Revocable identity-based proxy re-signature against signing key exposure

Xiaodong Yang1,2*, Chunlin Chen1, Tingchun Ma1, Jinli Wang1, Caifen Wang1

1 College of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu, China, 2 State Key Laboratory of Cryptology, Beijing, China

* These authors contributed equally to this work.
Current address: College of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu, China
*y200888@163.com

Abstract

Identity-based proxy re-signature (IDPRS) is a novel cryptographic primitive that allows a semi-trusted proxy to convert a signature under one identity into another signature under another identity on the same message by using a re-signature key. Due to this transformation function, IDPRS is very useful in constructing privacy-preserving schemes for various information systems. Key revocation functionality is important in practical IDPRS for managing users dynamically; however, the existing IDPRS schemes do not provide revocation mechanisms that allow the removal of misbehaving or compromised users from the system. In this paper, we first introduce a notion called revocable identity-based proxy re-signature (RIDPRS) to achieve the revocation functionality. We provide a formal definition of RIDPRS as well as its security model. Then, we present a concrete RIDPRS scheme that can resist signing key exposure and prove that the proposed scheme is existentially unforgeable against adaptive chosen identity and message attacks in the standard model. To further improve the performance of signature verification in RIDPRS, we introduce a notion called server-aided revocable identity-based proxy re-signature (SA-RIDPRS). Moreover, we extend the proposed RIDPRS scheme to the SA-RIDPRS scheme and prove that this extended scheme is secure against adaptive chosen message and collusion attacks. The analysis results show that our two schemes remain efficient in terms of computational complexity when implementing user revocation procedures. In particular, in the SA-RIDPRS scheme, the verifier needs to perform only a bilinear pairing and four exponentiation operations to verify the validity of the signature. Compared with other IDPRS schemes in the standard model, our SA-RIDPRS scheme greatly reduces the computation overhead of verification.

Introduction

A digital signature provides security services such as data integrity, authentication and non-repudiation; therefore, it is one a key technology to ensure information security. In particular,
digital signatures can be combined with passwords [1, 2], smart cards [3], biometrics [4], chaotic parallel keyed hash functions [5] and other technologies to achieve identity authentication of parties in communication. Consequently, they have practical applications in systems such as smart cities [6], body area networks [7], the Internet of Things [8] and ubiquitous networks [9]. To meet the security needs of different practical scenarios, researchers have proposed a series of digital signature variants. For instance, a blind signature can effectively protect the content of signed messages; therefore, it is applied to e-commerce, e-voting, e-currency and other systems to protect participants’ interests [10, 11]. A group signature allows any member of a group to sign messages on behalf of the entire group anonymously; therefore, group signatures have been used to protect the privacy and authenticate the identity of vehicles in vehicular ad-hoc networks [12]. A ring signature is a type of group signature with no administrator that can be used in cloud computing and block-chains to achieve unconditional anonymity to protect a user’s privacy [13, 14]. In many scenarios, we need, for example, to verify Alice’s signature, but we do not know her public key or her public key has expired. We want to transform Alice’s signature into Bob’s signature, because Bob’s public key is available. To achieve this transformation, Blaze et al. [15] introduced the concept of a proxy re-signature (PRS). In a PRS scheme, a semi-trusted proxy is allowed to transform Alice’s signature on a message into Bob’s signature on that same message. However, the proxy cannot independently generate a valid signature on any message on behalf of Alice or Bob. Instead, the proxy re-signature functions as a signature conversion method; consequently, it has been widely used in areas such as cross-domain authentication, digital rights management, privacy preservation and auditing in cloud computing environments.

Ateniese and Hohenberger [16] provided a security model for PRS and presented two concrete schemes in 2005: one is multi-use and the other is single-use. Generally speaking, in a multi-use PRS scheme, a proxy can transform a signature from Alice to Bob into a signature from Bob to Carol, but a proxy cannot further convert a transformed signature in a single-use PRS scheme. In the proxy re-signature field, a multi-use PRS is more useful than is a single-use PRS. After Ateniese and Hohenberger’s seminal work [16], other PRS schemes with special properties were proposed, including the universally composable secure PRS [17], the certificateless PRS [18] and the threshold PRS [19]. In particular, Shao et al. [20] proposed an identity-based proxy re-signature (IDPRS) scheme to address the problem of certificate management in traditional PRS schemes. In IDPRS, the user’s email address or other unique identifying information is used as the public key, and the user’s private key is generated by a trusted private key generator (PKG). IDPRS allows a semi-trusted proxy to convert a signature under one identity to another signature under another identity on the same message, but the proxy is unable to produce any signature on behalf of any of these two identities. As a result, IDPRS eliminates the requirement for certificates and simplifies key management, but challenges still remain to be addressed in practical applications.

Shao et al. [20] proposed the first IDPRS scheme in the standard model. Later, Feng et al. [21] proposed a secure IDPRS scheme, but it was not multi-use. Hu et al. also [22] presented an IDPRS scheme without random oracles, but its security relies on a strong difficult problem assumption. Tian [23] proposed an IDPRS scheme from lattices in the random oracle model, but the size of the signature was relatively large and its practicality was poor. In addition, Wang et al. [24] constructed two server-aided proxy re-signature schemes that were provably secure in the random oracle model, but the second scheme cannot resist a collusion attack from the server and a malicious proxy. Moreover, Canetti et al. [25] found that the provably secure scheme in the random oracle model might be insecure in reality when the random
oracle is instantiated by a specific hash function. Therefore, it is of practical significance to construct secure RIDPRS schemes in the standard model.

The existing IDPRS schemes do not consider the problem of user revocation [20–23]. A revocation mechanism is essential for practical identity-based cryptosystem [26]. If a user’s key is compromised or a user’s authorization expires, that user needs to be revoked from the system. When users have been revoked, they should no longer be able to use their previous private keys to gain access to sensitive data or to generate valid digital signatures. Researchers have designed a series of revocable cryptographic schemes to achieve user revocation in identity-based settings [27–29]. The main idea behind these schemes is that the PKG periodically updates the private keys of non-revoked users. Tsai et al. [30] proposed a revocable identity-based signature scheme in the standard model, in which a user’s signing key consists of a long-term secret key and a periodically changed update key. However, Tsai et al.’s scheme [30] is insecure against a signing key exposure attack: an adversary can obtain a fixed secret key from a compromised signature key and then combine it with subsequent update keys to forge the signature of any message. Lian et al. [31] proposed a revocable attribute-based signature scheme, but Wei et al. [32] revealed that Lian et al.’s scheme is vulnerable to signing key exposure. If a signature scheme can resist the signing key exposure attack, then the users store their secret keys on physical devices with relatively high security levels, while the update keys can be stored on devices with relatively low security levels (such as mobile phones, smart cards, etc.). However, to the best of our knowledge, no identity-based proxy re-signature scheme with a user revocation mechanism is available. Therefore, how to design a revocable identity-based proxy re-signature (RIDPRS) scheme is an open and interesting question.

In this paper, we formally define the syntax of RIDPRS and present a security model for RIDPRS against signing key exposure. Based on Shao et al.’s IDPRS scheme [20], we design a bidirectional and multi-use RIDPRS scheme. In the proposed scheme, the PKG’s master secret key is divided into two parts: one part is used to construct the user’s fixed secret key, and the other part is used to generate periodically changed update keys for the user. Only a non-revoked user can re-randomize their secret key and update key to generate a corresponding signing key. Consequently, our scheme can not only effectively revoke a user from the system, but also resist signing key exposure attacks. Furthermore, we introduce a new cryptographic primitive called server-aided revocable identity-based proxy re-signature (SA-RIDPRS), which is particularly suitable for verifiers with limited computing power. SA-RIDPRS allows the verifier to delegate most of the computational work involved in signature verification to a server with powerful computing capabilities; the verifier needs to perform only a small number of computational operations to verify the legitimacy of the signature. In addition, we formalize the security model for SA-RIDPRS. Based on our RIDPRS scheme, we also construct an SA-RIDPRS scheme and prove that the proposed scheme is secure against adaptive chosen message and collusion attacks. The results of our analysis show that our two schemes are bidirectional and multi-use, and they achieve user revocation functionality while maintaining efficiency in terms of computational complexity and storage overhead. In our SA-RIDPRS scheme, the verifier verifies the validity of the signature with minimal computational cost by executing the server-aided verification algorithm in conjunction with a server, which efficiently reduces the computational cost of the verifier.

The rest of this paper is organized as follows. Section 2 reviews some preliminaries used in our schemes. Section 3 presents security notions for RIDPRS and SA-RIDPRS. Section 4 constructs an IDPRS scheme and an SA-RIDPRS scheme and gives their security proof and performance analysis. Finally, Section 5 concludes the paper.
Preliminaries

In this section, we briefly review bilinear parings and the computational Diffie-Hellman (CDH) assumption.

Bilinear parings

Let $p$ be a prime, $G_1$ and $G_2$ be two multiplicative cyclic groups of order $p$, and $g$ be a generator of $G_1$. An efficiently computable bilinear paring $e: G_1 \times G_1 \rightarrow G_2$ is a map with the following properties [16]:

- **Bilinear**: $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$ for any $a, b \in \mathbb{Z}_p$.
- **Non-degenerate**: $e(g, g) \neq 1$.

Complexity assumption

The security of our scheme depends on the hardness of the CDH problem. Let $G_1$ be a cyclic group of prime order $p$, and $g$ be a generator of $G_1$. Given $(g, g^a, g^b) \in G_1^3$, where $a, b \in \mathbb{Z}_p$, the CDH problem is to compute $g^{ab} \in G_1$.

**Definition 1.** We say that the CDH assumption holds if no probabilistic polynomial-time (PPT) algorithm can solve the CDH problem in $G_1$ with a non-negligible probability [20].

Formal definition and security model

Security notions of RIDPRS

An RIDPRS scheme consists of the following eight algorithms (Setup, Extract, KeyUp, SKGen, ReKey, Sign, ReSign, Verify):

- **Setup($\lambda$) $\rightarrow (pp, msk)$**: Given a security parameter $\lambda$, this algorithm outputs public parameters $pp$ and a master secret key $msk$ of the PKG.
- **Extract($pp, msk, ID$) $\rightarrow sk_{ID}$**: Given $pp, msk$ and a user’s identity $ID$, this algorithm outputs a secret key $sk_{ID}$ of the identity $ID$.
- **KeyUp($pp, msk, ID, t$) $\rightarrow vk_{ID,t}$**: Given $pp, msk$, an identity $ID$ and a time period $t$, this algorithm outputs an update key $vk_{ID,t}$ with respect to identity $ID$ and time period $t$.
- **SKGen($pp, sk_{ID}, vk_{ID,t}$) $\rightarrow dk_{ID,t}$**: Given $pp$, a secret key $sk_{ID}$ and an update key $vk_{ID,t}$, this algorithm outputs an error symbol ⊥ if the identity $ID$ has been revoked during the time period $t$; otherwise, it outputs a signing key $dk_{ID,t}$ on $(ID, t)$.
- **ReKey($pp, dk_{A,0}, dk_{B,0}$) $\rightarrow rk_{A \rightarrow B,t}$**: Given $pp$ and two signing keys $(dk_{A,0}, dk_{B,0})$ corresponding to identities $(ID_A, ID_B)$ at time period $t$, this algorithm outputs a re-signing key $rk_{A \rightarrow B,t}$ of the proxy.
- **Sign($pp, dk_{ID,t}, M$) $\rightarrow \sigma$**: Given $pp$, a signing key $dk_{ID,t}$ and a message $M$, this algorithm generates a signature $\sigma$ on $M$.
- **ReSign($pp, rk_{A \rightarrow B, ID_A, t, M, \sigma_A}$) $\rightarrow \sigma_B$**: Given $pp$, a re-signing key $rk_{A \rightarrow B}$ and a signature $\sigma_A$ on a message $M$ with respect to identity $ID_A$ and time period $t$, this algorithm outputs $\bot$ if Verify($pp, ID_A, t, M, \sigma_A$) = 0; otherwise, it outputs a signature $\sigma_B$ on $M$ with respect to identity $ID_B$ and time period $t$. 


• **Verify**$(pp, ID, t, M, \sigma) \rightarrow \{0, 1\}$: Given $pp$, an identity $ID$, a time period $t$, a message $M$ and a signature $\sigma$, this algorithm outputs 1 if $\sigma$ is a valid signature of $M$ on $(ID, t)$; otherwise, it outputs 0.

**Correctness.** Let $(pp, msk)$ be the output of the algorithm **Setup**$(\lambda)$. For any message $M$ and any identities $(ID_A, ID_B)$, if $\sigma_A = \text{Sign}(pp, t, dk_{A,t}, M)$ and $\sigma_B = \text{ReSign}(pp, rk_{A\rightarrow B}, ID_A, t, M, \sigma_A)$, then the conditions $\text{Verify}(pp, ID_A, t, M, \sigma_A) = 1$ and $\text{Verify}(pp, ID_B, t, M, \sigma_B) = 1$ must both hold.

The security of an RIDPRS scheme should be existentially unforgeable under adaptive chosen identity and message attacks. Based on the security model of IDPRS in [20] and the security definition of revocable identity-based signature schemes in [30–32], the existential unforgeability of a bidirectional RIDPRS scheme that captures signing key exposure is formally defined by using the following security game between a challenger $B$ and an adversary $A$:

**Setup:** $B$ executes the algorithm **Setup**$(\lambda)$ to generate public parameters $pp$ and the PKG’s master secret key $msk$. Then, $B$ sends $pp$ to $A$.

**Queries:** The adversary $A$ adaptively makes the following queries:

• **Extract-query:** When $A$ asks for a secret key of an identity $ID$, $B$ executes the algorithm **Extract**$(pp, msk, ID)$ and returns the corresponding output $sk_{ID}$ to $A$.

• **KeyUp-query:** When $A$ inquires about an update key with respect to an identity $ID$ and a time period $t$, $B$ executes the algorithm **KeyUp**$(pp, msk, ID, t)$ and returns an update key $vk_{ID,t}$ to $A$.

• **SKGen-query:** When $A$ asks for a signing key with respect to an identity $ID$ and a time period $t$, $B$ first makes an **Extract-query** for $ID$ and a **KeyUp-query** for the tuple $(ID, t)$ to obtain a secret key $sk_{ID}$ and an update key $vk_{ID,t}$ respectively. Then, $B$ runs the algorithm **SKGen**$(pp, sk_{ID}, vk_{ID,t})$ to generate a signing key $dk_{ID,t}$ and sends it to $A$.

• **ReKey-query:** When $A$ requests a re-signing key of two identities $(ID_A, ID_B)$ at time period $t$, $B$ first makes the **SKGen-query** on tuples $(ID_A, t)$ and $(ID_B, t)$ to obtain the corresponding signing keys $dk_{A,t}$ and $dk_{B,t}$ respectively. Afterwards, $B$ runs the algorithm **ReKey**$(pp, dk_{A,t}, dk_{B,t})$ to output a re-signing key $rk_{A\rightarrow B,t}$ and then sends it to $A$.

• **Sign-query:** When $A$ requests a signature on a message $M$ with respect to an identity $ID$ and a time period $t$, $B$ first makes a **SKGen-query** on tuple $(ID, t)$ to obtain a signing key $dk_{ID,t}$. Then, $B$ runs the algorithm **Sign**$(pp, dk_{ID,t}, M)$ and returns a signature $\sigma$ on $M$ to $A$.

**Forgery:** The adversary $A$ finally outputs a forged signature $\sigma'$ on a message $M^*$ with respect to an identity $ID^*$ and a time period $t'$. We say that $A$ wins in the above game if the following conditions hold.

1. $\text{Verify}(pp, ID^*, t', M^*, \sigma') = 1$.
2. $(ID^*, t')$ has never been queried of the **SKGen-query**.
3. $ID^*$ has never been submitted to the **Extract-query**, and $(ID^*, t')$ has never been submitted to the **KeyUp-query**.
4. $ID^*$ has never been submitted to the **ReKey-query**.
5. $(ID^*, t', M^*)$ has never been queried of the **Sign-query**.

**Definition 2.** A bidirectional RIDPRS scheme is said to be existentially unforgeable against adaptive chosen identity and message attacks if for any polynomial-time adversary $A$ the probability of winning in the above game is negligible.
Security notions of SA-RIDPRS

An SA-RIDPRS scheme consists of an RIDPRS scheme and a server-aided verification protocol. Due to the weaker computing power, the verifier is unable to perform complicated cryptographic operations. Therefore, the verifier needs to execute an interactive verification protocol to verify the validity of the signature with the help of a server. Specifically, a bidirectional SA-RIDPRS scheme is a tuple of the following ten algorithms (Setup, Extract, KeyUp, SKGen, ReKey, Sign, ReSign, Verify, SA-Setup, SA-Verify):

- The Setup, Extract, KeyUp, SKGen, ReKey, Sign, ReSign and Verify algorithms are the same as those in the RIDPRS scheme described in Section 3.1.

- SA-Setup(pp) → V String: On input public parameters pp, this algorithm generates a secret string V String that contains the pre-computed information of the verifier.

- SA-Verify(pp, V String, ID, t, M, σ) → {0, 1}: On input pp, V String and a signature σ on a message M with respect to an identity ID and a time period t, this algorithm outputs 1 if the server convinces the verifier that σ is valid; otherwise, it outputs 0.

The security of an SA-RIDPRS scheme includes both the existential unforgeability of RIDPRS and the soundness of the algorithm SA-Verify. Existential unforgeability ensures that an attacker cannot generate a valid signature on a new message, whereas soundness ensures that the server cannot convince a verifier that an invalid signature is valid. The unforgeability of a bidirectional SA-RIDPRS scheme is the same as that of the RIDPRS scheme defined in Section 3.1. Based on the soundness of server-aided verification PRS [24], the soundness of the algorithm SA-Verify under adaptive chosen message and collusion attacks is defined by the following security game between a challenger C and an adversary A. In this game, the challenger C acts as a verifier while the adversary A acts as the server. A is allowed to collude with the signer or the proxy; therefore, A can generate or transform the signature of any message. The goal of A is to convince C that an invalid signature is valid. C and A interact as follows:

Setup: C executes the Setup and SA-Setup algorithms to generate the public parameters pp and the string V String, respectively. Then, C sends pp to A.

Queries: A can adaptively make a number of server-aided verification queries to C. For each query on (IDi, ti, Mi, σi), C runs the algorithm SA-Verify with A and then returns the corresponding output to A as a response.

Forgery: The adversary A eventually outputs an identity ID′, a time period t′, a message M′ and a string σ′. Let ΩM′ be the set of all valid signatures on M′, where σ′ /∈ ΩM′. If Verify (pp, ID′, t′, M′, σ′) = 0 and SA-Verify(pp, V String, ID′, t′, M′, σ′) = 1, which means that A has convinced C that σ′ is a valid signature on M′ with respect to the tuple (ID′, t′), then the adversary A wins the game.

Definition 3. The algorithm SA-Verify is soundness if the probability that any polynomial-time adversary A wins in the above game is negligible.

Definition 4. If an RIDPRS scheme is existentially unforgeable against adaptive chosen identity and message attacks, and the algorithm SA-Verify is soundness, then the resulting SA-RIDPRS scheme is said to be secure against adaptive chosen message and collusion attacks.

Our constructions

In this section, we first construct a bidirectional RIDPRS scheme based on the IDPRS scheme of Shao et al. [20]. Then, we extend it to the SA-RIDPRS scheme. Moreover, we provide security proofs and performance analyses for the proposed schemes.
Bidirectional and revocable ID-based proxy re-signature scheme

**Description.** In our RIDPRS scheme, we assume that the length of the identity and the length of the message are $n_i$-bit and $n_m$-bit strings, respectively. We can achieve these by two collision-resistant hash functions $H_1: \{0, 1\}^* \rightarrow \{0, 1\}^{n_i}$ and $H_2: \{0, 1\}^* \rightarrow \{0, 1\}^{n_m}$. The details of our RIDPRS scheme are described as follows:

- **Setup:** Given a security parameter $\lambda$, the PKG does the following:
  
  1. Choose two multiplicative cyclic groups $G_1$ and $G_2$ of prime order $p$, a generator $g$ of $G_1$ and a bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$.
  2. Select a collision-resistant hash function $H: \{0, 1\}^* \rightarrow \{0, 1\}^{n_i}$, where $n_i$ is the fixed length of the output of $H$.
  3. Pick four random elements $g_2, u_0, v_0, w_0 \in G_1$ and three random vectors $\vec{v} = (u_i)$, $\vec{v} = (v_j)$ from $G_1$, where $1 \leq i \leq n_i$, $1 \leq j \leq n_m$, and $1 \leq k \leq n_m$.
  4. Choose two random integers $\alpha, \beta \in Z_p^*$ and compute $g_1 = g^\alpha \cdot g_2^\beta$.
  5. Store the master secret key $msk = (g_2^\beta, g_1^\alpha)$ secretly and publish the public parameters
     
     $pp = (G_1, G_2, \alpha, \beta, g, g_1, g_2, H, u_0, v_0, w_0, \vec{u}, \vec{v}, \vec{w})$.
     
     To simplify the expression, for any identity $ID = (ID_1, \ldots, ID_{n_m}) \in \{0, 1\}^{n_m}$, any string $T = (T_1, \ldots, T_{n_m}) \in \{0, 1\}^{n_m}$ and any message $M = (M_1, \ldots, M_{n_m}) \in \{0, 1\}^{n_m}$, we define the following three functions
     
     $F_{W,1}(ID) = u_0 \prod_{j=1}^{n_m} (u_j)^{m_j}$, $F_{W,2}(T) = v_0 \prod_{j=1}^{n_m} (v_j)^{m_j}$ and $F_{W,3}(M) = w_0 \prod_{j=1}^{n_m} (w_j)^{m_j}$, respectively.

- **Extract:** Given a user’s identity $ID$, the PKG first chooses a random integer $r_{ID} \in Z_p$ and computes a secret key $sk_{ID} = (sk_{ID,1}, sk_{ID,2}) = (g_2^\alpha F_{W,1}(ID)^{r_{ID}}, g_1^\beta)$. Then, the PKG sends $sk_{ID}$ to the user via a secure channel.

- **KeyUp:** Given an identity $ID$ and a time period $t$, the PKG randomly chooses $s_{ID} \in Z_p$ and computes $T_{ID} = H(ID, t)$ and $vk_{ID,t} = (vk_{ID,t,1}, vk_{ID,t,2}) = (g_2^\beta F_{W,2}(T_{ID})^s, g_1^\alpha)$. Then, the PKG sends an update key $vk_{ID,t}$ to the user via a public channel.

- **SKGen:** On input $(ID, t)$, if identity $ID$ has been revoked within time period $t$, the user is unable to generate a valid signing key because he cannot obtain valid update keys. Otherwise, the user with identity $ID$ uses his secret key $sk_{ID} = (sk_{ID,1}, sk_{ID,2})$ and update key $vk_{ID,t} = (vk_{ID,t,1}, vk_{ID,t,2})$ to generate a signing key using the following steps:

  1. Choose two random integers $r_{ID}, s_{ID} \in Z_p$.
  2. Compute $T_{ID} = H(ID, t)$, $dk_{ID,t,2} = sk_{ID,2} \cdot g^{r_{ID}} = g^{r_{ID} + s_{ID}}$, $dk_{ID,t,3} = vk_{ID,t,2} \cdot g^{s_{ID}} = g^{s_{ID} + r_{ID}}$.

  
     $dk_{ID,t,1} = sk_{ID,1} \cdot F_{W,1}(ID)^{s_{ID}} \cdot vk_{ID,t,1} \cdot F_{W,3}(T_{ID})^{r_{ID}} = g^{s_{ID} + r_{ID}} \cdot F_{W,1}(ID)^{s_{ID}} \cdot F_{W,3}(T_{ID})^{r_{ID}}$.

  3. Output a signing key $dk_{ID,t} = (dk_{ID,t,1}, dk_{ID,t,2}, dk_{ID,t,3})$. 
- **ReKey**: On input two signing keys $dk_{A,t_1} = (dk_{A,t_1}, dk_{A,t_2}, dk_{A,t_3})$ and $dk_{B,t_2} = (dk_{B,t_2}, dk_{B,t_3})$ corresponding to two identities $ID_A$ and $ID_B$, the proxy outputs a re-signing key

$$rk_{A\rightarrow B,t_2} = (rk_{A\rightarrow B,t_1}, rk_{A\rightarrow B,t_2}, rk_{A\rightarrow B,t_3})$$

$$= (dk_{B,t_1}/dk_{A,t_1}, dk_{B,t_2}/dk_{A,t_2}, dk_{B,t_3}/dk_{A,t_3}).$$

- **Sign**: Given a message $M$, a signer randomly chooses $r_m \in Z_p$ and uses the signing key $dk_{ID,t} = (dk_{ID,t_1}, dk_{ID,t_2}, dk_{ID,t_3})$ to compute $\sigma_{ID,1} = dk_{ID,t_1} \cdot F_{W,3}(M)^{r_m}$, $\sigma_{ID,2} = dk_{ID,t_2} \cdot \sigma_{ID,3} = dk_{ID,t_3}$ and $\sigma_{ID,4} = g^{r_m}$. Then, the signer outputs a signature $\sigma_{ID} = (\sigma_{ID,1}, \sigma_{ID,2}, \sigma_{ID,3}, \sigma_{ID,4})$ on $M$.

- **ReSign**: Given a re-signing key $rk_{A\rightarrow B} = (rk_{A\rightarrow B,t_1}, rk_{A\rightarrow B,t_2}, rk_{A\rightarrow B,t_3})$ and a signature $\sigma_A = (\sigma_{A,1}, \sigma_{A,2}, \sigma_{A,3}, \sigma_{A,4})$ on a message $M$ with respect to an identity $ID_A$ and a time period $t$, the proxy outputs 1 if $Verify(pp, ID_A, t, M, \sigma_A) = 0$; otherwise, the proxy randomly chooses $r'_m \in Z_p$ and computes $\sigma_{B,1} = \sigma_{A,1} \cdot rk_{A\rightarrow B,t_1} \cdot F_{W,2}(M)^{r_m}$, $\sigma_{B,2} = \sigma_{A,2} \cdot rk_{A\rightarrow B,t_2}$, $\sigma_{B,3} = \sigma_{A,3} \cdot rk_{A\rightarrow B,t_3}$ and $\sigma_{B,4} = \sigma_{A,4} \cdot g^{r'_m}$. Then, the proxy outputs a signature $\sigma_B = (\sigma_{B,1}, \sigma_{B,2}, \sigma_{B,3}, \sigma_{B,4})$ for $M$ with respect to the identity $ID_B$ and the time period $t$.

- **Verify**: Given an identity $ID$, a time period $t$ and a signature $\sigma_{ID} = (\sigma_{ID,1}, \sigma_{ID,2}, \sigma_{ID,3}, \sigma_{ID,4})$ on a message $M$, the verifier first computes $T_{ID} = H(ID, t)$. Then, the verifier checks whether the following equation holds:

$$e(\sigma_{ID,1}, g) = e(g, g) e(F_{W,3}(ID), \sigma_{ID,2}) e(F_{W,3}(T), \sigma_{ID,3}) e(F_{W,3}(M), \sigma_{ID,4}).$$

If the above equation holds, the verifier accepts that $\sigma_{ID}$ is valid and outputs 1; otherwise, the verifier outputs 0.

**Correctness**: For any signature $\sigma_A = (\sigma_{A,1}, \sigma_{A,2}, \sigma_{A,3}, \sigma_{A,4})$ of message $M$ on $(ID_A, t)$, any re-signing key $rk_{A\rightarrow B,t} = (rk_{A\rightarrow B,t_1}, rk_{A\rightarrow B,t_2}, rk_{A\rightarrow B,t_3}) = (dk_{B,t_1}/dk_{A,t_1}, dk_{B,t_2}/dk_{A,t_2}, dk_{B,t_3}/dk_{A,t_3})$ and $ID_B$’s signing key $dk_{B,t_2} = (dk_{B,t_1}, dk_{B,t_2}, dk_{B,t_3})$, where

$$dk_{B,t_1} = g^{s_{B,1}} F_{W,1}(ID_B)^{r_{B,1}} F_{W,2}(T_B)^{s_{B,2}} F_{W,3}(M)^{s_{B,3}}$$

and $dk_{B,t_3} = g^{s_{B,4}}$. Then, we have a re-signature $\sigma_B = (\sigma_{B,1}, \sigma_{B,2}, \sigma_{B,3}, \sigma_{B,4})$ of $M$ with respect to identity $ID_B$ and time period $t$, where

$$\sigma_{B,1} = \sigma_{A,1} \cdot rk_{A\rightarrow B,t_1} \cdot F_{W,2}(M)^{s_{B,1}}$$

$$= dk_{A,t_1} \cdot F_{W,3}(M)^{r_m} \cdot (dk_{B,t_1}/dk_{A,t_1}) \cdot F_{W,2}(M)^{r_m}$$

$$= dk_{B,t_1} \cdot F_{W,3}(M)^{r_m+s_{B,1}}$$

$$= g^{s_{B,1}} F_{W,1}(ID_B)^{r_{B,1}} F_{W,2}(T_B)^{s_{B,2}} F_{W,3}(M)^{r_m+s_{B,1}}.$$  

$$\sigma_{B,2} = \sigma_{A,2} \cdot rk_{A\rightarrow B,t_2} = dk_{A,t_2}/dk_{A,t_1} = dk_{B,t_2} = g^{s_{B,2}},$$

$$\sigma_{B,3} = \sigma_{A,3} \cdot rk_{A\rightarrow B,t_3} = dk_{A,t_3}/dk_{A,t_1} = dk_{B,t_3} = g^{s_{B,3}},$$

$$\sigma_{B,4} = \sigma_{A,4} \cdot g^{s_{B,4}} = g^{s_{B,4}} \cdot g^{s_{B,3}} = g^{s_{B,4}+s_{B,3}}.$$
The following equation shows that our RIDPRS scheme satisfies correctness:

\[
e(\sigma_{B,\cdot}, g) = e(g^{\alpha + \beta}F_{W,1}(ID_B)^{\alpha + \beta}F_{W,2}(T_B)^{\alpha + \beta}F_{W,3}(M)^{\alpha + \beta}, g)
= e(g^{\alpha + \beta}, g)e(F_{W,1}(ID_B)^{\alpha + \beta}, g)e(F_{W,2}(T_B)^{\alpha + \beta}, g)e(F_{W,3}(M)^{\alpha + \beta}, g)
= e(g_2, g^\alpha e)F_{W,1}(ID_B), g^{\alpha + \beta}e(F_{W,2}(T_B), g^\alpha + \beta)e(F_{W,3}(M), g^\alpha + \beta)
= e(g_2, g_3)F_{W,1}(ID_B)F_{W,2}(T_B)F_{W,3}(M, \sigma_{B,\cdot})e(F_{W,3}(M), \sigma_{B,\cdot}).
\]

From the above equation, we can see that \(\sigma_B\) satisfies the condition \(\text{Verify}(pp, ID_B, t, M, \sigma_B) = 1\), so our RIDPRS scheme is correct. The distribution of the re-signature \(\sigma_B\) generated by the algorithm \(\text{ReSign}\) is the same as that of the signature generated by \(ID_B\) himself using the algorithm \(\text{Sign}\), which demonstrates that our RIDPRS scheme is multi-use. By using the re-signing key \(rk_{A \rightarrow B,t}\) between \(ID_A\) and \(ID_B\), it is easy to compute the re-signing key \(rk_{B \rightarrow A,t}\) between \(ID_B\) and \(ID_A\), which implies that our RIDPRS scheme is bidirectional.

**Security analysis.** In our RIDPRS scheme, the algorithm \(\text{SKGen}\) re-randomizes a secret key \(sk_{ID}\) and an update key \(vk_{ID,t}\) to generate a corresponding signing key for an identity \(ID\) and time period \(t\). Even if an adversary obtains the signing key \(dk_{ID,t}\), it is difficult to extract \(ID\)'s secret key \(sk_{ID}\) from \(dk_{ID,t}\). Moreover, the exposure of \(dk_{ID,t}\) at time period \(t\) does not reveal any information concerning other signing keys for identity \(ID\) at other time periods. Therefore, our RIDPRS scheme can resist signing key exposure attacks.

To simplify the security analysis of our RIDPRS scheme, we classify adversaries into two types: external and internal. The main difference between these two types of attackers is that an external adversary is not allowed to make queries about the secret key of the challenged identity whereas an internal adversary is not allowed to make queries about the update key of the challenged identity or the challenged time period. The following two lemmas prove that our RIDPRS scheme is existentially unforgeable against adaptive chosen identity and message attacks.

**Lemma 1.** If a polynomial-time external adversary \(A_1\) breaks our RIDPRS scheme with non-negligible probability, then there exists an algorithm \(B\) that can solve the CDH problem with non-negligible probability.

**Proof.** Assume that an adversary \(A_1\) breaks the existential unforgeability of our RIDPRS scheme with the probability \(\epsilon\) after making at most \(q_{sk}\) secret key queries, \(q_{usk}\) update key queries, \(q_{sk}\) signing key queries, \(q_{usk}\) re-signing key queries and \(q_s\) signing queries. We will construct an algorithm \(B\) to solve the CDH problem in \(G_1\) with the probability at least \(\epsilon_1\) by using \(A_1\)'s forgery. Given a random instance \((g, g^\alpha, g^\beta) \in G_1^3\) of the CDH problem, the goal of \(B\) is to compute \(g^{ab}\). The algorithm \(B\) interacts with \(A_1\) as follows.

**Setup:** \(B\) randomly chooses two integers \(k_u(0 \leq k_u \leq n_u)\) and \(k_m(0 \leq k_m \leq n_m)\), and sets \(s_2 = g^b, l_u = 2(q_{sk} + q_{usk} + q_{sk} + q_s)\) and \(l_m = 2q_s\) such that \(l_u(n_u + 1) < \rho\) and \(l_m(n_m + 1) < \rho\). To generate the public parameters \(pp, B\) executes the following operations:

1. Randomly choose \(x_0, x_1, \ldots, x_u, y_0, y_1, \ldots, y_m \in Z_p\), then compute \(u_i = g_{x_i}^{-k_u}g_{y_i}^{x_i}g_{y_i}^{x_i}g_{y_i}^{x_i}\) and a vector \(\bar{u} = (u_i)\), where \(u_i = g_{x_i}^{x_i}g_{y_i}^{x_i}\) for \(1 \leq i \leq n_u\).
2. Randomly choose \(z_0, z_1, \ldots, z_l \in Z_p\) and then compute \(v_j = g_{z_j}^{x_j}\) and a vector \(\bar{v} = (v_j)\), where \(v_j = g_{z_j}^{x_j}\) for \(1 \leq j \leq n_v\).
3. Randomly choose \(c_0, c_1, \ldots, c_{n_u} \in Z_{l_u}\) and \(d_0, d_1, \ldots, d_{n_m} \in Z_{l_m}\), then compute \(w_k = g_{c_k}^{x_k}g_{d_k}^{x_k}\) and a vector \(\bar{w} = (w_k)\), where \(w_k = g_{c_k}^{x_k}g_{d_k}^{x_k}\) for \(1 \leq k \leq n_v\).
4. Select a collision-resistant hash function \(H : \{0, 1\}^* \rightarrow \{0, 1\}^n\).
5. Choose a random integer $\beta \in \mathbb{Z}_p^*$ and compute $g^\beta = g^{a+\beta}$. This calculation implies that the master secret key is $msk = (g^a, g^\beta)$ but $a$ is unknown to $B$.

6. Send the public parameters $(G_1, G_2, e, p, g, g_1, g_2, H, u_0, v_0, w_0, \bar{u}, \bar{v}, \bar{w})$ to $A_1$.

For any identity $ID = (ID_1, \ldots, ID_n) \in \{0, 1\}^n$, any string $T = (T_1, \ldots, T_n) \in \{0, 1\}^n$ and any message $M = (M_1, \ldots, M_n) \in \{0, 1\}^n$, we define the following five functions:

\[ F(ID) = x_0 - l_s k_s + \sum_{i=1}^{n_s} x_i ID_i, \]
\[ J(ID) = y_0 + \sum_{i=1}^{n_s} y_i ID_i, \]
\[ E(T) = z_0 + \sum_{j=1}^{n} z_j T_j, \]
\[ K(M) = c_0 - l_m k_m + \sum_{k=1}^{n_m} c_k M_k, \]
\[ L(M) = d_0 + \sum_{k=1}^{n_m} d_k M_k. \]

Thus, we have

\[ F_{W,1}(ID) = \prod_{i=1}^{n_s} (u_i)^{ID_i} = g^F(ID) g^{J(ID)}, \]
\[ F_{W,2}(T) = \prod_{j=1}^{n} (v_j)^{T_j} = g^{E(T)}, \]
\[ F_{W,3}(M) = \prod_{k=1}^{n_m} (w_k)^{M_k} = g^{K(M)} g^{L(M)}. \]

**Queries**: The adversary $A_1$ can issue the following types of queries adaptively:

- **Extract-query**: When $A_1$ asks for a secret key query on an identity $ID$, $B$ first computes $F(ID)$. We note that $F(ID) \equiv 0 \pmod{p}$ implies $F(ID) \equiv 0 \pmod{l_s}$ (and thus $F(ID) \not\equiv 0 \pmod{l_s}$ implies $F(ID) \not\equiv 0 \pmod{p}$). If $F(ID) \equiv 0 \pmod{l_s}$, $B$ aborts; otherwise, $B$ randomly chooses $r_{ID} \in \mathbb{Z}_p$ computes

\[ sk_{ID} = (sk_{ID,1}, sk_{ID,2}) \]
\[ = \left( g^{F(ID)} F_{W,1}(ID)^y, \left( g^{F(ID)} F_{W,1}(ID)^y \right)^{-1} \right), \]

and then sends the secret key $sk_{ID}$ to $A_1$.

- **KeyUp-query**: When $A_1$ issues an update key query on an identity $ID$ and a time period $t$, $B$ randomly chooses $s_{ID} \in \mathbb{Z}_p$ and computes $T_{ID} = H(ID, t)$. Then, $B$ uses the secret value $\beta$ to compute

\[ vk_{ID,t} = (vk_{ID,t,1}, vk_{ID,t,2}) = \left( g^{\beta} F_{W,2}(T_{ID})^y, g^{\beta} \right), \]

and returns an update key $vk_{ID,t}$ to $A_1$.

- **SKGen-query**: For a signing key query on $(ID, t)$, if identity $ID$ has been revoked at time period $t$, $B$ outputs $\bot$. Otherwise, $B$ randomly chooses $r_{ID}', s_{ID}' \in \mathbb{Z}_p$ and performs an Extract-query on $ID$ and a KeyUp-query on $(ID, t)$ to obtain a secret key $sk_{ID} = (sk_{ID,1}, sk_{ID,2})$ and an update key $vk_{ID,t} = (vk_{ID,t,1}, vk_{ID,t,2})$, respectively. Then, $B$ computes $T_{ID} = H(ID, t)$, and computes

\[ dk_{ID,t,1} = sk_{ID,t,1} F_{W,1}(ID)^y \cdot vk_{ID,t,1} F_{W,2}(T_{ID})^y, \]
\[ dk_{ID,t,2} = sk_{ID,t,2} \cdot g^y, \]
\[ dk_{ID,t,3} = vk_{ID,t,2} \cdot g^{ty}, \]

and returns a signing key $dk_{ID,t} = (dk_{ID,t,1}, dk_{ID,t,2}, dk_{ID,t,3})$ to $A_1$. 

PLOS ONE | https://doi.org/10.1371/journal.pone.0194783 March 26, 2018 10 / 22
• **ReKey-query:** When $A_1$ issues a re-signing key query on two identities ($ID_A, ID_B$) and a time period $t$, $B$ first makes $SKGen-query$ on $(ID_A, t)$ and $(ID_B, t)$ to obtain the corresponding signing keys $dk_{A,t}$ and $dk_{B,t}$ respectively. Then, $B$ executes the algorithm $ReKey(pp, dk_{A,t}, dk_{B,t})$ to produce a re-signing key $rk_{A→B,t}$ and sends it to $A_1$.

• **Sign-query:** When $A_1$ issues a signature query on a message $M$ with respect to an identity $ID$ and a time period $t$, $B$ considers the following two cases:

  - Case 1: If $F(ID) \neq 0 \pmod {l_n}$, $B$ makes a $SKGen-query$ on $(ID, t)$ to obtain a signing key $dk_{ID,t}$. Then, $B$ executes the algorithm $Sign(pp, dk_{ID,t}, M)$ to produce a signature $\sigma$ on $M$ and returns it to $A_1$.

  - Case 2: If $F(ID) = 0 \pmod {l_n}$, $B$ further considers the following two subcases:

    * Case 2.1: If $K(M) = 0 \pmod {l_m}$, $B$ aborts.

    * Case 2.2: If $K(M) \neq 0 \pmod {l_m}$, $B$ randomly chooses $\tilde{r}_{m}, \tilde{\tilde{r}}_{m}, r_{m} \in \mathbb{Z}_p^*$, and computes:

    $T_{ID} = H(ID, t), \sigma_1 = g^{r_{m}}, \sigma_3 = g^{\tilde{r}_{m}}, \sigma_4 = \left( g^a \right)^{-1} K(M) g^{\tilde{\tilde{r}}_{m}}$ and

    \[
    \sigma_1 = g^a g^{r_{m}} F_{W_{1}}(ID)^{r_{m}} F_{W_{2}}(T_{ID})^{r_{m}} (g^a)^{-1} \left( K(M) \right) g^{\tilde{\tilde{r}}_{m}} F_{W_{3}}(M)^{r_{m}}. 
    \]

  Then, $B$ returns a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on $M$ to $A_1$.

**Forgery:** The adversary $A_1$ finally outputs a signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*)$ on a message $M^*$ with respect to an identity $ID^*$ and a time period $t^*$. If $F(ID^*) \neq 0 \pmod {p}$ or $K(M^*) \neq 0 \pmod {p}$, $B$ aborts. Otherwise, to solve the instance of the CDH problem, $B$ computes $T_{ID^*} = H(ID^*, t^*)$ and outputs $g^{ab}$ as follows:

\[
\frac{\sigma_1^*}{g^a} = \frac{g^{a+b} F_{W_{1}}(ID^*)^{r_{m}} F_{W_{2}}(T_{ID^*})^{r_{m}} F_{W_{3}}(M^*)^{r_{m}}}{g^{a+b} (g^{r_{m}})^{r_{m}} (g^{\tilde{r}_{m}})^{r_{m}} (g^{\tilde{\tilde{r}}_{m}})^{r_{m}}} \\
= \frac{g^{a+b} g^{r_{m}} (g^{r_{m}})^{r_{m}} (g^{\tilde{r}_{m}})^{r_{m}} (g^{\tilde{\tilde{r}}_{m}})^{r_{m}}}{(g^{r_{m}})^{r_{m}} (g^{\tilde{r}_{m}})^{r_{m}} (g^{\tilde{\tilde{r}}_{m}})^{r_{m}}} \\
= g^a \quad (\text{since} \quad F(ID^*) = K(M^*) = 0 \pmod {p})
\]

Now, we analyse the probability that $B$ does not abort in the above simulation. If $B$ completes the simulation without aborting, then the following events must have occurred:

- In the Extract-query, SKGen-query and ReKey-query phases, the condition $F(ID) \neq 0 \pmod {l_n}$ must be satisfied for any queried identity $ID$.

- In the Sign-query phase, $F(ID) \neq 0 \pmod {l_n}$ or $K(M) \neq 0 \pmod {l_n}$ must be satisfied for any queried identity $ID$ and message $M$.

- In the forgery phase, $F(ID^*) = 0 \pmod {p}$ and $K(M^*) = 0 \pmod {p}$ must be satisfied for the challenged identity $ID^*$ and message $M^*$.

Similar to the probability analysis described in [20, 33], we define the following events:

- $X_i : F(ID_i) \neq 0 \pmod {l_n}$ for $i = 1, \ldots, q_{ab} + q_{dk} + q_{sk} + q_{t}$. 

\(X^* : F(ID^*) = 0 \mod p\).

\(Y_j : K(M_j) \neq 0 \mod l_m\) for \(i = 1, \ldots, q_s\).

\(Y^* : K(M^*) = 0 \mod p\).

Hence, the probability that \(B\) will not abort in the above simulation is

\[
\Pr[\neg \text{abort}] = \Pr \left[ \bigcap_{i=1}^{q_s} X_i \cap X^* \cap \bigcap_{j=1}^{q_l} Y_j \cap Y^* \right] \\
\geq \Pr \left[ \bigcap_{i=1}^{q_s} X_i \cap X^* \right] \cdot \Pr \left[ \bigcap_{j=1}^{q_l} Y_j \cap Y^* \right] \\
= \Pr[X^*] \cdot \Pr \left[ \bigcap_{i=1}^{q_s} X_i \cap X^* \right] \cdot \Pr \left[ \bigcap_{j=1}^{q_l} Y_j \cap Y^* \right].
\]

Because \(l_n(n_m+1) < p\), we can find that

\[
\Pr[X^*] = \Pr[F(ID^*) = 0 \mod p] \\
\geq \Pr[F(ID^*) = 0 \mod p \cap F(ID^*) = 0 \mod l_n] \\
= \Pr[F(ID^*) = 0 \mod l_n] \Pr[F(ID^*) = 0 \mod p | F(ID^*) = 0 \mod l_n] \\
= \frac{1}{l_n n_m + 1},
\]

and

\[
\Pr \left[ \bigcap_{i=1}^{q_s} X_i \cap X^* \right] = 1 - \Pr \left[ \bigcup_{i=1}^{q_s} \neg X_i \cap X^* \right] \\
\geq 1 - \sum_{i=1}^{q_s} \Pr[\neg X_i \cap X^*] \\
= 1 - \frac{q_s}{l_n}.
\]

Since \(l_m(n_m+1) < p\), we have that

\[
\Pr[Y^*] = \Pr[K(M^*) = 0 \mod p] \\
\geq \Pr[K(M^*) = 0 \mod p \cap K(M^*) = 0 \mod l_m] \\
= \Pr[K(M^*) = 0 \mod l_m] \Pr[K(M^*) = 0 \mod p | K(M^*) = 0 \mod l_m] \\
= \frac{1}{l_m n_m + 1},
\]

\[
\Pr \left[ \bigcap_{j=1}^{q_l} Y_j \cap Y^* \right] = 1 - \Pr \left[ \bigcup_{j=1}^{q_l} \neg Y_j \cap Y^* \right] \\
\geq 1 - \sum_{j=1}^{q_l} \Pr[\neg Y_j \cap Y^*] \\
= 1 - \frac{q_l}{l_m}.
\]
As \( l_u = 2(q_{ak} + q_{dk} + q_{rk} + q_i) \) and \( l_m = 2q_i \) we can obtain the resulting probability

\[
\Pr[-\text{abort}] = \frac{1}{l_u n_u + 1} \left( 1 - \frac{q_{ak} + q_{dk} + d_k + q_i}{l_u} \right) \frac{1}{l_m n_m + 1} \left( 1 - \frac{q_{rk} + q_{ak} + d_k + q_i}{l_m} \right)
\]

\[
= \frac{1}{2q_i n_u + 1} \left( 1 - \frac{q_{rk} + q_{ak} + d_k + q_i}{2q_i} \right)
\]

\[
= \frac{1}{16(n_u + 1)(n_m + 1)q_i(q_{ak} + q_{dk} + d_k + q_i)}
\]

Therefore, \( B \) can successfully solve the CDH problem in \( G_1 \) with a probability of

\[
e_1 > \frac{e_2}{16(n_u + 1)(n_m + 1)q_i(q_{ak} + q_{dk} + d_k + q_i)}
\]

**Lemma 2.** If a polynomial-time internal adversary \( A_2 \) breaks our RIDPRS scheme with a non-negligible probability, then there exists an algorithm \( B \) that can solve the CDH problem with non-negligible probability.

**Proof.** Assume that an adversary \( A_2 \) breaks the existential unforgeability of our RIDPRS scheme with the probability \( e \) after making at most \( q_{ak} \) secret key queries, \( q_{dk} \) update key queries, \( q_{rk} \) signing key queries, \( q_{ak} \) re-signing key queries and \( q_i \) signing queries. We can construct an algorithm \( B \) that solves the CDH problem in \( G_1 \) with a probability \( e_2 \) using the adversary’s forgery \( A_2 \). \( B \) is given a random instance \((g, g^a, g^b) \in G_1^3\) of the CDH problem. To calculate \( g^{ab} \), \( B \) interacts with \( A_2 \) as follows:

**Setup:** \( B \) randomly chooses two integers \( k_u(0 \leq k_u \leq n_u) \) and \( k_m(0 \leq k_m \leq n_m) \), and sets \( g_2 = g^k, l_u = 2(q_{ak} + q_{dk} + q_{rk} + q_i) \) and \( l_m = 2q_i \) such that \( l_u(n_u + 1) < p \) and \( l_m(n_m + 1) < p \). Then, \( B \) generates public parameters \( pp \) according to the following steps:

1. Randomly choose \( x_0, x_1, \ldots, x_n \in Z_p \) and \( y_0, y_1, \ldots, y_n \in Z_p^* \); then, compute \( v_0 = g_2^{-l_u x_0} g_0^y \) and a vector \( \vec{v} = (v_j) \), where \( v_j = g_2^{y_j} g_0^y \) for \( 1 \leq j \leq n_u \).
2. Randomly choose \( z_0, z_1, \ldots, z_n \in Z_p^* \); then, compute \( u_0 = g_0^z \) and a vector \( \vec{u} = (u_i) \), where \( u_i = g_0^z \) for \( 1 \leq i \leq n_u \).
3. Select a collision-resistant hash function \( H : \{0, 1\}^* \rightarrow \{0, 1\}^{n_u} \).
4. Choose a random integer \( \tau \in Z_p^* \) and compute \( g_1 = g_0^{g_1^\tau} \), which implies that the master secret key is \( msk = (g_2^a, g_1^b) \), but \( a \) is unknown to \( B \).
5. Send the public parameters \((G_1, G_2, \tau, g, g_1, g_2, H, u_0, v_0, w_0, \vec{u}, \vec{v}, \vec{w}) \) to \( A_2 \).

For an identity \( ID = (ID_1, \ldots, ID_n) \in \{0, 1\}^n \), a string \( T = (T_1, \ldots, T_n) \in \{0, 1\}^n \) and a message \( M = (M_1, \ldots, M_n) \in \{0, 1\}^n \), we define five functions:

\[
E_k(ID) = z_0 + \sum_{i=1}^{n_u} z_i ID_i, \quad F_k(T) = x_0 - l_u k_i + \sum_{i=1}^{n_u} x_i T_i, \quad I_k(T) = y_0 + \sum_{i=1}^{n_u} y_i T_i,
\]

\[
K(M) = c_0 - l_m k_m + \sum_{i=1}^{n_u} c_i M_i, \quad L(M) = d_0 + \sum_{i=1}^{n_u} d_i M_i.
\]
Thus, we also have

\[ F_{w,1}(ID) = u_1 \prod_{i=1}^{n} (u_i)^{v_1} = g^{F_{2}(ID)}, \]

\[ F_{w,2}(T) = v_1 \prod_{j=1}^{n} (v_j)^{T_j} = g_2^{F_{2}(T)} g_3^{F_{2}(T)}, \]

\[ F_{w,3}(M) = \prod_{k=1}^{n} (w_k)^{v_3} = g_2^{K(M)} g_3^{L(M)}. \]

**Queries:** \( B \) answers \( A_2 \)’s queries as follows.

- **Extract-query:** Upon receiving a secret key query for identity \( ID \), \( B \) randomly chooses \( r_{ID} \in Z_p \) and computes

  \[ sk_{ID} = (sk_{ID,1}, sk_{ID,2}) = (g_2^{F_{w,1}(ID)} y_1, g_3^{y_2}). \]

  Then, \( B \) sends a secret key \( sk_{ID} \) to \( A_2 \).

- **KeyUp-query:** Upon receiving an update key query on an identity \( ID \) and a time period \( t \), \( B \) first computes \( T_{ID} = H(ID, t) \). If \( F_{1}(T_{ID}) \not\equiv 0 \mod l_2 \), \( B \) aborts. Otherwise, \( B \) randomly chooses \( s_{ID} \in Z_p \) and outputs an update key \( v_{k_{ID},t} = (v_{k_{ID,1},t}, v_{k_{ID,2},t}) \) in response, where

  \[ v_{k_{ID,1},t} = (g^{s})^{-1}_{2} \prod_{i=1}^{n} (w_i)^{v_3} F_{w,2}(T_{ID})^{y_1}, \]

  \[ v_{k_{ID,2},t} = (g^{s})^{-1}_{2} \prod_{i=1}^{n} (w_i)^{v_3} g_3^{y_2}. \]

- **SKGen-query:** \( B \) answers a signing key query in the same way as it does the **SKGen-query** in Lemma 1.

- **ReKey-query:** \( B \) answers a re-signing key query in the same way as it does the **ReKey-query** in Lemma 1.

- **Sign-query:** Upon receiving a signature query on a message \( M \) with respect to an identity \( ID \) and a time period \( t \), \( B \) computes \( T_{ID} = H(ID, t) \) and considers the following two cases:

  - Case 1: If \( F_{2}(T_{ID}) \not\equiv 0 \mod l_3 \), \( B \) generates a signature on \( M \) in the same manner as in Case 1 in Lemma 1.

  - Case 2: If \( F_{2}(T_{ID}) \equiv 0 \mod l_3 \), \( B \) further considers the following two subcases:

    * Case 2.1: If \( K(M) \not\equiv 0 \mod l_3 \), \( B \) aborts.

    * Case 2.2: If \( K(M) \equiv 0 \mod l_3 \), \( B \) creates a signature on \( M \) in the same manner as in Case 2.2 in Lemma 1, except that

      \[ \sigma_1 = g_2^{F_{w,1}(ID)} y_1 F_{w,2}(T_{ID})^{y_2} (g^{s})^{-1}_{2} F_{w,3}(M)^{y_3}. \]

- **Forgery:** The adversary \( A_3 \) finally outputs a signature \( \sigma^* = (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \) on a message \( M^* \) with respect to an identity \( ID^* \) and a time period \( t^* \). \( B \) first computes \( T_{ID^*} = H(ID^*, t^*) \). If
F_2(T_{IP'}) \neq 0 \mod p \text{ or } K(M') \neq 0 \mod p, \mathcal{B} \text{ aborts. Otherwise, } \mathcal{B} \text{ computes } g^{ab} \text{ as follows:}

\begin{align*}
&g_1^a (\sigma_1^* F_2(ID)^{1/2u} F_{W,2}(T_{IP'})^{1/u'} F_{W,3}(M')^{1/n'})
= g_1^a (g_1^{u'} F_2(ID)^{1/2u'} F_{W,2}(T_{IP'})^{1/u'} (g_1^{u''})^{1/n'} (g_1^{u'''} K(M')^{1/n'')})^{1/n''})
= g_1^a (g_1^{u''} (g_1^{u'''} K(M')^{1/n''})^{1/n''})^{1/n''}
= g_1^a \text{ (since } F_2(T_{IP'}) = K(M') = 0 \mod p) 
= g^{ab}.
\end{align*}

Similar to the probabilistic analysis for Lemma 1, the probability that algorithm \mathcal{B} successfully solves the CDH problem in \mathcal{G}_1 is at least \epsilon = \frac{1}{6(n+1)(m+1)q_0 q_4 q_6 q_8 q_9}.

Combining Lemma 1 and Lemma 2, we obtain the following theorem.

**Theorem 1.** Our RIDPRS scheme is existentially unforgeable against adaptive chosen identity and message attacks in the standard model under the CDH assumption.

## Server-aided revocable identity-based proxy re-signature scheme

**Description.** Based on our RIDPRS scheme, we construct a bidirectional SA-RIDPRS scheme in which the verifier can verify the legality of a signature at relatively low computational cost with the help of a server. Our SA-RIDPRS scheme is described as follows:

- **The Setup, Extract, KeyUp, SKGen, ReKey, Sign, ReSign and Verify** algorithms are the same as those in our RIDPRS scheme described in Section 4.1.1.

- **SA-Setup:** Given the public parameters pp, the verifier chooses a random integer \( x \in Z_p \), computes \( K_0 = e(g_2, g_1)^x \) and sets a string V String = (\( x, K_0 \)).

- **SA-Verify:** Given a signature \( \sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \) on a message \( M \) with respect to an identity \( ID \) and a time period \( t \), the verifier interacts with a server through the following protocol:

  1. The verifier computes \( \sigma' = (\sigma_1', \sigma_2', \sigma_3', \sigma_4') = (g_1^x, (g_1^x)^{1/2u'}, (g_1^x)^{1/u''}, (g_1^x)^{1/u'''}) \) and sends the tuple \((ID, t, M, \sigma_1', \sigma_2', \sigma_3', \sigma_4')\) to the server.

  2. The server computes \( T_{ID} = H(ID, t) \), \( K_2 = e(F_{W,2}(ID), \sigma_2') \), \( K_3 = e(F_{W,2}(T_{ID}), \sigma_3') \) and \( K_4 = e(F_{W,3}(M), \sigma_4') \). Then, the server sends \((K_2, K_3, K_4)\) to the verifier.

  3. The verifier first computes \( K_1 = e(\sigma_1', g) \) and checks whether the equation \( K_1 = K_0 \cdot K_2 \cdot K_3 \cdot K_4 \) holds. If the equation holds, the verifier accepts that \( \sigma \) is valid and outputs 1; otherwise, the verifier outputs 0.

**Correctness.** For any signature \( \sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4) = (g_2^{x'}, F_{W,1}(ID)^{1/u} F_{W,2}(T_{ID})^{1/u'} F_{W,3}(M)^{1/n'}, g_1^{u'}, g_1^{u''}, g_1^{u'''}) \) and a string
\[ V \text{ String} = (x, K_0 = e(g_2, g)^y) \], we have

\[
K_1 = e(\sigma_1^y, g) = e((\sigma_1)^y, g)
\]

\[
= e(g_2^{\alpha_1}g_1^{\beta_1}F_{w, 1}(ID)^{\gamma_1}, g_2^{\alpha_2}F_{w, 2}(ID)^{\gamma_2}F_{w, 3}(M)^{\gamma_3}, g)
\]

\[
= e(g_2, g_1^y) e(F_{w, 1}(ID), (g^y)^x) e(F_{w, 2}(ID), (g^y)^x) e(F_{w, 3}(M), (g^x)^y)
\]

\[
= e(g_2, g_1^y) e(F_{w, 1}(ID), (\sigma_1)^y) e(F_{w, 2}(ID), (\sigma_1)^y) e(F_{w, 3}(M), (\sigma_1)^y)
\]

\[
= e(g_2, g_1^y) e(F_{w, 1}(ID), \sigma_1^y) e(F_{w, 2}(ID), \sigma_1^y) e(F_{w, 3}(M), \sigma_1^y)
\]

\[
= K_0 \cdot K_1 \cdot K_2 \cdot K_3.
\]

Thus, the above equation shows that our SA-RIDPRS scheme satisfies the correctness.

**Security analysis.** In Section 4.1.3, we demonstrated that our RIDPRS scheme is existentially unforgeable in the standard model. According to Definition 4, our SA-RIDPRS scheme is secure if we prove that the algorithm **SA-Verify** has the soundness property.

**Lemma 3.** The algorithm **SA-Verify** in our SA-RIDPRS scheme has soundness against adaptive chosen message and collusion attacks.

**Proof.** Assume that the adversary \( \mathcal{A} \) is acting as a server with powerful computational capabilities, and the challenger \( \mathcal{C} \) is acting as a verifier. Because \( \mathcal{A} \) is allowed to collude with the signer or the proxy, \( \mathcal{A} \) has access to the signing or re-signing key to generate a valid signature for any message. First, \( \mathcal{A} \) sends an invalid message-signature pair \( (M', \sigma') \) to \( \mathcal{C} \). Then, \( \mathcal{A} \)'s task is to convince \( \mathcal{C} \) that \( \sigma' \) is a valid signature on a message \( M' \) with respect to an identity \( ID' \) and a time period \( t' \).

**Setup:** \( \mathcal{C} \) first runs the algorithm **Setup** to generate public parameters \( pp \), and runs the **Extract**, **KeyUp** and **SKGen** algorithms to generate a signing key \( dk_{ID', t'} \), with respect to the tuple \( (ID', t') \). Then, \( \mathcal{C} \) randomly chooses \( x' \in Z_p^* \), computes \( K_0 = e(g_2, g_1)^y \) and secretly stores the string \( V \text{ String} = (x', K_0) \). Finally, \( \mathcal{C} \) sends \( (pp, dk_{ID', t'}, ID', t') \) to the adversary \( \mathcal{A} \).

**Queries:** \( \mathcal{A} \) can adaptively make at most \( q_0 \) sever-aided verification queries to \( \mathcal{C} \). For each query on the tuple \( (ID, t, M, \sigma) \), \( \mathcal{C} \) runs the algorithm **SA-Verify** with \( \mathcal{A} \) and \( \mathcal{C} \) returns the resulting output to \( \mathcal{A} \).

**Forgery:** The adversary \( \mathcal{A} \) eventually sends a message-signature pair \( (M', \sigma' = (\sigma_1', \sigma_2', \sigma_3', \sigma_4')) \) to \( \mathcal{C} \), where \( \sigma' \) is not a valid signature on message \( M' \) with respect to identity \( ID' \) and time period \( t' \), which means that **Verify**(\( pp, ID', t', M', \sigma' \)) = 0. Subsequently, \( \mathcal{C} \) interacts with \( \mathcal{A} \) as follows.

1. \( \mathcal{C} \) uses the secret string \( V \text{ String} = (x', K_0) \) to compute

\[
(\sigma_1')' = ((\sigma_1')^x, (\sigma_2')^x, (\sigma_3')^x, (\sigma_4')^x).
\]

Then, \( \mathcal{C} \) sends \( (ID', t', M', (\sigma_1')', (\sigma_2')', (\sigma_3')', (\sigma_4')') \) to \( \mathcal{A} \).

2. \( \mathcal{A} \) computes \( T_{ID'} = H(ID', t') \), \( K'_1 = e(F_{w, 1}(ID'), (\sigma_1')') \), \( K'_2 = e(F_{w, 2}(ID'), (\sigma_1')') \) and \( K'_3 = e(F_{w, 3}(M'), (\sigma_1')') \) and returns \( (K'_2, K'_3, K'_4) \) to \( \mathcal{C} \).

3. \( \mathcal{C} \) computes \( K'_1 = e(\sigma_1', g) \) and verifies whether \( K'_1 = K'_0 \cdot K'_2 \cdot K'_3 \cdot K'_4 \) holds. If this equation holds, then **SA-Verify**(\( pp, V \text{ String}, ID', t', M', \sigma' \)) = 1.

In the following, we analyse the probability that the invalid message-signature pair \( (M', \sigma' = (\sigma_1', \sigma_2', \sigma_3', \sigma_4')) \) satisfies the condition \( K'_1 = K'_0 \cdot K'_2 \cdot K'_3 \cdot K'_4 \). Since \( (\sigma_1')' = (\sigma_1')^x \) and \( x' \) is randomly chosen by \( \mathcal{C} \) from \( Z_p^* \), the probability that \( \mathcal{A} \) can derive \( x' \) from \( (\sigma_1')' = (\sigma_1')^x \) is \( \frac{1}{p-1} \).

As \( x' \in Z_p^* \), the probability of finding an integer \( x' \) that satisfies the condition
$K_i^i = e(g_2, g_1)^v \cdot K_2^2 \cdot K_1^1$ is also $\frac{1}{p^{2 \cdot |p|}}$. Moreover, because $p$ is a large prime, $A$ can convince $C$ that $\sigma'$ is a valid signature for message $M'$ on $(ID', t')$ with a negligible probability ($\frac{1}{p^{2 \cdot |p|}}$).

Therefore, the algorithm $\text{SA-Verify}$ in our SA-RIDPRS scheme satisfies the soundness requirement.

By combining Theorem 1, Lemma 3 and Definition 4, the following theorem can be deduced.

**Theorem 2.** In the standard model, our SA-RIDPRS scheme is secure against adaptive chosen message and collusion attacks.

### Performance analysis

In this section, we discuss the performance of our schemes and other IDPRS schemes in the standard model in terms of their computational cost and security properties. Because multiplication and hash function operations have relatively low computational costs, we consider only bilinear pairing and exponentiation operations, which have relatively high computational costs. In Table 1, "Size" denotes the size of a signature; "Sign", "Verify" and "ReSign" denote the computational cost of the $\text{Sign}$, $\text{Verify}$ and $\text{ReSign}$ algorithms, respectively; and "Revocation" denotes whether the scheme includes a user revocation function. $E$ represents the time to execute an exponentiation operation; $P$ represents the time to execute a bilinear pairing operation; $|p|$ represents the length of an element in $Z_p$; and $|G_1|$ represents the length of an element in group $G_1$.

As shown in Table 1, in our SA-RIDPRS scheme, the verifier must perform one pairing operation and 4 exponentiations to verify the validity of a signature. In contrast, the verifier must compute 4 pairings in our RIDPRS scheme. Based on the results reported in [34], the computational overhead of a bilinear pair is, at minimum, more than five exponentiations in $G_1$. Obviously, our SA-RIDPRS scheme is considerably less computationally intensive than is our RIDPRS scheme; therefore, it substantially reduces the verifier’s computational overhead, making it suitable for limited-resource devices.

In terms of signature size, the signature in our SA-RIDPRS scheme contains 4 elements in $G_1$. The signatures in Shao et al.’s scheme [20] and Feng et al.’s scheme [21] contain 3 elements in $G_1$, while the signature in Hu et al.’s scheme [22] contains 4 elements in $G_1$ and one element in $Z_p$. However, our two schemes are the only ones that provide user revocation functionality and can withstand signing key exposure attacks.

In the signing phase, our SA-RIDPRS scheme and Shao et al.’s scheme [20] require 2 exponentiations to generate a signature on a message, whereas the schemes of Feng et al. [21] and Hu et al. [22] require 3 and 6 exponentiations respectively.

In the re-signing phase, our SA-RIDPRS scheme performs one pairing operation and 6 exponentiations to create a valid re-signature, while the scheme of Shao et al. [20] requires 3

| Scheme                | Size   | Sign | Verify   | ReSign   | Revocation |
|-----------------------|--------|------|----------|----------|------------|
| Shao et al.’s scheme  | $3|G_1|$ | $2E$ | $3P$     | $2E + 3P$| No         |
| Feng et al.’s scheme  | $3|G_1|$| $3E$ | $3E + 5P$| $5E + 4P$| No         |
| Hu et al.’s scheme    | $4|G_1| + |p|$| $6E$ | $4E + 3P$| $6E + 3P$| No         |
| Our RIDPRS scheme     | $4|G_1|$| $2E$ | $4P$     | $2E + 4P$| Yes        |
| Our SA-RIDPRS scheme  | $4|G_1|$| $2E$ | $4E + P$ | $6E + P$ | Yes        |

https://doi.org/10.1371/journal.pone.0194783.t001
pairing operations and 2 exponentiations, and that of Feng et al. [21] must perform 4 pairing operations and 5 exponentiations. Hu et al.’s scheme [22] requires 3 pairing operations and 6 exponentiations.

During signature verification, the computational cost in our SA-RIDPRS scheme involves one pairing operation and 4 exponentiations. The schemes of Shao et al. [20] requires 3 pairings, and that of Feng et al. [21] requires 5 pairings and 3 exponentiations. Hu et al.’s scheme [22] requires 3 pairings and 4 exponentiations. In summary, our SA-RIDPRS scheme has considerably less computational overhead than do the other schemes.

We evaluate the performance of our SA-RIDPRS scheme and the three other IDPRS schemes [20–22]. The experiments are implemented in a hardware environment consisting of an Intel Core i7-6500 CPU 2.5 GHz with 8 GB of RAM. The corresponding simulation algorithms are implemented in the C language using the PBC-0.47-VC library and executed on a Windows 10 operating system. We chose the standard parameter $a_{\text{param}}$ in the PBC library and set the group order $p = 160$ bits. The performance comparison results for all the schemes are shown in Figs 1, 2, 3 and 4.

In Fig 1, we can see that the length of the signature and re-signature in Shao et al.’s scheme [20] are both 532 bits. The length of the signature in Feng et al.’s scheme [21] is 384 bits, but the length of the re-signature is 512 bits. The length of the signature and re-signature in Hu et al.’s scheme [22] are both 532 bits. The length of the signature and re-signature in our SA-RIDPRS scheme are both 512 bits. However, Feng et al.’s scheme [21] is not multi-use because the lengths of signatures and re-signatures are different. While our SA-RIDPRS scheme includes one more element in $G_1$ than the scheme of Shao et al. [20] does, it also effectively solves the user revocation problem. Therefore, compared with the other three schemes, our SA-RIDPRS scheme has comparable efficiency in terms of signature size. This is consistent with the above theoretical analysis.

According to Table 1, the cost of generating a signature in our SA-RIDPRS scheme is equivalent to that of Shao et al.’s scheme [20]. The experimental results shown in Fig 2 indicate that our SA-RIDPRS scheme has lower computational overhead for signature generation than do the other two schemes [21, 22]. Fig 3 demonstrates that our SA-RIDPRS scheme improves on the other three schemes [20–22] in terms of the computational overhead for re-signature generation. From Fig 4, we can observe that the computational cost of the verifier is not related to
Fig 2. Signature generation cost comparison.
https://doi.org/10.1371/journal.pone.0194783.g002

Fig 3. Re-signature generation cost comparison.
https://doi.org/10.1371/journal.pone.0194783.g003

Fig 4. Verifier’s computational cost comparison.
https://doi.org/10.1371/journal.pone.0194783.g004
the size of the message in our SA-RIDPRS scheme. Because the verifier in our SA-RIDPRS scheme offloads most of the time-consuming bilinear pairing computations to the server via the server-aided verification algorithm, it can perform signature verification at a relatively low computational cost. Thus, our SA-RIDPRS scheme greatly reduces the cost of the verifier. Moreover, our SA-RIDPRS scheme is more efficient than are the other schemes [20–22] in terms of the verifier’s computational overhead. Clearly, the results of these experiments are consistent with the results of the theoretical analysis in Table 1.

Conclusions
In this paper, we present two IDPRS schemes that include a user revocation mechanism. In our schemes, the PKG constructs an identity-dependent long-term key for all users and periodically generates update keys related to their identities and time periods for non-revoked users. Only non-revoked users can re-randomize the secret key and the update key to generate a corresponding signing key; therefore, our schemes are resistant to signing key exposure attacks. In the standard model, we prove that our schemes are secure under the CDH assumption. In particular, our SA-RIDPRS scheme substantially reduces the verifier’s computational burden. However, the security of our schemes relies on the hardness assumption of the CDH problem, which can be easily solved by quantum algorithms. In the future, we plan to construct a post-quantum secure RIDPRS scheme.

Acknowledgments
The authors would like to thank anonymous reviewers for their constructive suggestions.

Author Contributions
Data curation: Chunlin Chen, Jinli Wang.
Formal analysis: Caifen Wang.
Validation: Tingchun Ma.
Writing – original draft: Xiaodong Yang.
Writing – review & editing: Xiaodong Yang.

References
1. Kumari S, Chaudhry SA, Wu F, Li X, Farash MS, Khan M.K (2017) An improved smart card based authentication scheme for session initiation protocol. Peer-to-Peer Networking and Applications 10(1): 92–105. https://doi.org/10.1007/s12088-015-0409-0
2. Li X, Niu J, Liao J, Liang W (2015) Cryptanalysis of a dynamic identity-based remote user authentication scheme with verifiable password update. International Journal of Communication Systems 28(2): 374–382. https://doi.org/10.1002/dac.2676
3. Kumari S, Khan MK (2014) More secure smart card-based remote user password authentication scheme with user anonymity. Security and Communication Networks 7(11): 2039–2053. https://doi.org/10.1002/sec.916
4. Li F, Khan MK (2012) A biometric identity-based signcryption scheme. Future Generation Computer Systems 28(1): 306–310. https://doi.org/10.1016/j.future.2010.11.004
5. Wang X, Guo W, Zhang W, Khan MK, Alghathbar K (2013) Cryptanalysis and improvement on a parallel keyed hash function based on chaotic neural network. Telecommunication Systems 52(2): 515–524.
6. Li X, Niu J, Kumari S, Wu F, Choo KKR (2017) A robust biometrics based three-factor authentication scheme for global mobility networks in smart city. Future Generation Computer Systems. Available from: https://doi.org/10.1016/j.future.2017.04.012
7. Li X, Ibrahim MH, Kumari S, Sangaiah AK, Gupta V, Choo KKR (2017) Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. Computer Networks 129: 429–443. [https://doi.org/10.1016/j.comnet.2017.03.013]

8. Li X, Niu J, Kumari S, Wu F, Sangaiah AK, Choo KKR (2018) A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. Journal of Network and Computer Applications 103: 194–204. [https://doi.org/10.1016/j.jnca.2017.07.001]

9. Farash MS, Chaudhry SA, Heydari M, Sadough S, Mohammad S, Kumari S, Khan MK (2017) A lightweight anonymous authentication scheme for consumer roaming in ubiquitous networks with provable security. International Journal of Communication Systems. [https://doi.org/10.1002/dac.3019]

10. Kutubi MAAR, Alam KMR, Tahsin R, Ali GMN, Chong PHJ, Morimoto Y (2017) An off-line electronic payment system based on an untraceable blind signature scheme. KSII Transactions on Internet and Information Systems 11(5): 2628–2645.

11. Kumar M, Katti CP, Saxena PC (2017) A secure anonymous e-voting system using identity-based blind signature scheme. In: Proc. International Conference on Information Systems Security. pp. 29–49.

12. Zhang L, Li C, Li Y, Luo Q, Zhu R (2017) Group signature based privacy protection algorithm for mobile ad hoc network. In: Proc. IEEE Information and Automation. pp. 947–952.

13. Annamalai R, Srikanth J, Prakash M (2015) Integrity and privacy sustenance of shared large scale images in the cloud by ring signature. International Journal of Computer Applications 114(12): 13–18. [https://doi.org/10.5120/20029-1945]

14. Sun SF, Au MH, Liu JK, Yuen TH (2017) RingCT 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In: Proc. European Symposium on Research in Computer Security. pp. 456–474.

15. Blaze M, Bleumer G, Strauss M (1998) Divertible protocols and atomic proxy cryptography. In: Proc. Advances in Cryptology-Eurocrypt’98. pp. 127–144.

16. Ateniese G, Hohenberger S (2005) Proxy re-signatures: new definitions, algorithms, and applications. In: Proc. the 12th ACM conference on Computer and Communications Security. pp. 310–319.

17. Hong X, Gao J, Pan J, Zhang B (2017) Universally composable secure proxy re-signature scheme with effective calculation. Cluster Computing 78(20): 1–10.

18. Hu X, Liu Y, Xu H, Wang J, Zhang X (2015) Analysis and improvement of certificateless signature and proxy re-signature schemes. In: Proc. Advanced Information Technology, Electronic and Automation Control Conference. pp. 166–170.

19. Yang X, Gao G, Li Y, Li Y, Wang C (2015) On-line/off-line threshold proxy re-signature scheme through the simulation approach. Applied Mathematics and Information Sciences 9(6): 3251–3261.

20. Shao J, Cao Z, Wang L, Liang X (2007) Proxy re-signature schemes without random oracles. In: Proc. International Conference on Cryptology in India. pp. 197–209.

21. Feng J, Lan C, Jia B (2014) ID-based proxy re-signature scheme with strong unforgeability. Journal of Computer Applications 34(11): 3291–3294.

22. Hu X, Zhang Z, Yang Y (2009) Identity based proxy re-signature schemes without random oracle. In: Proc. International Conference on Computational Intelligence and Security. pp. 256–259.

23. Tian M (2015) Identity-based proxy re-signatures from lattices. Information Processing Letters 115(4): 462–467. [https://doi.org/10.1016/j.ipl.2014.12.002]

24. Wang Z, Lv W (2013) Server-aided verification proxy re-signature. In: Proc. 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. pp. 1704–1707.

25. Canetti R, Goldreich O, Halevi S (2014) The random oracle methodology, revisited. Journal of the ACM (JACM) 51(4): 557–594. [https://doi.org/10.1145/1008731.1008734]

26. Jia X, He D, Zeadally S, Li L (2017) Efficient revocable ID-based signature with cloud revocation server. IEEE Access 5: 2945–2954. [https://doi.org/10.1109/ACCESS.2017.2676021]

27. Lee K, Lee DH, Park JH (2017) Efficient revocable identity-based encryption via subset difference methods. Designs, Codes and Cryptography 85(1): 39–76. [https://doi.org/10.1007/s10623-016-0287-3]

28. Ishida Y, Shikata J, Watanabe Y (2017) CCA-secure revocable identity-based encryption schemes with decryption key exposure resistance. International Journal of Applied Cryptography 3: 288–311. [https://doi.org/10.1504/IJACT.2017.086229]

29. Shen L, Zhang F, Sun Y, Ma J (2017) An efficient revocable ID-based encryption scheme in the standard model. International Journal of Embedded Systems 9(2): 168–176. [https://doi.org/10.1504/IJES.2017.083736]

30. Tsai TT, Tseng YM, Wu TY (2013) Provably secure revocable ID-based signature in the standard model. Security and Communication Networks 6(10): 1250–1260.
31. Lian Y, Xu L, Huang X (2013) Attribute-based signatures with efficient revocation In: Proc. Intelligent Networking and Collaborative Systems. pp. 573–577.
32. Wei J, Huang X, Hu X, Liu W (2015) Revocable threshold attribute-based signature against signing key exposure. In: Proc. Information Security Practice and Experience. pp. 316–330.
33. Waters B (2005) Efficient identity-based encryption without random oracles. In: Proc. Eurocrypt 2005. pp. 114–127.
34. Fan CI, Sun WZ, Huang VSM (2010) Provably secure randomized blind signature scheme based on bilinear pairing. Computers and Mathematics with Applications 60(2): 285–293. https://doi.org/10.1016/j.camwa.2010.01.021