A Secure Proxy Re-Signature Scheme for IoT

Vinh Duc Tran[1], Phi Thuong Le[2] and Viet Cuong Trinh[3*]

[1]Hanoi University of Science and Technology, Ha Noi, Vietnam. ductv@soict.hust.edu.vn,
Orcid: https://orcid.org/0000-0003-0752-0634

[2]Nghi Son Vocational College, Thanh Hoa, Vietnam. legia1986.com@gmail.com,
Orcid: https://orcid.org/0009-0004-0057-8707

[3*]Faculty of Information and Communication Technology, Hong Duc University, 565 Quang Trung, Thanh Hoa, Vietnam. trinhvietcuong@hdu.edu.vn,
Orcid: https://orcid.org/0000-0003-2076-019X

Received: December 26, 2022; Accepted: January 27, 2023; Published: March 30, 2023

Abstract

In a proxy re-signature (PRS) scheme, a semi-trusted proxy is able to transform a signature signed by a user to a new signature of the same message signed by another user, it however cannot sign on any other message on behalf of those users. PRS thus has found many practical applications such as distributed storage, distributed rights management or cloud infrastructure. To our knowledge, all of the PRS schemes without using random oracle (RO for short) need to use the Waters' hash function, which leads to PRS schemes with large key size and inefficient computing time. In this paper, we propose a new bidirectional PRS scheme without using RO and without using the Waters' hash function. Our scheme is therefore the best PRS scheme without using RO to date in term of efficiency.

Keywords: Proxy Re-Signature, Bidirectional, Standard Model, IoT.

1 Introduction

Proxy re-signature (PRS) scheme allows a semi-trusted proxy to transform a signature signed by a user (called a delegatee) to a new signature of the same message signed by another user (called a delegator). The primary goal of proxy re-signature scheme is to ensure that the proxy is unable to sign on any other message on behalf of delegator or delegatee. Note that there is an another primitive with distinct goals which is called *Proxy signature* [Mambo et~al., 1996]. In this type of scheme, a signer is able to delegate a proxy signer to sign messages on its behalf. So, the difference between these two kinds of primitives is that:

- The proxy in a proxy re-signature scheme only can transform an existing signature (signed by delegatee) to a new signature of other user (delegator); it however cannot generate a new signature (cannot sign on any new message);
- Regarding the proxy signature scheme, the proxy, on behalf of a user, is able to generate a new

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 14, number: 1 (March), pp. 174-188. DOI: *10.58346/JOWUA.2023.I1.014*

*Corresponding author: Faculty of Information and Communication Technology, Hong Duc University, 565 Quang Trung, Thanh Hoa, Vietnam.

signature on any arbitrary message. However, the proxy cannot transform a signature signed by a user to a new signature of other user.

Proxy re-signature scheme has many interesting applications in practice such as Blockchain, distributed storage, distributed rights management, cloud infrastructure, simplified key management, simplified certificate management, we refer the reader to [Blaze et~al., 1998, Ateniese and Hohenberger, 2005, Taban et~al., 2006, Shao et~al., 2007, Chow and Phan, 2008, Libert and Vergnaud, 2008, Hong and Long, 2012, Asaar et~al., 2015, Lee and Kim, 2018, Wu et~al., 2020, Zhang et~al., 2019, Lei et~al., 2020, Lv et~al., 2020, Chaudhari et~al., 2021] for more details. In addition, recently Xiong *et al.* [Xiong et~al., 2021] showed that proxy re-signature scheme could be used for authentication. Besides the efficiency, to estimate a PRS scheme, we rely on the following properties:

- *Private Proxy*: the re-signature key $R_{i\to j}$ is kept private, having a signature signed by user i (delegatee) as well as a transformed signature signed by user j (delegator), one cannot derive $R_{i\to j}$.
- *Bidirectional*: from a re-signature $R_{i\leftrightarrow j}$, one can easily compute $R_{j\leftrightarrow i}$.
- *Multi-use*: the signature which is outputted by $ReSign$ algorithm (see the definition of this algorithm in Section 2.1) can be transformed again using $ReSign$ algorithm, while for single-use it cannot be transformed again.
- *Transparent*: one cannot distinguish between a signature which is outputted by $Sign$ algorithm (see the definition of this algorithm in Section 2.1) and a signature which is outputted by $ReSign$ algorithm.
- *Key optimality*: secret key and public key should be in constant size.
- *Non-transitivity*: proxy is unable to re-delegate their re-signing rights.
- *Threshold*: the proxy is decentralized in the sense that to transform a signature we need at least a threshold number of proxies to collude.
- *Conditional*: a re-signature key is associated with a specific condition, and based on this key one can transform a signature with that specific condition.

We consider in this work the PRS schemes without using RO and has the following properties: *Bidirectional*; *Private Proxy*; *Multi-use*; *Transparent*; *Key optimality*.

Related Work on Proxy Re-signature Scheme

Proxy re-signature scheme was first introduced by Blaze *et. al.* [Blaze et~al., 1998]. We categorize the PRS schemes into two types, using RO and not using RO. Regarding the first one, we can list here several schemes [Ateniese and Hohenberger, 2005, Libert and Vergnaud, 2008, Yuqiao and Ge, 2011, Hong and Long, 2012, Menon, 2012, Asaar et~al., 2015, Lee and Kim, 2018, Wu et~al., 2020, Zhang et~al., 2019]. We note that RO is a powerful technique to design efficient PRS schemes, unfortunately, the security of RO is now still questionable. In fact, there are some known attacks on the schemes which are proved securely in the RO model. Hence, the scheme using RO only achieves a weak level of security. We always want to avoid using RO to design PRS scheme if possible.

We consider in this work the PRS schemes without using RO. Regarding this type of PRS schemes, Shao *et al.* [Shao et~al., 2007] introduced the first PRS scheme using Waters' hash function [Waters, 2005]. Later, Chow *et al.* [Chow and Phan, 2008] showed that this scheme is insecure, and also

showed how to design a generic single-use unidirectional PRS scheme without using RO, but at the cost of transparency. Moreover, this generic method also needs to use the Waters' hash function. Continue making use of the Waters' hash function, Libert *et al.* [Libert and Vergnaud, 2008] introduced the first multi-use unidirectional PRS scheme, Yang *et al.* [Yang et~al., 2011] and Yang *et al.* [Yang et~al., 2015] proposed threshold bidirectional PRS scheme. Lee *et al.* [Lee and Kim, 2018] continued this line of research to introduce a single-use conditional unidirectional PRS scheme.

Recently, Lei *et al.* [Lei et~al., 2020] introduced a bidirectional PRS scheme without using RO, and additionally supports blind signing property [Chaum, 1982]. Lv *et al.* [Lv et~al., 2020] introduced a threshold bidirectional PRS scheme without using RO. Very recently, Chaudhari *et al.* [Chaudhari et~al., 2021] gave a survey on PRS scheme.

Due to the complexity of using public key infrastructure, identity-based PRS scheme was first introduced in [Matsuo, 2007], and then in [Shao et~al., 2011]. These both schemes do not use RO but still need to use the Waters' hash function. Later, in [Menon, 2012, Asaar et~al., 2015] and recently in [Zhang et~al., 2019] the authors introduced three other identity-based PRS schemes, but using RO.

When using Waters' hash function, one can prove the security of their scheme without using RO. However, the downside of this technique is that the scheme has large key size and inefficient computing time, this obviously is not desirable for lightweight device-based applications. To our knowledge, the only PRS scheme without using RO and without using Waters' hash function was proposed by Hu *et al.* [Hu et~al., 2009]. However, Menon [Menon, 2012] later showed that this scheme is in fact insecure.

Contribution of the Paper

Our bidirectional PRS construction relies on the standard signature scheme Pointcheval-Sanders [Pointcheval and Sanders, 2016] (PS scheme) without using RO. More precisely, our scheme retains the efficiency of their scheme while adding proxy re-signature function. When comparing to other PRS schemes without using RO, our scheme is better all of them in term of efficiency. The reason is that other schemes need to use the Waters' hash function while our scheme doesn't. Note that only our scheme achieves the key optimality property, other schemes do not since users in those schemes need to store a large public key to sign and verify. Our scheme thus fits for lightweight device-based applications.

Regarding the security, we use the same security model with other existing PRS schemes to prove the security of our scheme. However, our scheme is secure under the PS 2 assumption introduced in [Pointcheval and Sanders, 2016], while some other PRS schemes are secure under the standard CDH assumption or some modifications of it. Note that, in [Pointcheval and Sanders, 2018], the authors have showed that the hardness of two assumptions PS 2 assumption and q-SDH assumption [Boneh and Boyen, 2008] are the same. Even though, obviously, CDH assumption is a more standard assumption than the PS 2 assumption. On the other hand, it is also fair to say that some other existing PRS schemes may have some properties for which our scheme doesn't have, such as conditional, unidirectional, identity-based, blind or threshold properties.

Overall, our PRS scheme has the following properties: *Private Proxy*, *Bidirectional*, *Multi-use*, *Transparent*, *Key optimality*. We give in Figure 0 the comparison among our PRS scheme and some

other PRS schemes without using RO. From the comparison table we see that our scheme is better in term of public key size, signature size and signing time.

	Sig	Public key	SK	Signing complexity	Verifying complexity
	$2\|\Gamma\|$	$(n_m+3)\|\Gamma\|$	$1\|\Gamma\|$	$3E+(n_m+2)M_\Gamma$	$2P+(n_m+2)M_\Gamma$
	$3\|\Gamma\|$	$(n_{id}+n_m+5)\|\Gamma\|$	$2\|\Gamma\|$	$2E+(n_m+2)M_\Gamma$	$4P+(n_{id}+n_m+2)M_\Gamma$
	$2\|\Gamma\|$	$(n_m+2)\|\Gamma\|$	$1\|\Gamma\|$	$3E+(n_m+2).M_\Gamma$	$3P$
	$2\|\Gamma\|$	$(n_m+4)\|\Gamma\|$	$1\|\Gamma\|$	$3E+n_m.M_\Gamma$	$3P+(n_m+1)M_\Gamma$
	$3\|\Gamma\|$	$(3n_m+4)\|\Gamma\|$	$1\|\Gamma\|$	$5E+2n_m.M_\Gamma$	$10P+2E+(3n_m+2)M_\Gamma$
	$3\|\Gamma\|$	$(2n_m+4)\|\Gamma\|$	$1\|p\|$	$(2n_m+5)E+(2n_m+3)M_\Gamma$	$4P+2n_mE+(2n_m+2)M_\Gamma$
	$2\|\Gamma\|$	$(n_m+2)\|\Gamma\|$	$1\|p\|$	$(n_m+3)E+(n_m+2)M_\Gamma$	$2P$
Ours	$2\|\Gamma\|$	$5\|\Gamma\|$	$2\|p\|$	$1E$	$2P+1E+1M_\Gamma$

	Bidi	Uni	ID-based	Thres	Key Optimal	Transpa	Multi-use	Private	Condi
	No	Yes	No	No	No	No	No	Yes	No
	Yes	No	Yes	No	No	Yes	Yes	Yes	No
	Yes	No	No	Yes	No	Yes	Yes	Yes	No
	Yes	No	No	Yes	No	Yes	Yes	Yes	No
	No	Yes	No	No	No	No	No	Yes	Yes
	Yes	No	No	No	No	Yes	Yes	Yes	No
	Yes	No	No	Yes	No	Yes	Yes	Yes	No
Ours	Yes	No	No	No	Yes	Yes	Yes	Yes	No

Table 1: In the comparison table, n_{id}, n_m are the parameters of the Waters' function. E, P, M_Γ are the exponentiation operation, the pairing operation, the multiplication operation, respectively. Notice that the scheme [Lei et~al., 2020] supports blind property. 1 is [Chow and Phan, 2008]; 2 is [Shao et~al., 2011]; 3 is [Yang et~al., 2011]; 4 is [Yang et~al., 2015]; 5 is [Lee and Kim, 2018]; 6 is [Lei et~al., 2020]; 7 is [Lv et~al., 2020].

2 Security Model for Bidirectional Proxy Re-Signature Scheme

The definition and security model of a bidirectional PRS scheme from [Ateniese and Hohenberger, 2005, Chow and Phan, 2008] are recalled as follows.

Definition

Formally, a bidirectional PRS scheme includes six probabilistic algorithms.

- *Setup*: the inputs of this algorithm is a security parameter λ and the output of the algorithm is the system parameters param.
- *KeyGen*: the inputs of this algorithm is param, the outputs are a public key and a corresponding secret key for the user: (pk, sk).
- *ReKeyGen*: this algorithm takes as input the secret keys of user i, j, outputs a re-signature key $R_{i\leftrightarrow j}$ for the proxy. Note that proxy only knows $R_{i\leftrightarrow j}$, and user i doesn't know the secret key

of user j as well as $R_{i\leftrightarrow j}$. Similar for user j.

- $Sign$: the inputs of this algorithm are param, a message m, and a user's secret key. It outputs a signature σ.
- $ReSign$: : the inputs of this algorithm are param, a message m, a re-signature key $R_{i\leftrightarrow j}$, a public key pk_i and a signature σ_i of user i. It outputs a new signature σ_j for the user j on the message m. Note that if the algorithm takes the public key pk_j and the signature σ_j of user j instead of pk_i and σ_i, then it will output a new signature σ_i for the user i on the message m.
- $Verify$: the inputs of this algorithm are param, a message m, a user's public key, and a signature σ. It outputs 1 if σ is correct and 0 if it is not correct.

$Correctness$: For any $m, pk_i, sk_i, pk_j, sk_j$ are all correct and $\sigma_i = Sign(m, sk_i)$ and $R_{i\leftrightarrow j} = ReKeyGen(sk_i, sk_j)$, then the following equations must verify:

$$Verify(m, pk_i, \sigma_i) = 1$$

and

$$Verify(m, pk_j, ReSign(m, R_{i\leftrightarrow j}, pk_i, \sigma_i)) = 1$$

Adversary's Oracles

We denote Φ a forger that attacks a private bidirectional PRS scheme. In the security game Φ will interact with a challenger X and can ask X the following oracles:

- $RequestPK(i)$: when Φ asks for the public key of user i, X returns public key pk_i.
- $RequestSK(i)$: when Φ asks for the secret key of user i, X returns secret key sk_i.
- $RequestReKey(i, j)$: when Φ asks for the re-signature key for user i, j, X returns the re-signature key $R_{i\leftrightarrow j}$.
- $Sign(m, pk_i)$: when Φ, on the inputs message m and public key pk_i of user i, asks for a corresponding signature, X sends back a valid signature σ_i.
- $ReSign(m, pk_i, pk_j, \sigma_i)$: when Φ, on the inputs message m, public key pk_i, pk_j and σ_i of user i, asks for a corresponding re-signature, X sends back a valid signature σ_j.

Security Model

We use the security model of a private bidirectional PRS scheme from Section 3.5 in [Chow and Phan, 2008]. Intuitively, there are two types of attackers in a private bidirectional PRS scheme. The first one is a forger Φ who represents for a delegatee who tries to forge the signature of a delegator, or a delegator who tries to forge the signature of a delegatee. In this security game, forger Φ can know the public parameters, public keys of all users, either the secret key of delegatee (target user is delegator) or the secret key of delegator (target user is delegatee), signature of target user on any new message except the target message, any re-signed signature except on the target message. The second one is a forger Φ

who represents the proxy. That means, the forger Φ in this security game can know the public parameters, public keys of all users, re-signature key and signature of delegatee or delegator on any new message except the target message. The goal of forger Φ is to forge the signature of delegatee or delegator on the target message. Note that in a private bidirectional PRS scheme, the roles of delegatee and delegator are equivalent.

More formally, a private bidirectional PRS scheme is secure if a forger has negligible probability to win the following games.

External Security Game: the challenger X and the forger Φ perform the game as follows.

Initialization: X uses the *Setup* algorithm to generate param and gives param to Φ.

Queries: Φ asks X the following oracles: *RequestPK*, *RequestSK*, *Sign* and *ReSign*.

Output: Finally, Φ outputs $(m^*, pk_{i^*}, \sigma_{i^*})$. Φ wins the game if the following conditions hold:

(a) the oracles $RequestSK(i^*)$, $Sign(m^*, pk_{i^*})$ and $ReSign(m^*, -, pk_{i^*}, -)$ have never been queried, note that "–" stands for all public keys and all signatures;

(b) $Verify(m^*, pk_{i^*}, \sigma_{i^*})$ returns 1.

Let $SuccE_{\Phi}^{\Pi}$ be the success probability that Φ wins the above security game.

Bidirectional Limited-Proxy Security Game: X and Φ perform the game as follows.

Initialization: X uses the *Setup* to produce param and sends param to Φ.

Queries: Φ asks X the following oracles: *RequestPK*, *RequestReKey* and *Sign*.

Output: Φ returns $(m^*, pk_{i^*}, \sigma_{i^*})$. Φ successes if the two conditions below hold:

(a) the oracle $Sign(m^*, "-")$ has never been asked;

(b) $Verify(m^*, pk_{i^*}, \sigma_{i^*})$ returns 1.

Let $SuccP_{\Phi}^{\Pi}$ be the probability that Φ successes.

Definition 1 *We say that a private bidirectional* PRS *scheme* Π *is existentially unforgeable against polynomially bounded forger* Φ *if the success probabilities* $SuccE_{\Phi}^{\Pi}(\lambda)$ *and* $SuccP_{\Phi}^{\Pi}(\lambda)$ *of* Φ *are negligible.*

3 Preliminaries

The PS signature scheme [Pointcheval and Sanders, 2016] at CT-RSA'06 is recalled as follows.

PS signature scheme [Pointcheval and Sanders, 2018] has four algorithms.

- *Setup*: receives λ as input, returns the global parameters param = $(p, \Gamma, \tilde{\Gamma}, \Gamma_T, g, \tilde{g}, e)$.

- KeyGen: receives λ as input, returns secret key $(x, y) \in Z_p^*$, and public key $(\tilde{h} \in \tilde{\Gamma}, \tilde{X} = \tilde{h}^x, \tilde{Y} = \tilde{h}^y)$.
- Sign: receives $m \in Z_p$ and secret key (x, y) as inputs. It randomly chooses $h \xleftarrow{\$} \Gamma$, $h \neq 1_\Gamma$, and returns the signature $\sigma = (\sigma_1 = h, \sigma_2 = h^{(x+ym)})$. message $m \in Z_p$
- Verify: receives $\sigma = (\sigma_1, \sigma_2)$, m , and $(\tilde{h}, \tilde{X}, \tilde{Y})$ as inputs. It returns 1 (the signature is valid) if and only if two conditions below are satisfied:

$$\sigma_1 \neq 1_\Gamma;$$
$$e(\sigma_1, \tilde{X}\tilde{Y}^m) = e(\sigma_2, \tilde{h}).$$

This signature scheme is secured under the PS assumption 2 or q-SDH assumption. These two assumptions are defined as below.

Definition 2 ***PS Assumption 2:*** *Assume* $(p, \Gamma, \tilde{\Gamma}, \Gamma_T, g, \tilde{g}, e)$ *be a bilinear group setting of type 3. Pick* $u, v \xleftarrow{\$} Z_p$ *, define the oracle* $O(m)$ *on input* $m \in Z_p$ *that chooses a random* $h \in \Gamma$ *and returns the pair* (h, h^{u+mv}). *Given* $(g$, $\tilde{g}$, $\tilde{g}^u$, $\tilde{g}^v)$ *and unlimited access to this oracle, it is hard to generate a pair* (h, h^{u+m^*v}), *with* $h \neq 1_\Gamma$, *for a new scalar* m^* *not asked to* O .

Definition 3 q ***-SDH Assumption:*** *Assume* $(p, \Gamma, \tilde{\Gamma}, \Gamma_T, e)$ *be a bilinear group setting of type 3, with g and* $\tilde{g}$ *are two generator of* Γ *and* $\tilde{\Gamma}$. *Picks* $s \xleftarrow{\$} Z_p$, *given* $(g, \tilde{g}, g^s, g^{s^2}, \cdots, g^{s^q}, \tilde{g}^s)$, *it is hard to generate a pair* $(c, g^{\frac{1}{s+c}}) \in Z_p \times \Gamma$.

4 Our Construction

In this section, as a warm up we first present the idea of our scheme, and then we describe the details of our construction. The rest of the section is devoted for the security analysis.

Intuition

Intuitively, to transform from $(h, h^{x_i + y_i.m})$ to $(h', h'^{x_j + y_j.m})$ one should have the value $\frac{y_j}{y_i}$ in hands to eliminate y_i. However, when we compute $h^{(x_i + y_i.m).\frac{y_j}{y_i}}$ it will appear the redundant value $\frac{x_i.y_j}{y_i}$ while still lacking the value x_j. Therefore, one should have the additional value $\frac{x_i.y_j}{y_i} - x_j$ to

eliminate the redundant value $\frac{x_i.y_j}{y_i}$ and to provide the lacking value x_j. Overall, to transform from $(h, h^{x_i+y_i.m})$ to $(h', h'^{x_j+y_j.m})$, one should have both the values $\frac{y_j}{y_i}$ and $\frac{x_i.y_j}{y_i} - x_j$, or the re-signature key should be $R_{i\leftrightarrow j} = (\frac{y_j}{y_i}, \frac{x_i.y_j}{y_i} - x_j)$. In the construction, we also show that from $R_{i\leftrightarrow j}$, one can easily compute $R_{j\leftrightarrow i}$. This leads to the fact that our scheme is a bidirectional PRS scheme.

Regarding the security, fortunately, from $R_{i\leftrightarrow j}$, one cannot derive the values x_i, x_j, y_i, y_j since one just has two equations but with four variables. In fact, in the proof we show that in case the adversary knows the re-signature keys our scheme is still secure.

The construction looks simple, however the challenging task is in the proof when the challenger needs to answer the oracle queries from adversary. Especially, when the challenger doesn't know the secret keys of users i and j (that means she doesn't know the values x_i, x_j, y_i, y_j) but still has to manage to answer $R_{i\leftrightarrow j}$. With some tricks in the proof, we can deal with this challenge.

Construction

Our bidirectional PRS scheme is constructed as follows.

Setup(1^λ): receives λ as input, sends back $(p, \Gamma, \tilde{\Gamma}, \Gamma_T, g, \tilde{g}, e)$ as output.

KeyGen: returns $(x, y) \in Z_p^*$ as private key, and $(\tilde{X}, \tilde{Y})$ where $\tilde{X} = \tilde{h}^x$ and $\tilde{Y} = \tilde{h}^y$ as the public key pk where $\tilde{h} \in \tilde{\Gamma}$ is a random element.

ReKeyGen: it takes as input user i's secret key (x_i, y_i) and user j's secret key (x_j, y_j), outputs the re-signature key $R_{i\leftrightarrow j} = (\frac{y_j}{y_i}, \frac{x_i.y_j}{y_i} - x_j)$. Note that from $R_{i\leftrightarrow j}$ we can compute $R_{j\leftrightarrow i}$ as follows.

$$R_{j\leftrightarrow i} = \left((\frac{y_j}{y_i})^{-1}, -1.(\frac{x_i.y_j}{y_i} - x_j).(\frac{y_j}{y_i})^{-1} \right)$$

$$= (\frac{y_i}{y_j}, \frac{x_j.y_i}{y_j} - x_i)$$

Sign : receives private key (x, y) and $m \in Z_p$. It chooses $h \xleftarrow{\$} \Gamma, h \neq 1_\Gamma$, and returns $\sigma = (\sigma_1, \sigma_2)$, $\sigma_1 = h$ and $\sigma_2 = h^{(x+ym)}$.

ReSign: receives $R_{i\leftrightarrow j}$, m, pk_i and $\sigma_i = (h, h^{(x_i+y_i.m)})$ of user i, the algorithm first checks the correctness of σ_i. If σ_i is not correct, it outputs $\perp$; otherwise, it chooses $t \xleftarrow{\$} Z_p^*$ computes $h' = h^t \neq 1_\Gamma$ (in case $h' = 1_\Gamma$, chooses another t) and $\sigma_j =$

$$\left(h', \frac{(h')^{(x_i+y_i.m).\frac{y_j}{y_i}}}{h'^{\frac{x_i.y_j}{y_i}-x_j}}\right) = \left(h', \frac{h'^{\frac{x_i.y_j}{y_i}+y_j.m}}{h'^{\frac{x_i.y_j}{y_i}-x_j}}\right)$$
$$= (h', h'^{(x_j+y_j.m)})$$

It is obvious that σ_j is a correct signature. Note that the algorithm knows $R_{i\leftrightarrow j} = (\frac{y_j}{y_i}, \frac{x_i.y_j}{y_i} - x_j)$ to compute σ_j.

Verify: receives m, $\sigma = (\sigma_1, \sigma_2)$, and $pk = (\tilde{h}, \tilde{X}, \tilde{Y})$, σ is a correct signature if and only if the two conditions below are satisfied:

$$\sigma_1 \neq 1_\Gamma;$$
$$e(\sigma_1, \tilde{X}\tilde{Y}^m) = e(\sigma_2, \tilde{h}).$$

Correctness

We can easily show that if $\sigma = (\sigma_1 = h, \sigma_2 = h^{(x_i+y_i.m)})$ then

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

Properties

- Since from $R_{i\leftrightarrow j}$ one can compute $R_{j\leftrightarrow i}$, our scheme is bidirectional PRS scheme.
- Since we cannot publish $R_{i\leftrightarrow j}$, and from $\sigma_i = (h, h^{(x_i+y_i.m)}), \sigma_j = (h', h'^{(x_j+y_j.m)})$ one cannot derive $R_{i\leftrightarrow j}$, our scheme is private proxy PRS scheme.
- Since $\sigma_j = (h', h'^{(x_j+y_j.m)})$ can be used to transform again, our scheme is multi-use PRS scheme.
- Since one cannot distinguish between a signature outputted by the *Sign* algorithm or the *ReSign* algorithm (since both h and h' are random elements), our scheme is transparent PRS scheme.
- Since users in our scheme just need to store a constant number of elements to sign and verify, our scheme is key optimality PRS scheme.

Efficiency

Regarding efficiency, our scheme has the same efficiency as that of PS signature scheme [Pointcheval and Sanders, 2016], which can be comparable to the state of the art of not only the existing PRS schemes but also the existing standard signature schemes. More precisely, regarding the storage, as shown in Figure 0 our scheme has a nice property that the public key of our scheme is very short. To verify a signature, one just needs to store 4 elements in group $\tilde{\Gamma}$ and one element in group Γ.

If we decide to use NIST's figures [Barker et~al., 2005], then with 80 bits and 128 bits of security, the size of each element in Γ is approximately 160 bits and 256 bits and in $\tilde{\Gamma}$ is approximately 1024 bits and 3072 bits. That means the public key in our scheme is approximately 1664 bits and 4096 bits. Similarly, the signature size, re-signing key size and the secret key size of our scheme are also very short.

Regarding the computational complexity, to sign a message a signer needs to compute one exponentiation in group Γ, one multiplication and one addition in Z_p^*. To check the correctness of a signature, one needs to compute two parings, one exponentiation and one multiplication in group $\tilde{\Gamma}$. For re-signing, the proxy needs to compute three exponentiations in group Γ. To conclude, storage as well as computational complexity of our scheme fits for lightweight device.

Security

The following two theorems show that our scheme is secure. More precisely, we show that in the both security games External Security and Bidirectional Limited-Proxy Security the $SuccE_{\Phi}^{\Pi}$ and $SuccP_{\Phi}^{\Pi}$ are negligible.

Theorem 4 *The* $SuccE_{\Phi}^{\Pi}$ *is negligible under the PS assumption 2.*

Proof. As defined in the security model, forger Φ is the adversary of our bidirectional PRS scheme, and X is the adversary of PS assumption 2. The following proof shows that X can relies on Φ to break the security of the PS assumption 2.

To this aim, X gets an instance of the PS assumption 2: $(p, \Gamma, \tilde{\Gamma}, \Gamma_T, g, \tilde{g}, e, \tilde{g}^u, \tilde{g}^v)$. We recall that $O(m)$, on input $m \in Z_p$, outputs the pair $(h, h^{x+y.m})$ where h is a random element in Γ.

X first gives $\text{param} = (p, \Gamma, \tilde{\Gamma}, \Gamma_T, g, \tilde{g}, e)$ to Φ as required.

We assume that in the External Security Game, Φ asks queries on users from 1 to q. X pick a random index $i^* \in [1, q]$ where she guesses that user i^* is the targeted user. We note that the proof technique where the challenger X guesses the targeted user at the beginning of the game also considered in other signature schemes [Boneh and Boyen, 2004, Shao et~al., 2011, Yang et~al., 2011, Yang et~al., 2015] to name a few.

X now responds the required oracles as below.

$RequestPK(i)$ or $RequestSK(i)$. It depends on the requested user, we have two cases. If $i \neq i^*$, X first chooses randomly $x_i, y_i \in Z_p^*, \tilde{h} \neq 1_{\tilde{\Gamma}} \in \tilde{\Gamma}$ then computes $pk_i = (\tilde{h}, \tilde{X} = \tilde{h}_i^x, \tilde{Y} = \tilde{h}_i^y)$. X finally gives pk_i or $sk_i = (x_i, y_i)$ to Φ.

If $i = i^*$, X chooses randomly $t \in Z_p^*$ and sets $pk_i = (\tilde{h} = \tilde{g}^t, \tilde{X} = \tilde{g}^{t.u}, \tilde{Y} = \tilde{g}^{t.v})$. X finally gives pk_i to Φ. In case of requesting $RequestSK(i)$, X returns FAIL and stops.

$Sign(m, pk_i)$. It depends on the requested user, we have two cases.

- If $i \neq i^*$, X simply uses her knowledge of sk_i to run $Sign$ algorithm as usual and then gives the result to Φ.

- If $i = i^*$, X simply queries the oracle O on the message m to get a signature $(h, h^{u+m.v})$, then gives the result to Φ.

$ReSign(m, pk_i, pk_j, \sigma_i)$. It depends on the requested users, we have two cases.

- If $j \neq i^*$, X simply uses her knowledge of sk_j to run $Sign$ algorithm as usual and then gives the result to Φ. Note that if sk_j doesn't exist, X first runs the $KeyGen(j)$ algorithm to get sk_j.

- If $j = i^*$, X simply queries the oracle O on the message m to get a signature $(h, h^{u+m.v})$, then gives the result to Φ.

Finally, Φ with non negligible probability successfully returns $(m^*, pk_\ell, \sigma_\ell)$. If $\ell \neq i^*$, X returns FAIL and stops. Otherwise, σ_ℓ should be in the form $(h, h^{u+v.m^*})$ and m^* has not asked before. Therefore σ_ℓ is a correct pair breaking the security of the PS assumption 2, which concludes our proof.

The next theorem shows that $SuccP_\Phi^\Pi$ is negligible under the PS assumption 2.

Theorem 5 *The $SuccP_\Phi^\Pi$ is negligible under the PS assumption 2.*

Proof. As defined in the security model, forger Φ is the adversary of our bidirectional PRS scheme, and X is the adversary of PS assumption 2. The following proof shows that X can relies on Φ to break the security of the PS assumption 2.

X first uses the instance of the PS assumption 2 to send $\mathrm{param} = (p, \Gamma, \tilde{\Gamma}, \Gamma_T, g, \tilde{g}, e)$ to Φ.

To correctly simulate the Bidirectional Limited-Proxy Security Game for Φ, assume that that Φ asks queries on users from 1 to q. X picks a random index $i^* \in [1, q]$ where she guesses that user i^* is the targeted user. X also creates an empty list $\mathrm{P}\Lambda$ to store the re-signature keys. X now simulates the required oracles as below.

$RequestPK(i)$. It depends on the requested user, we have two cases. If $i \neq i^*$, X picks $k, k' \xleftarrow{\$} Z_p^*$ and implicitly sets $R_{i \leftrightarrow i^*} = (k, k') = (\frac{v}{y_i}, \frac{x_i.v}{y_i} - u)$, where (x_i, y_i) is the unknown private key of user i. X next computes the public key pk_i for user i as follows.

- chooses randomly $r \xleftarrow{\$} Z_p^*$, computes $\tilde{h} = \tilde{g}^r$.

- computes $\tilde{h}^{x_i} = (\tilde{g}^{k'}.\tilde{g}^u)^{r.k^{-1}} = (\tilde{g}^{\frac{x_i.v}{y_i}-u}.\tilde{g}^u)^{r.\frac{y_i}{v}} = \tilde{g}^{r.x_i}$, note that X knows $\tilde{g}^u$ from the assumption.

- computes $\tilde{h}^{y_i} = \tilde{g}^{v.r.k^{-1}} = \tilde{g}^{v.r.\frac{y_i}{v}} = \tilde{g}^{r.y_i}$.

X finally gives pk_i to Φ, adds $i, i^*, R_{i \leftrightarrow i^*}$ to the $\mathrm{P}\Lambda$ list. Note that from $R_{i \leftrightarrow i^*}$ we can easily to compute $R_{i^* \leftrightarrow i}$, so we do not need to add $i^*, i, R_{i^* \leftrightarrow i}$ to the $\mathrm{P}\Lambda$ list.

If $i = i^*$, X chooses randomly $t \xleftarrow{\$} Z_p^*$ and sets $pk_i = (\tilde{h} = \tilde{g}^t, \tilde{X} = \tilde{g}^{t.u}, \tilde{Y} = \tilde{g}^{t.v})$. X finally gives pk_i to Φ.

$RequestReKey(i, j)$. It depends on the requested user, we have two cases.

- If $i, j \neq i^*$, X first check the $\mathrm{P}\Lambda$ list to get $R_{i \leftrightarrow i^*} = (\frac{v}{y_i}, \frac{x_i.v}{y_i} - u)$ and $R_{j \leftrightarrow i^*} = (\frac{v}{y_j}, \frac{x_j.v}{y_j} - u)$. Note that if $R_{i \leftrightarrow i^*}, R_{j \leftrightarrow i^*}$ do not exist, X will first query the $RequestPK(i)$ and $RequestPK(j)$ oracles (note that X queries the oracle means X runs the algorithm of the oracle). To compute $R_{i \leftrightarrow j} = (a, b)$, she works as below.

$$a = \frac{v}{y_i}.(\frac{v}{y_j})^{-1} = \frac{y_j}{y_i}$$

$$b = (\frac{x_i.v}{y_i} - u - \frac{x_j.v}{y_j} + u).(\frac{v}{y_j})^{-1} = \frac{x_i.y_j}{y_i} - x_j$$

X finally adds $i, j, R_{i \leftrightarrow j}$ to the $\mathrm{P}\Lambda$ list and gives $R_{i \leftrightarrow j}$ to Φ.

- If $j = i^*$ (or $i = i^*$, for simplicity we suppose that $j = i^*$), X first check the $\mathrm{P}\Lambda$ list to get $R_{i \leftrightarrow i^*} = (\frac{v}{y_i}, \frac{x_i.v}{y_i} - u)$, then gives $R_{i \leftrightarrow i^*}$ to Φ. Note that if $R_{i \leftrightarrow i^*}$ does not exist, X will first query the $RequestPK(i)$ oracle.

$Sign(m, pk_i)$. It depends on the requested user, we have two cases.

- If $i \neq i^*$, X first queries O on the message m to receive a signature $(h, h^{u+m.v})$. Next, X uses the re-signature key $R_{i^* \leftrightarrow i}$ to run the $ReSign$ algorithm to get σ_i, then gives σ_i to Φ. Note that if $R_{i^* \leftrightarrow i}$ does not exist, X will first query the $RequestPK(i)$ oracle to get $R_{i \leftrightarrow i^*}$ then uses $R_{i \leftrightarrow i^*}$ to compute $R_{i^* \leftrightarrow i}$.

- If $i = i^*$, X simply asks O on the message m to get a signature $(h, h^{u+m.v})$, then gives the result to Φ.

Finally, Φ with non-negligible probability successfully returns $(m^*, pk_\ell, \sigma_\ell)$. If $\ell \neq i^*$, X returns FAIL and stops. Otherwise, σ_ℓ should be in the form $(h, h^{u+v.m^*})$ and m^* has not asked before. Therefore σ_ℓ is a correct pair to break the security of the PS assumption 2, which concludes our proof.

5 Conclusion

Proxy re-signature scheme is an important primitive with many practical applications such as distributed storage, distributed rights management, cloud infrastructure, etc. However, to our knowledge, all of the PRS schemes without using RO need to use the Waters' hash function, this leads to PRS schemes with large key size and inefficient computing time. In this paper, we propose a new bidirectional PRS scheme without using RO and without using the Waters' hash function. Our scheme therefore can overcome some weaknesses of a Waters' hash function-based PRS scheme.

Acknowledgments

This research is funded by "Nghien cuu ung dung cong nghe chuoi khoi (Blockchain) xay dung phan mem truy xuat nguon goc nong san thuc pham tinh Thanh Hoa" project.

References

[1] Asaar, M.R., Salmasizadeh, M., & Susilo, W. (2015). An identity-based multi-proxy multi-signature scheme without bilinear pairings and its variants. *The Computer Journal*, *58*(4), 1021-1039.

[2] Ateniese, G., & Hohenberger, S. (2005). Proxy re-signatures: new definitions, algorithms, and applications. *In Proceedings of the 12th ACM conference on Computer and communications security*, 310-319.

[3] Barker, E., Barker, W., Burr, W., Polk, W., & Smid, M. (2012). Nist special publication 800-57: Recommendation for key management–part 1: General (revision 3). *US Department of Commerce-National Institute of Standards and Technology*.

[4] Blaze, M., Bleumer, G., & Strauss, M. (1998). Divertible protocols and atomic proxy cryptography. *In Advances in Cryptology—Eurocrypt'98: International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, May 31 Proceedings 17*, 127-144. Springer Berlin Heidelberg.

[5] Boneh, D., & Boyen, X. (2004). Short signatures without random oracles. In *Advances in Cryptology-Eurocrypt 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6. Proceedings 23*, 56-73. Springer Berlin Heidelberg.

[6] Boneh, D., & Boyen, X. (2008). Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of cryptology*, *21*(2), 149-177.

[7] Chaudhari, S., Aparna, R., & Rane, A. (2021). A Survey on Proxy Re-Signature Schemes for Translating One Type of Signature to Another. *Cybernetics and Information Technologies*, *21*(3), 24-49.

[8] Chaum, D. (1983). Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto 82*, 199-203. Springer US.

[9] Chow, S.S., & Phan, R.C.W. (2008). Proxy re-signatures in the standard model. In *Information Security: 11th International Conference, ISC 2008, Taipei, Taiwan. Proceedings 11*, 260-276. Springer Berlin Heidelberg.

[10] Hong, X., & Long, Y. (2012). A Novel Unidirectional Proxy Re-Signature Scheme and Its Application for MANETs. *Journal of Computational*, *7*(7), 1796-1800.

[11] Hu, X., Zhang, Z., & Yang, Y. (2009). Identity based proxy re-signature schemes without random oracle. *In IEEE International conference on computational intelligence and security*, *2*, 256-259.

[12] Lee, E., & Kim, S.W. (2018). Non-interactive conditional proxy re-signature in the standard model. *The Computer Journal*, *61*(12), 1772-1782.

[13] Lei, Y., Jia, Z., Wang, L., Gong, B., Cheng, Y., & Fu, J. (2020). Partial Blind Proxy Re-signature Scheme for Mobile Internet. In *Trusted Computing and Information Security: 13th Chinese Conference, CTCIS 2019, Shanghai, China, October 24–27, 2019, Revised Selected Papers 13*, 16-30. Springer Singapore.

[14] Libert, B., & Vergnaud, D. (2008). Multi-use unidirectional proxy re-signatures. In *Proceedings of the 15th ACM conference on Computer and communications security*, 511-520.

[15] Lv, G., Lei, Y., Hu, M., Cheng, Y., Gong, B., & Fu, J. (2020). Provably Secure Server-Assisted Verification Threshold Proxy Re-signature Scheme. *In Trusted Computing and Information Security: 13th Chinese Conference, CTCIS 2019, Shanghai, China, October 24–27, 2019, Revised Selected Papers 13*, 44-57. Springer Singapore.

[16] Mambo, M., Usuda, K., & Okamoto, E. (1996). Proxy signatures for delegating signing operation. *In Proceedings of the 3rd ACM Conference on Computer and Communications Security*, 48-57.

[17] Matsuo, T. (2007). Proxy re-encryption systems for identity-based encryption. In *Pairing-Based Cryptography–Pairing 2007: First International Conference, Tokyo, Japan, July 2-4, 2007. Proceedings 1*, 247-267. Springer Berlin Heidelberg.

[18] Menon, T. (2012). An identity based proxy re-signature scheme. *International Journal of Engineering and Technology*, *4*(3), 303-306.

[19] Pointcheval, D., & Sanders, O. (2016). Short randomizable signatures. In *Topics in Cryptology-CT-RSA 2016: The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29-March 4, 2016, Proceedings*, 111-126. Springer International Publishing.

[20] Pointcheval, D., & Sanders, O. (2018). Reassessing security of randomizable signatures. *In Topics in Cryptology–CT-RSA 2018: The Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, Proceedings*, 319-338. Cham: Springer International Publishing.

[21] Shao, J., Cao, Z., Wang, L., & Liang, X. (2007). Proxy re-signature schemes without random oracles. In *Progress in Cryptology–INDOCRYPT 2007: 8th International Conference on Cryptology in India, Chennai, India. Proceedings 8*, 197-209. Springer Berlin Heidelberg.

[22] Shao, J., Wei, G., Ling, Y., & Xie, M. (2011). Unidirectional identity-based proxy re-signature. *In IEEE International Conference on Communications (ICC)*, 1-5.

[23] Taban, G., Cárdenas, A.A., & Gligor, V.D. (2006). Towards a secure and interoperable DRM architecture. In *Proceedings of the ACM workshop on Digital rights management*, 69-78.

[24] Talegaon, S., & Krishnan, R. (2020). Administrative models for role-based access control in android. *Journal of internet services and information security*, *10*(3), 31-46.

[25] Waters, B. (2005). Efficient identity-based encryption without random oracles. In *Advances in Cryptology–Eurocrypt: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark. Proceedings 24*, 114-127. Springer Berlin Heidelberg.

[26] Wu, Y., Xiong, H., & Jin, C. (2020). A multi-use unidirectional certificateless proxy re-signature scheme. *Telecommunication Systems*, *73*, 455-467.

[27] Xiong, H., Kang, Z., Chen, J., Tao, J., Yuan, C., & Kumari, S. (2020). A novel multiserver authentication scheme using proxy resignature with scalability and strong user anonymity. *IEEE Systems Journal*, *15*(2), 2156-2167.
[28] Yang, P., Cao, Z., & Dong, X. (2011). Threshold proxy re-signature. *Journal of Systems Science and Complexity*, *24*(4), 816-824.
[29] 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 & Information Sciences*, *9*(6), 3251-3261.
[30] Yuqiao, D., & Ge, S. (2011). Proxy re-signature scheme based on quadratic residues. *Journal of networks*, *6*(10), 1459-1465.
[31] Zhang, J., Bai, W., & Wang, Y. (2019). Non-interactive ID-based proxy re-signature scheme for IoT based on mobile edge computing. *IEEE Access*, *7*, 37865-37875.

Authors Biography

Vinh Duc Tran is currently a lecturer in the School of Information and Communication Technology at Ha noi University of Science and Technology since 2012. He received Master degree in Computer Science from Ha noi University of Science and Technology, Ha noi, Viet Nam, and received Ph.D. degree in Computer Science from University of Nice Sophia Antipolis in 2011. His research interests are in Cryptography, specifically Public-Key Encryption, Attribute-Based Encryption, Digital Signature and Decentralized Cryptosystems.

Phi Thuong Le is currently a lecturer in the Nghi Son Vocational College, Thanh Hoa, Viet Nam. He received Master degree in Computer Science from Hong Duc University, Thanh Hoa, Viet Nam in 2022. His research interests are in Cryptography, specifically Public-Key Encryption, Attribute-Based Encryption, Blockchain and Decentralized Cryptosystems.

Viet Cuong Trinh received the Ph.D. degrees in computer science from Paris 8 University, Paris, France, in 2013. Before joining Hong Duc University, Viet Nam, in 2016, he was a postdoctoral researcher with the Orange Labs, Caen, France and then KINDI research center, Qatar University, Doha, Qatar. He is currently an Associate Professor with the Faculty of Information and Communication Technology, Hong Duc University, Viet Nam. His research is focused on cryptography, in particular on provable security for cryptographic schemes including Public-key encryption, Digital Signature, Broadcast Encryption, Blockchain, Attribute-based Encryption and Decentralized Cryptosystem.