, $SK^{(2)}_{ID,0} = H_1(SK^{(1)}_{ID,0} \| ID) \cdot x + HK_{ID} \cdot H_2(ID \| 0) \bmod p$, and sets $SK_{ID,0} = (SK^{(1)}_{ID,0}, SK^{(2)}_{ID,0})$. In fact, for a user, his $SK^{(1)}_{ID}$ can be obtained by other users, only $SK^{(2)}_{ID}$ is kept private. Because it is obvious to find that the user's signature has $SK^{(1)}_{ID}$ as one of the components, and $SK^{(1)}_{ID}$ is not updated in our proposed scheme.

• HUpdate$(HK_{ID}, ID, i, j) \rightarrow HKU_{ID,j\rightarrow i}$: On receiving the helper key HK_{ID} for ID and the time period j, i, the helper outputs the update key $HKU_{ID,j\rightarrow i}$ by computing $HKU_{ID,j\rightarrow i} = HK_{ID} \cdot (H_2(ID\|i) - H_2(ID\|j))$.

• UUpdate$(SK_{ID,j}, HKU_{ID,j\rightarrow i}, ID, i, j) \rightarrow SK_{ID,i}$: After receiving the private key $SK_{ID,j}$ in the old time period j and the update key $HKU_{ID,j\rightarrow i}$, the user updates the private key $SK_{ID,i}$ as follows:

1) Keep the value of $SK_{ID,0}^{(1)}$ so that $SK_{ID,i}^{(1)} = SK_{ID,j}^{(1)}$.

2) *Compute* $SK_{ID,i}^{(2)} = SK_{ID,j}^{(2)} + HKU_{ID,j\rightarrow i}$

$$= H_1(SK_{ID,j}^{(1)}\|ID) \cdot x + HK_{ID} \cdot H_2(ID\|i) + HK_{ID} \cdot (H_2(ID\|j) - H_2(ID\|j))$$
$$= H_1(SK_{ID,i}^{(1)}\|ID) \cdot x + HK_{ID} \cdot H_2(ID\|i).$$

• RKGen$(ID_A, SK_{ID_B,i}) \rightarrow RK_{A\rightarrow B,i}$: After the delegator receives the identity ID_A of the delegatee, the delegator calculates $RK_{A\rightarrow B,i}$ through its private key $SK_{ID_B,i}$ as follows and then gives it to the proxy.

1) $RK_{A\rightarrow B,i}^{(1)} = \left(X^{H_1(SK_{ID_A,i}^{(1)}\|ID_A)} \cdot (SK_{ID_A,i}^{(1)})^{H_2(ID_A\|i)} \right)^{1/SK_{ID_B,i}^{(2)}}$.

2) $RK_{A\rightarrow B,i}^{(2)} = SK_{ID_B,i}^{(1)}$.

• Sign$(ID, m, SK_{ID,i}) \rightarrow \begin{cases} \boldsymbol{\sigma}^{[1]} = (\boldsymbol{\sigma}^{(1)}, \boldsymbol{\sigma}^{(2)}) \\ \sigma^{[2]} = (\sigma^{(1)}, \sigma^{(2)}, \sigma^{(3)}, \sigma^{(4)}) \end{cases}$: After receiving the message, the delegatee signs the message m with ID_A and the private key $SK_{ID_A,i}$. First-level signature $\sigma_A^{[1]}$ and second-level signature $\sigma_A^{[2]}$ are computed as follows, and then the first-level signature $\sigma_A^{[1]}$ is given to the proxy to generate the re-signature of the delegator.

1) For $\sigma_A^{[1]}$, compute $\sigma_A^{(1)} = H_3(m)^{SK_{ID_A,i}^{(2)}}$, $\sigma_A^{(2)} = SK_{ID_A,i}^{(1)}$.

2) For $\sigma_A^{[2]}$, randomly choose $t \xleftarrow{R} \mathbb{Z}_p^*$ and compute $\sigma_A^{(1)} = H_3(m)^{SK_{ID_A,i}^{(2)} \cdot t}$,

$$\sigma_A^{(2)} = \left(X^{H_1(SK_{ID_A,i}^{(1)}||ID_A)} \cdot (SK_{ID_A,i}^{(1)})^{H_2(ID_A\|i)} \right)^t, \ \sigma_A^{(3)} = g^t, \ \sigma_A^{(4)} = SK_{ID_A,i}^{(1)}.$$

• ReSign$(\sigma_A^{[1]}, RK_{A\rightarrow B,i}) \rightarrow \sigma_B^{[2]}$: After receiving the delegatee's first-level signature $\sigma_A^{[1]}$ and the re-signature key $RK_{A\rightarrow B,i}$, the proxy first ensures the legality of the signature $\sigma_A^{[1]}$ by examining whether the equation $e(\sigma_A^{(1)}, g) = e\left(H_3(m), X^{H_1\left(\sigma_A^{(2)}||ID_A\right)} \cdot (\sigma_A^{(2)})^{H_2(ID_A\|i)} \right)$ can be satisfied. If not, an error is output. Otherwise, the proxy calculates the second-level signature $\sigma_B^{[2]}$ of the delegator as follows:

1) Randomly choose $t \overset{R}{\leftarrow} \mathbb{Z}_p^*$.

2) *Compute* $\sigma_B^{[2]} = \left(\sigma_B^{(1)'}, \sigma_B^{(2)'}, \sigma_B^{(3)'}, \sigma_B^{(4)'}\right)$

$$= \left((\sigma_A^{(1)})^t, \quad (X^{H_1(SK_{ID_A,i}^{(1)}||ID_A)} \cdot (\sigma_A^{(2)})^{H_2(ID_A||i)})^t, \quad (RK_{A\to B,i}^{(1)})^t, \quad RK_{A\to B,i}^{(2)}\right)$$

$$= H_3(m)^{SK_{ID_A,i}^{(2)} \cdot t' \cdot \frac{SK_{ID_B,i}^{(2)}}{SK_{ID_A,i}^{(2)}}}, \quad (g^{SK_{ID_A,i}^{(2)}})^{t' \cdot \frac{SK_{ID_B,i}^{(2)}}{SK_{ID_A,i}^{(2)}}}, \quad (g^{SK_{ID_A,i}^{(2)} \cdot \frac{1}{SK_{ID_B,i}^{(2)}}})^{t' \cdot \frac{SK_{ID_B,i}^{(2)}}{SK_{ID_A,i}^{(2)}}}, \quad SK_{ID_B,i}^{(1)}$$

$$= H_3(m)^{SK_{ID_B,i}^{(2)} \cdot t'}, \quad \left(X^{H_1(SK_{ID_B,i}^{(1)}||ID_B)} \cdot (SK_{ID_B,i}^{(1)})^{H_2(ID_B\|i)}\right)^{t'}, \quad g^{t'}, \quad SK_{ID_B,i}^{(1)}$$

where, $\begin{cases} t = t' \cdot \frac{SK_{ID_B,i}^{(2)}}{SK_{ID_A,i}^{(2)}} \\ g^{SK_{ID,i}^{(2)}} = X^{H_1(SK_{ID,i}^{(1)}||ID)} \cdot (SK_{ID,i}^{(1)})^{H_2(ID\|i)} \end{cases}$

• Verify$(\sigma, i, m, ID) \to 1/\perp$: After entering the signature $\sigma^{[1]}/\sigma^{[2]}$, message m, and identity ID, the algorithm checks the validity of the first-level signature through equation $e\left(H_3(m), X^{H_1(\sigma^{(2)}||ID)} \cdot (\sigma^{(2)})^{H_2(ID\|i)}\right)$
$= e(\sigma^{(1)}, g)$, and the validity of the second-level signature through equation $e\left(\sigma^{(1)}, g\right) = e\left(H_3(m), \sigma^{(2)}\right)$,
$e\left(\sigma^{(2)}, g\right) = e\left(X^{H_1(\sigma^{(4)}||ID)} \cdot (\sigma^{(4)})^{H_2(ID||i)}, \sigma^{(3)}\right)$.

3.2 Security Analysis

In the random oracle model, our proposal is existentially unforgeable under the eCDH assumption in $\mathbb{G}$.

Proof: Assume that there is an adversary $\mathscr{A}$ can break the existential unforgeability of our proposal with non-negligible probability ε, then we can build another algorithm $\mathscr{B}$ to solve the eCDH problem. The input of eCDH problem is (g, g^a, g^b), and the goal of security proof is to get (A, A^{ab}), where A could be any element in $\mathbb{G}$.

Setup: The challenger $\mathscr{C}$ initializes the system according to the following steps:

1) Set $X = g^a$;
2) Send $mpk = (g, p, \mathbb{G}, \mathbb{G}_T, e, X, H_1, H_2, H_3)$ to $\mathscr{A}$, where the values of H_1, H_2 and H_3 can be obtained through the following $O(H_1)$, $O(H_2)$ and $O(H_3)$ queries, respectively.

Query:

• $O(H_1) \to (R||ID, h_1) \in H_1^{list}$

1) $\mathscr{A}$ inputs $(R||ID)$ to query the value of $H_1(R||ID)$;
2) If there is the item $(R||ID)$ in the H_1^{list}, $\mathscr{C}$ returns the corresponding h_1 to $\mathscr{A}$;
3) If not, $\mathscr{C}$ randomly chooses one element h_1 from $\mathbb{Z}_p^*$ as output and adds it to the H_1^{list}.

- $O(H_2) \rightarrow (ID||i, h_2) \in H_2^{list}$

1) $\mathscr{A}$ inputs $(ID||i)$ to query the value of $H_2(ID||i)$;
2) If there is the item $(ID||i)$ in the H_2^{list}, $\mathscr{C}$ returns the corresponding h_2 to $\mathscr{A}$;
3) If not, $\mathscr{C}$ randomly chooses one element h_2 from $\mathbb{Z}_p^*$ as output and adds it to the H_2^{list}.

- $O(H_3) \rightarrow (m, \alpha, (g^b)^\alpha) \in H_3^{list}$

1) $\mathscr{A}$ inputs m to query the value of $H_3(m)$;
2) If m can be found in H_3^{list}, $\mathscr{C}$ returns the corresponding $h_3 = (g^b)^\alpha$ to $\mathscr{A}$;
3) If not, $\mathscr{A}$ randomly chooses one element α from $\mathbb{Z}_p^*$ and computes $(g^b)^\alpha$ as output, then adds it to the H_3^{list}.

- $O(Extract) \rightarrow (ID, SK_{ID,0})$

1) $\mathscr{A}$ inputs ID to query the corresponding initial secret key $SK_{ID,0}$;
2) $\mathscr{C}$ randomly chooses $h_1, h_2, SK_{ID,0}^{(2)} \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and computes $SK_{ID,0}^{(1)} = (g^{SK_{ID,0}^{(2)}} / X^{h_1})^{\frac{1}{h_2}}$;
3) If $(SK_{ID,0}^{(1)}||ID, h_1) \in H_1^{list}$ and $(ID||0, h_2) \in H_2^{list}$, $\mathscr{C}$ outputs "failure" and aborts;
4) If not, $\mathscr{C}$ adds $(SK_{ID}^{(1)}||ID, h_1)$, $(ID||0, h_2)$ to the H_1^{list}, H_2^{list} and returns $(SK_{ID,0}^{(1)}, SK_{ID,0}^{(2)})$ as the initial secret key to $\mathscr{A}$.

- $O(UUpdate) \rightarrow (ID, i, SK_{ID,i})$

1) $\mathscr{A}$ inputs ID and the time period i to query the temporary secret key $SK_{ID,0}$ in the time period i;
2) $\mathscr{C}$ randomly chooses $h_1, h_2, SK_{ID,i}^{(2)} \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and computes $SK_{ID,i}^{(1)} = (g^{SK_{ID,i}^{(2)}} / X^{h_1})^{\frac{1}{h_2}}$;
3) If $(SK_{ID,i}^{(1)}||ID, h_1) \in H_1^{list}$ and $(ID||i, h_2) \in H_2^{list}$, $\mathscr{C}$ outputs "failure" and aborts;
4) If not, $\mathscr{C}$ adds $(SK_{ID}^{(1)}||ID, h_1)$, $(ID||i, h_2)$ to the H_1^{list}, H_2^{list} and returns $(SK_{ID,i}^{(1)}, SK_{ID,i}^{(2)})$ as the temporary secret key for time period i to $\mathscr{A}$.

- $O(RKGen), O(Sign), O(ReSign)$

1) $\mathscr{A}$ obtains the corresponding private key via $O(Extract)$ and $O(UUpdate)$;
2) $\mathscr{C}$ computes the required query value via the corresponding private key and returns it to $\mathscr{A}$;

Forgery Output:

According to the forking lemma [42], for level 1, $\mathscr{A}$ can counterfeit two valid signatures $(SK_{ID^*,i^*}^{(1)}, h_1, SK_{ID^*,i^*}^{(2)}, \sigma^{(1)}, \sigma^{(2)}, m^*)$, $(SK_{ID^*,i^*}^{(1)}, h_1', SK_{ID^*,i^*}^{(2)}, \sigma'^{(1)}, \sigma'^{(2)}, m^*)$. h_1 and h_1' are two different random responses from $O(H_1)$ on input $(SK_{ID^*,i^*}^{(1)}||ID^*)$, then $\mathscr{B}$ can compute:

$$\left(\frac{\sigma^{(1)}}{\sigma'^{(1)}}\right)^{\frac{1}{h_1 - h_1'}} = \left(\frac{\left(\left(g^b\right)^\alpha\right)^{h_1 \cdot a + h_2 \cdot HK_{ID*}}}{\left((g^b)^\alpha\right)^{h_1' \cdot a + h_2 \cdot HK_{ID*}}}\right)^{\frac{1}{h_1 - h_1'}} = (g^\alpha)^{ab}.$$

Similarly, for level 2, $\mathscr{A}$ can counterfeit two valid signatures $(SK^{(1)}_{ID^*,i^*}, h_1, SK^{(2)}_{ID^*,i^*}, \sigma^{(1)}, \sigma^{(2)}, \sigma^{(3)}, \sigma^{(4)})$, $(SK^{(1)}_{ID^*,i^*}, h'_1, SK^{(2)}_{ID^*,i^*}, \sigma'^{(1)}, \sigma'^{(2)}, \sigma^{(3)}, \sigma^{(4)})$. Then, $\mathscr{B}$ can compute:

$$\left(\frac{\sigma^{(1)}}{\sigma'^{(1)}}\right)^{\frac{1}{h_1-h'_1}} = \left(\frac{\left(\left(g^b\right)^{\alpha}\right)^{(h_1\cdot a+h_2\cdot HK_{ID*})\cdot t}}{\left(\left(g^b\right)^{\alpha}\right)^{(h'_1\cdot a+h_2\cdot HK_{ID*})\cdot t}}\right)^{\frac{1}{h_1-h'_1}} = \left(g^{\alpha\cdot t}\right)^{ab}.$$

4 Comparison

Comparison between our KI-IDPRS scheme and state-of-the-art [21,30,43] will be comprehensively discussed from the perspective of properties, computation cost, and communication overhead. The schemes involved in the comparison are the first PRS scheme featured with the multi-use and unidirectional translation [21], the PRS scheme with key-leakage resistance [43] and the up-to-dated PRS scheme [30]. Then, the simulated implementation of these schemes is conducted through the experimental platform.

4.1 Simulated Implementation

To make the following theoretical analysis more convincing, these schemes are simulated on a specific experimental platform. The computer's operating system is 64-bits Windows 10, the processor is Intel Core i7-7700 @ 3.60 GHz, and the memory is 8GB. Based on VC++6.0, cryptography operations are implemented with the Pairing-Based Cryptography (PBC) library [44], where parameter is the standard parameter *a.param* and $|G|$=128 bytes, $\left|Z_p^*\right|$=20 bytes. In addition, the unit time of critical operations is separately measured and listed in Table 2.

Table 2: The time costs of cryptographic operations

Notion	Operation	Time (ms)
T_P	Bilinear pairing	11.982463
T_E	Exponentiation operation in $\mathbb{G}$	5.993156
T_A	point addition in $\mathbb{G}$	0.001278
T_{SM}	Scalar multipliaction in $\mathbb{G}$	0.000312
T_H	Hash funtion	0.000268
T_{MM}	Modular multipliaction in $\mathbb{Z}_p^*$	0.000049

4.2 Properties

We list the properties of the proposed scheme and the relevant work [21,30,43] in Table 3, where "√" means that the property is supported, "×" means that the feature is not supported. According to the comparison results, it is obvious that all the desired properties can be satisfied in our protocol. Although the scheme [21] owns several desired properties of PRS, this scheme is built in the public key infrastructure and is difficult to be deployed in practical in view of the burden brought by the public-key certificates. In scheme [43], the identity-based cryptographic system is applied to simplify certificate management. Besides, the schemes [30,43] have forward security to prevent the key leakage from affecting the previous key. However, they ignore the security of the private key after key exposure. Only our scheme could provide the key insulated

property and thus achieve forward and backward security simultaneously. Although schemes [43] and [30] realize forward security, they divide the time into a presetting limited time period for updating private keys. In contrast to the works in [30,43], the proposed scheme provides unlimited periods.

Table 3: The properties comparison in different schemes

Scheme	[21]	[43]	[30]	Ours
Unidirectional	√	×	√	√
Single-use	√	×	√	√
Transparent	√	√	×	√
Non-interactive	√	×	√	√
ID-based	×	√	√	√
Forward-Secure	×	√	√	√
Backward-Secure	×	×	×	√

4.3 Communication Overhead

The communication overhead of these schemes is discussed in Table 4, Figs. 2 and 3, where $|G|$ and $\left|Z_p^*\right|$ represent the length of an element in group $\mathbb{G}$ and group $\mathbb{Z}_p^*$, respectively. In the experiment, we take $|G|$=128 bytes and $\left|Z_p^*\right|$=20 bytes. Scheme [21], [30] and our signature length are both $2|G|$, and scheme [43] is the longest. In addition, the length of the re-signature is the same. In terms of the re-signature key, our size is one more $|G|$ than scheme [21] and one $|G|$ less than schemes [43] and [30]. In general, the length of signature and re-signature in KI-IDPRS is less than or equal to other schemes. Therefore, the proposed scheme does not occupy more communication overhead compared with other schemes.

Table 4: The comparison of computation costs

Scheme	Signature	Re-signature	Re-signature key	Private key
[21]	$2\|G\|$	$4\|G\|$	$\|G\|$	$\left\|Z_p^*\right\|$
[43]	$4\|G\|$	$4\|G\|$	$3\|G\|$	$3\|G\|$
[40]	$2\|G\|$	$4\|G\|$	$3\|G\|$	$2\left\|Z_p^*\right\|$
Ours	$2\|G\|$	$4\|G\|$	$2\|G\|$	$\left\|Z_p^*\right\|+\|G\|$

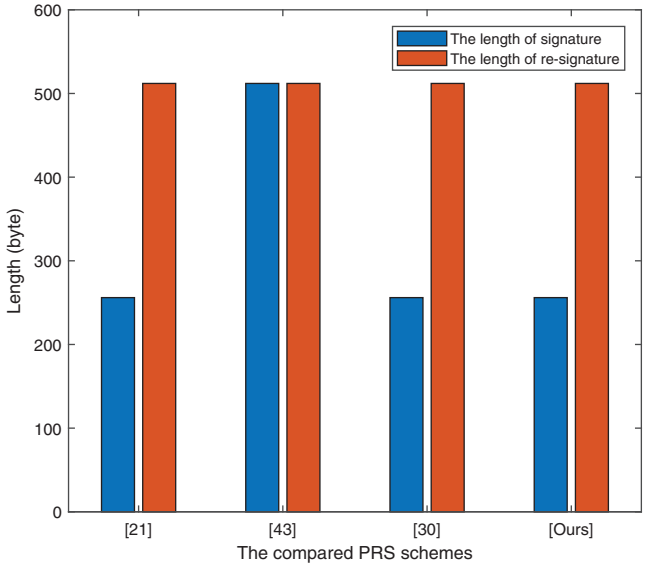

Figure 2: Comparison of signature/re-signature size

4.4 Computation Cost

From the data in Table 2, it is clear that the hash function, scalar multiplication, and modular multiplication take relatively little time. Therefore, exponential and pairing operations are mainly considered in the theoretical analysis of these schemes. T_E is used to represent the time required to perform an exponential operation in group $\mathbb{G}$, and T_P represents the runtime of a bilinear pairing. In the signature phase, results in Table 5, Figs. 4–6 can illustrate that our scheme is less than or equal to other schemes in terms of computational overhead. In the process of re-signing, the scheme [21] requires six exponential operations, which is more than our scheme. However, the schemes [43] and [30] only require two and three exponential operations, respectively. Our KI-PRS scheme requires five, which is acceptable, because our scheme can support both forward and backward security. When checking the validity of the first-level signature, it only takes more time than the scheme [30], where $T_P \approx 2T_E$. When performing the verification of second-level signature, our scheme requires four exponential operations and two pairing operations (that is, approximately equal to five T_P), which is no more than other schemes. In short, compared with other schemes, KI-IDPRS may not cost more time than some schemes, but this is acceptable for the acquisition of functional and safety enhancements.

Table 5: The comparison of computation costs

Scheme	Sign-level 1	Sign-leve 2	Resign	Verify-level 1	Verify-level 2
[21]	$3T_E$	$5T_E$	$6T_E$	$3T_P$	$5T_P$
[43]	$2T_E$	-	$2T_E$	-	$5T_P$
[30]	T_E	-	$3T_E$	$2T_P$+T_E	$4T_P$+$2T_E$
Ours	T_E	$5T_E$	$5T_E$	$2T_P+2T_E$	$4T_P+2T_E$

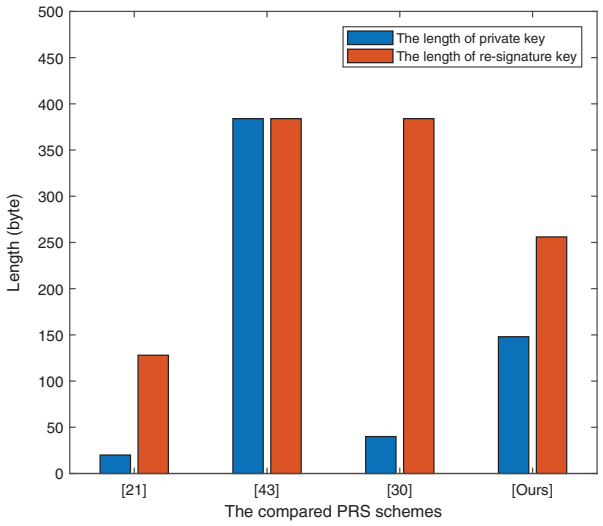

Figure 3: Comparison of re-signature key/private key size

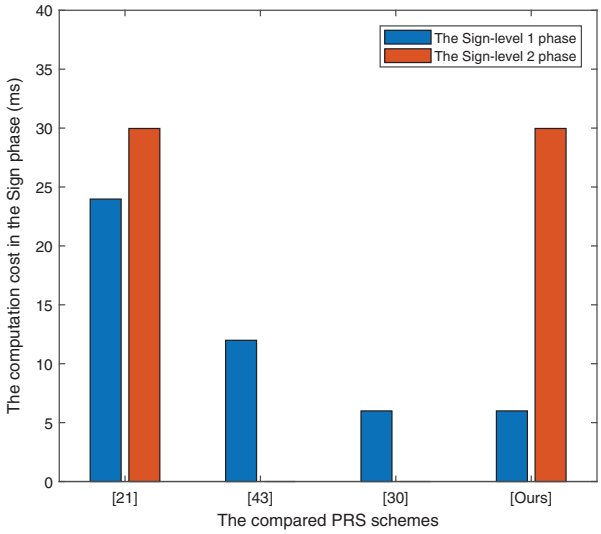

Figure 4: Comparison of sign phase time

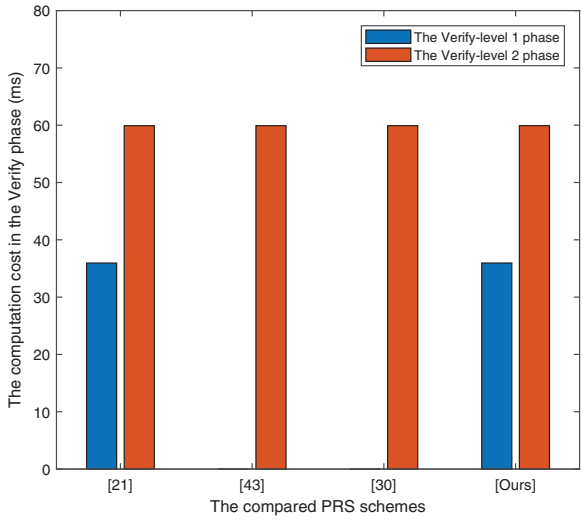

Figure 5: Comparison of verify phase time

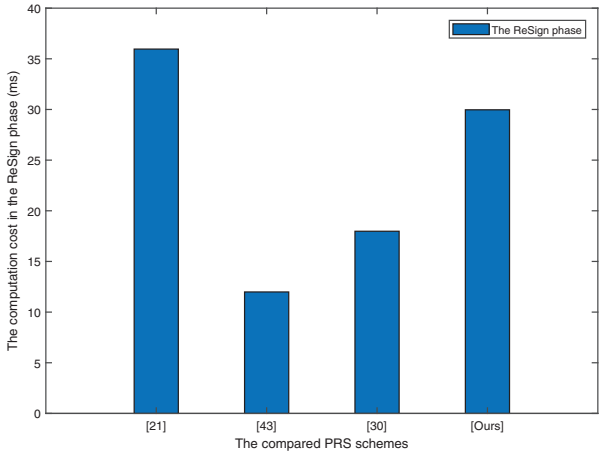

Figure 6: Comparison of ReSign phase time

5 Conclusion

This paper proposes the first KI-IDPRS scheme, which is conducive to timely dealing with the occurrence of key leakage in the EHR sharing system. Then, the formal definition and the security model of KI-IDPRS are given. On this basis, a concrete KI-IDPRS scheme is constructed and proved to have unforgeable security under the eCDH assumption in the random oracle model. What's more, the presented KI-IDPRS scheme can support both forward and backward security, updating private key within unlimited periods. Accordingly, key leakage will not cause a catastrophic threat to the EHR sharing system. Finally, from theoretical analysis and experiment

evaluation, related schemes are compared from three dimensions in properties, communication and computation costs. The presented KI-IDPRS is the only scheme with all the promising properties.

Funding Statement: This work is partially supported by the Network and Data Security Key Laboratory of Sichuan Province under the Grant No. NDS2021-2, in part by Science and Technology Project of Educational Commission of Jiangxi Province under the Grant No. GJJ190464, and in part by National Natural Science Foundation of China under the Grant No. 71661012.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Ranchal, R., Bastide, P., Wang, X., Gkoulalas-Divanis, A., Mehra, M. et al. (2020). Disrupting healthcare silos: Addressing data volume, velocity and variety with a cloud-native healthcare data ingestion service. *IEEE Journal of Biomedical and Health Informatics, 24(11),* 3182–3188. DOI 10.1109/JBHI.6221020.
2. Schiza, E. C., Kyprianou, T. C., Petkov, N., Schizas, C. N. (2019). Proposal for an ehealth based ecosystem serving national healthcare. *IEEE Journal of Biomedical and Health Informatics, 23(3),* 1346–1357. DOI 10.1109/JBHI.6221020.
3. Hamza, R., Yan, Z., Muhammad, K., Bellavista, P., Titouna, F. (2020). A Privacy-preserving cryptosystem for iot e-healthcare. *Information Sciences, 527,* 493–510. DOI 10.1016/j.ins.2019.01.070.
4. Nie, L., Wang, M., Zhang, L., Yan, S., Zhang, B. et al. (2015). Disease inference from health-related questions via sparse deep learning. *IEEE Transactions on Knowledge and Data Engineering, 27(8),* 2107–2119. DOI 10.1109/TKDE.2015.2399298.
5. Kuo, T. T., Kim, H. E., Ohno-Machado, L. (2017). Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association, 24(6),* 1211–1220. DOI 10.1093/jamia/ocx068.
6. Wu, T. Y., Wang, T., Lee, Y. Q., Zheng, W., Kumari, S. et al. (2021). Improved authenticated key agreement scheme for fog-driven iot healthcare system. *Security and Communication Networks*. DOI 10.1155/2021/6658041.
7. Xiong, H., Huang, X., Yang, M., Wang, L., Yu, S. (2021). Unbounded and efficient revocable attribute-based encryption with adaptive security for cloud-assisted Internet of Things. *IEEE Internet of Things Journal*. DOI 10.1109/JIOT.2021.3094323.
8. Xiong, H., Yao, T., Wang, H., Feng, J., Yu, S. (2021). A survey of public key encryption with search functionality for cloud-assisted IoT. *IEEE Internet of Things Journal*, *9(1),* 401–418. DOI 10.1109/JIOT.2021.3109440.
9. Chen, C. M., Tie, Z., Wang, E. K., Khan, M. K., Kumar, S. et al. (2021). Verifiable dynamic ranked search with forward privacy over encrypted cloud data. *Peer-to-Peer Networking and Applications*, *14*, 2977–2991. DOI 10.1007/s12083-021-01132-3.
10. Xiong, H., Jin, C., Alazab, M., Yeh, K. H., Wang, H. et al. (2021). On the design of blockchain-based ECDSA with fault-tolerant batch verication protocol for blockchain-enabled IoMT. *IEEE Journal of Biomedical and Health Informatics*. DOI 10.1109/JBHI.2021.3112693.
11. Xiong, H., Hou, Y., Huang, X., Zhao, Y., Chen, C. M. (2021). Heterogeneous signcryption scheme from IBC to PKI with equality test for WBANS. *IEEE Systems Journal*, 1–10. DOI 10.1109/JSYST.2020.3048972.
12. Wang, Q., Wang, C., Ren, K., Lou, W., Li, J. (2010). Enabling public auditability and data dynamics for storage security in cloud computing. *IEEE Transactions on Parallel and Distributed Systems, 22(5),* 847–859. DOI 10.1109/TPDS.2010.183.
13. Yu, Y., Xue, L., Au, M. H., Susilo, W., Ni, J. et al. (2016). Cloud data integrity checking with an identity-based auditing mechanism from RSA. *Future Generation Computer Systems, 62,* 85–91. DOI 10.1016/j.future.2016.02.003.

14. Xiong, H., Zhao, Y., Hou, Y., Huang, X., Jin, C. et al. (2020). Heterogeneous signcryption with equality test for IIoT environment. *IEEE Internet of Things Journal*, *8(21)*. DOI 10.1109/JIOT.2020.3008955.
15. Xiong, H., Chen, J., Mei, Q., Zhao, Y. (2020). Conditional privacy-preserving authentication protocol with dynamic membership updating for vanets. *IEEE Transactions on Dependable and Secure Computing*. DOI 10.1109/TDSC.2020.3047872.
16. Yang, X., Wang, C., Zhang, Y., Wei, W. (2009). A new forward-secure threshold proxy re-signature scheme. *2009 IEEE International Conference on Network Infrastructure and Digital Content*, pp. 566–569.
17. Sunitha, N., Amberker, B. (2009). Multi-use unidirectional forward-secure proxy re-signature scheme. *2009 IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*, pp. 1–6.
18. Blaze, M., Bleumer, G., Strauss, M. (1998). Divertible protocols and atomic proxy cryptography. *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 127–144. Espoo, Finland.
19. Ateniese, G., Hohenberger, S. (2005). Proxy re-signatures: New definitions, algorithms, and applications. *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pp. 310–319. Alexandria VA USA.
20. Shao, J., Cao, Z., Wang, L., Liang, X. (2007). Proxy re-signature schemes without random oracles. *International Conference on Cryptology in India*, pp. 197–209. Chennai, India.
21. Libert, B., Vergnaud, D. (2008). Multi-use unidirectional proxy re-signatures. *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 511–520. Alexandria Virginia USA.
22. Yang, P., Cao, Z., Dong, X. (2011). Threshold proxy re-signature. *Journal of Systems Science and Complexity, 24(4),* 816–824. DOI 10.1007/s11424-011-8370-3.
23. Shao, J., Wei, G., Ling, Y., Xie, M. (2011). Unidirectional identity-based proxy re-signature. *2011 IEEE International Conference on Communications (ICC)*, pp. 1–5. Kyoto, Japan.
24. Wang, Z., Lv, W. (2013). Server-aided verification proxy re-signature. *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1704–1707. Melbourne, VIC, Australia.
25. Patonico, S., Shabisha, P., Braeken, A., Touhafi, A., Steenhaut, K. (2020). Elliptic curve-based proxy re-signcryption scheme for secure data storage on the cloud. *Concurrency and Computation: Practice and Experience, 32(17),* e5657. DOI 10.1002/cpe.5657.
26. Luo, F., Al-Kuwari, S., Susilo, W., Duong, D. H. (2021). Attribute-based proxy re-signature from standard lattices and its applications. *Computer Standards & Interfaces, 75,* 103499. DOI 10.1016/j.csi.2020.103499.
27. Chen, W., Li, J., Huang, Z., Gao, C., Yiu, S. et al. (2021). Lattice-based unidirectional infinite-use proxy re-signatures with private re-signature key. *Journal of Computer and System Sciences, 120,* 137–148. DOI 10.1016/j.jcss.2021.03.008.
28. Sun, Y., Lu, R., Lin, X., Shen, X., Su, J. (2010). An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *IEEE Transactions on Vehicular Technology, 59(7),* 3589–3603. DOI 10.1109/TVT.2010.2051468.
29. Xiong, H., Chen, Z., Li, F. (2012). Efficient privacy-preserving authentication protocol for vehicular communications with trustworthy. *Security and Communication Networks, 5(12),* 1441–1451. DOI 10.1002/sec.515.
30. Xiong, H., Wu, Y., Jin, C., Kumari, S. (2020). Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT. *IEEE Internet of Things Journal, 7(12),* 11713–11724. DOI 10.1109/JIoT.6488907.
31. Xiong, H., Kang, Z., Chen, J., Tao, J., Yuan, C. et al. (2020). A novel multiserver authentication scheme using proxy resignature with scalability and strong user anonymity. *IEEE Systems Journal, 15(2),* 2156–2167. DOI 10.1109/JIOT.2020.2999510.
32. Xiong, H., Zhou, Z., Wang, L., Zhao, Z., Huang, X. et al. (2021). An anonymous authentication protocol with delegation and revocation for content delivery networks. *IEEE Systems Journal*, 1–12. DOI 10.1109/JSYST.2021.3113728.

33. Wang, B., Li, B., Li, H. (2013). Panda: Public auditing for shared data with efficient user revocation in the cloud. *IEEE Transactions on Services Computing, 8(1),* 92–106. DOI 10.1109/TSC.2013.2295611.
34. He, K., Huang, C., Yang, K., Shi, J. (2015). Identity-preserving public auditing for shared cloud data. *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)*, pp. 159–164. Portland, OR, USA.
35. Liu, X., Sun, W., Lou, W., Pei, Q., Zhang, Y. (2017). One-tag checker: Message-locked integrity auditing on encrypted cloud deduplication storage. *IEEE Conference on Computer Communications*, pp. 1–9. Atlanta, GA, USA.
36. Luo, Y., Xu, M., Huang, K., Wang, D., Fu, S. (2018). Efficient auditing for shared data in the cloud with secure user revocation and computations outsourcing. *Computers & Security, 73,* 492–506. DOI 10.1016/j.cose.2017.12.004.
37. Rabaninejad, R., Ahmadian, M., Asaar, M. R., Aref, M. R. (2019). A lightweight auditing service for shared data with secure user revocation in cloud storage. *IEEE Transactions on Services Computing*. DOI 10.1109/TSC.2019.2919627.
38. Huang, C., Lu, R., Lin, X., Shen, X. (2018). Secure automated valet parking: A privacy-preserving reservation scheme for autonomous vehicles. *IEEE Transactions on Vehicular Technology, 67(11),* 11169–11180. DOI 10.1109/TVT.2018.2870167.
39. Song, W., Wu, Y., Cui, Y., Liu, Q., Shen, Y. et al. (2021). Public integrity verification for data sharing in cloud with asynchronous revocation. *Digital Communications and Networks*. DOI 10.1016/j.dcan.2021.02.002.
40. Dodis, Y., Katz, J., Xu, S., Yung, M. (2003). Strong key-insulated signature schemes. *International Workshop on Public Key Cryptography*, pp. 130–144. Miami, FL, USA.
41. Weng, J., Liu, S., Chen, K., Li, X. (2006). Identity-based key-insulated signature with secure key-updates. *International Conference on Information Security and Cryptology*, pp. 13–26. Beijing, China.
42. Pointcheval, D., Stern, J. (2000). Security arguments for digital signatures and blind signatures. *Journal of Cryptology, 13(3),* 361–396. DOI 10.1007/s001450010003.
43. Yang, X., Chen, C., Ma, T., Wang, J., Wang, C. (2018). Revocable identity-based proxy re-signature against signing key exposure. *PLos One, 13(3),* e0194783. DOI 10.1371/journal.pone.0194783.
44. Lynn, B. (2010). The Pairing-Based Cryptography (PBC) library. http://crypto.stanford.edu/pbc.