An Efficient Lattice-Based Proxy Signature with Message Recovery

Faguo Wu, Wang Yao, Xiao Zhang(✉), and Zhiming Zheng

Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, School of Mathematics and Systems Science, Beihang University, Beijing 100191, China
09621@buaa.edu.cn

Abstract. Proxy signature scheme is an important cryptographic primitive in which an entity can delegate its signing rights to another entity, the purpose of proxy signature with message recovery is to shorten the length of proxy signatures which can effectively reduce the communication overhead. Although message recovery proxy signature scheme based on conventional number-theoretic problems has been proposed for a long time, the message recovery technique draws no attention to proxy signature scheme from lattice. In this paper, we firstly propose a proxy signature scheme with message recovery from lattice which is more efficient than previous proxy signature schemes in signature size, time and energy cost, and we prove that in the random oracle, our scheme is secure model under the hardness assumption of SIS. Our proxy signature scheme with message recovery would work well in the quantum age based on the underlying lattice problems.

Keywords: Lattice-based · Proxy signature · Message recovery
Communication overhead · Quantum age

1 Introduction

Proxy signatures, proposed by Mambo et al. [1], are significant cryptographic systems that are widely used in different situations, such as E-commerce, cloud computing and Electronic election. In a proxy signature scheme, Original signer A delegates his signing power and responsibility to another one which is called proxy signer. According to the hardness of traditional Number Theory problems, the researchers have prosed many effective proxy signature schemes such as integer factoring-based schemes, discrete logarithm schemes and elliptic curve schemes [2,3]. However, in a long run, all these proxy signature schemes are not secure, because both discrete logarithms and large prime factorization algorithms can be solved in polynomial-time [4] when we take quantum computing into account. For purpose of reducing the threat from the quantum age, many researchers pay their attention to Post-Quantum Cryptography, some proxy signature schemes, like hash-based schemes, MPKC schemes, lattice-based schemes, which are considered

G. Wang et al. (Eds.): SpaCCS 2017, LNCS 10656, pp. 321–331, 2017.
https://doi.org/10.1007/978-3-319-72389-1_26

as Post-Quantum Cryptography, and many corresponding efficient Post-Quantum proxy signature schemes have been proposed, such as [5–9].

Lattice-based signature occupies a position of particular interest, as it relies on well-studied problems and comes with uniquely strong security guarantees [10]. Organizations and research groups are looking forward for efficient lattice-based cryptography schemes to replace RSA and ECC based schemes, and many creative and constructive results show that properly optimized lattice schemes may be competitive with, or even outperform, classical factoring and discrete logarithm-based cryptography.

The first signature scheme with message recovery was proposed by Nyberg and Rueppel [11]. In a signature scheme with message recovery, we only need to transmit the signature without the signed message, because verifier can easily recover the signed message from the signature. This construction quietly adapts to situations where small signed message to be transmitted or strict bandwidth requirements [12,13]. The existent proxy signature schemes with such property can be categorized into two different types: discrete logarithm based and RSA based [14–16] which efficiently improve the performance of previous signature without message recovery. Although there are many very effective latticed-based proxy signature schemes, based on the hardness assumption of lattice, have been proposed [17–19], as far as we know, there are no efficient lattice-based proxy signature schemes with message recovery have been proposed.

In this paper, benefit from the techniques [20,21], we propose an efficient lattice-based proxy signature scheme with message recovery, and prove that it has existentially security in the random oracle model, in addition, contribution to message recovery technology and mainly multiplication of matrix and vector given in our scheme, our scheme enjoy higher performance when compared with others' proxy signature scheme, finally, when we take piratical situation into consideration [22], we are surprised to find that this kind of scheme consume less energy even when we add some operations for message recovery, which means our proxy signature schemes are extremely suitable for system with low energy and low bandwidth. As the hardness assumption of lattice problem SIS, our lattice-based message recovery proxy signature scheme would work well in the quantum age.

The remainder of our paper is organized as follow. In Sect. 2, we provide necessary preliminaries of our scheme. In Sect. 3, we describe two models of our lattice-based proxy signature scheme with message recovery:syntax model and security model. In Sect. 4, we propose our efficient message recovery proxy scheme from lattice. In Sect. 5, we present the formal security analysis of our scheme. In Sect. 6, we introduce some necessary criterions, and give detailed comparisons between our scheme and some existing proxy schemes from lattice.

2 Preliminaries

2.1 Notations

In this paper, following notations would be used:

- $x \parallel y$ denotes the connection of two string x and y, and they are effectively recoverable.

- $M^{n\times(k_1+k_2)} = M_1^{n\times k_1} \parallel M_2^{n\times k_1}$ denotes the concatenation of Matrices M_1, M_2.
- $\|v\|_p$ denotes the l_p norm of v.
- $|x|$ denotes the quantity of bits of v.
- $|x|^{l_1}$ denotes the first left l_1 bits of x.
- $|x|_{l_2}$ denotes the first right l_2 bits of x.

2.2 Lattice

Definition 1. A lattice L is a discrete subgroup of some space $\mathbb{R}^n$, it is generated by independent vector $v_1, v_2, \cdots, v_k \in \mathbb{R}^n$ through the following way:

$$\Lambda = L(v_1, v_2, \cdots, v_k) = \{\sum_{i=1}^{k} a_i v_i \mid a_i \in \mathbf{Z}\}$$

The basis of L are vectors $v_1, v_2, \cdots, v_n$, lattice's rank is the integer n where $k < n$ and a_i is coefficient.

Definition 2. Given integers q, m, n and a matrix $A \in Z_q^{n\times m}$, for some $S \in Z^m$.

$$\Lambda_q(A) = \{x \in Z^m : x = A^T S = u(modq)\}$$
$$\Lambda_q^{\perp}(A) = \{x \in Z^m : x = A^T S = 0(modq)\}$$

From the above definition, these two types of lattices are dual to each other.

2.3 Gaussian on Lattice

In lattice-based signature scheme, Gaussian series are very effective techniques which are widely used, and we have a briefly review of it here.

Definition 3 (Discrete Gaussian distribution). $\sigma \in \mathbb{R}^m$ is standard deviation, vector $c \in Z^m$ is center, Continuous Gaussian distribution $\rho_{c,\sigma}^m(x)$ and Discrete Gaussian distribution $D_{c,\sigma}^m(x)$ are defined as follow:

$$\begin{aligned} \rho_{c,\sigma}^m(x) &= (\frac{1}{\sqrt{2\pi\delta^2}})^m e^{\frac{-\|x-c\|^2}{2\sigma^2}} \\ D_{c,\sigma}^m(x) &= \frac{\rho_{c,\sigma}^m(x)}{\sum_{z\in Z^m} \rho_{c,\sigma}^m(z)} \end{aligned} \tag{1}$$

When $c = 0$, we can simply write $\rho_{c,\sigma}^m(x), D_{c,\sigma}^m(x)$ as $\rho_\sigma^m, D_\sigma^m$, and from [21], an important theorem of Discrete Gaussian distribution is described as follow

Theorem 1. $\forall\sigma > 0$ and $m \in Z^+$

(1) $\mathrm{P}[x \in D_\sigma^1 : |x| > 12\sigma] < 2^{-100}$;
(2) $\mathrm{P}[x \in D_\sigma^m : \|x\| > 2\sigma\sqrt{m}] < 2^{-m}$.

Algorithm 1. Rejection sampling technique

Input: $H : \{\{0,1\}^* \rightarrow v : v \in -1, 0, 1^k, \| v \| < c\}$(Where $k \in Z$ and $\ll$ m), message u, a matrix A randomly sampled from $Z_q^{m \times n}$, S(signature key) sampled from $\{-d, \cdots, 0, \cdots, d\}^{m \times k_1}$

Output: Vector v and c

1: Obtain y randomly from D_σ^m
2: $C = H(Ay, u)$
3: $Z = SC + y$
4: return (Z, C) with probability $min(\frac{D_\sigma^m(Z)}{MD_{SC,\sigma}(Z)}, 1)$

2.4 RST: Rejection Sampling Technique

For a lattice-based signature scheme, the most important conception of the RST is to eliminate the relationship between signing key and output signature's distribution [10], the algorithm as follow (Algorithm 1).

2.5 Small Integer Solution (SIS) and its Hardness Assumption

Definition 4. For an integer modular homogeneous scheme $As = 0 mod q$, get a proper solution $s \in Z^m$ where q, matrix coefficient, small solution s satisfy $q \in Z^m$, $A \in Z_q^{n \times m}$ and $\| s \| \leq \beta$ where β is a real value.

In reference [10,23], they proved that for any polynomial-bound m, β and any prime p, with small factors and the Gaussian measure, there is no difference between the hardness of some worst-case approximation and average-case harness of SIS. Even the hardness of SIS problem has been proved, there still exist overwhelming probability that anyone can solve some case if the trapdoor of $f = Ax(mod q)$ is got.

3 Lattice-Based Proxy Signature Scheme with Message Recovery

Syntax model and security model of our lattice-based proxy signature scheme with message recovery are proposed in this section.

3.1 Syntax

Definition 5. In such lattice-based proxy signature scheme with message recovery scheme, there are three participants: An original signer with ID_o, a proxy signer: ID_p, and a verifier, and this scheme is consists of six PPT algorithm (Setup, KeyGen, DelGen, DelVer, Psign, Pver), where:

1. Setup: Given a security parameters n, and select appropriate parameters and functions. $(par, n) \leftarrow Setup$.

2. KeyGen: Given a security parameters n, this algorithm output the secret and public key of original signer and proxy signer: Original signer's = (PK_o, SK_o), Proxy signer' = (PK_p, SK_p).$(PK_o, SK_o, PK_p, SK_p) \leftarrow KeyGen(n, M)$.
3. DelGen: Given the original signer's public key PK_o and secret key SK_o hash function H_i, proxy signer's ID_p, this algorithm output the Delegation Key (PK_D, SK_D) for original signer, original signer sends this pair to proxy signer in security channel. $(PK_D, SK_D) \leftarrow DelGen(H_i, ID_p, PK_D, SK_D)$.
4. DelVer: Given the original signer's public key PK_o and Delegation Key $(PK_D$,$SK_D)$, and check $DelVer(PK_o, PK_D, SK_D) = 1$ or not. If it outputs 1, it's a valid delegation of original signer. $\{0, 1\} \leftarrow DelVer(PK_o, PK_D, SK_D)$.
5. Psign: Given the secret delegation key SK_D, proxy signer's secret key SK_p and the message $u = u_1 \parallel u_2$, output the proxy signatureθ, that is $\theta \leftarrow Psign(u, SK_D, SK_p)$.
6. Pver: Given the public key PK_o of original signer, public key PK_o of proxy signer, public delegation key PK_D, proxy signer's ID_p, hash function H_i, partial message u_2, and proxy signature θ, if the proxy signature is valid, output 1, otherwise, output 0, that is $(m, \{0, 1\}) \leftarrow Pver(PK_o, PK_o, PK_D, ID_p, H_i, \theta, u_2)$.

Remark 1. For consistency requirements, partial message u_2, the proxy signature θ of secret Delegation Key SK_D and proxy signer's secret key SK_P must hold with overwhelming probability with following equation Verify $(Sign(SK_D, SK_P, u), u_2, PK_D, SK_D) = 1$.

3.2 Security Model

In a lattice-based proxy signature scheme with message recovery, the properties of Unforgeability, Verifiability, Strong identifiability, Strong undeniability and Key dependence are satisfied naturally. Therefore, we consider in this lattice-based proxy signature scheme under adaptive chosen message and identity attack. To have a formal security definition for this scheme, the security model is an security game played between a adversary A and a challenger C:

1. Setup: In this game, the challenger C firstly run the algorithm Setup(n), get the necessary parameters and send them to the adversary A.
2. Queries: In such query-game, following types of queries can be adaptively issued by adversary A within polynomial bound number of questions.
   - KeyGen-query: The adversary A can issue a query on the ID which he want to get the secret key, and the challenger run the algorithm KeyGen, and return A with SK_{ID} in response.
   - DelGen-query: To get the delegation key SK_D, the adversary A input two secret key corresponding to the identity ID_o and IDp, in response, the challenger C run the algorithm DelGen, and return A with SK_D.
   - Psign-query: When adversary issues such on ID_p with message u, the challenger C run the algorithm Psign, and return A with signature.

3. By the above queries, the adversary A generate a valid proxy signature θ' on message u^* under the identity ID', if the following holds, Adversary A wins the game: (i) $Vfy(Sign(SK_D, SK_P, m), PK_D, SK_D, par) = 1$; (ii) u^* has never been send to the Psign-query; (iii) all identities which is related to ID' have never been sent to KeyGen-query.

An lattice-based proxy signature scheme with message recovery is considered as existential unforgeable if the advantage of Adversary A wins the above query game in polynomial time is negligible.

4 Our Lattice-Based Proxy Signature Scheme with Message Recovery

We gave a detailed account of our efficient message recovery proxy signature scheme from lattice in this section. Like the traditional signature systems, our scheme have three participants: an original signer with ID_o, a proxy signer with identity ID_p, and a verifier, and this scheme is consists of six probabilistic polynomial time (PPT) algorithm (Setup, KeyGen, DelGen, DelVer, Psign, Pver), where:

1. Setup: Given the security parameter n of this system, we select l_1, l_2, k_1, k_2, m, $q \in N$, where q is a prime, select five hash function: $H_1 = Z_q^n \to \{0,1\}^{l_1+l_2}$, $H_2 = \{0,1\}^* \to \{0,1\}^{k_1+k_2}$, $H_3 = ID \to \{-1,0,1\}^{k_1 \times k_2}$, $F_1 = \{0,1\}^{l_2} \to \{0,1\}^{l_1}$, $F_2 = \{0,1\}^{l_1} \to \{0,1\}^{l_2}$. $H_1, H_2$are seen as a random oracle.
2. KeyGen: On input security parameter n, randomly choose $A \in Z_q^{n \times m}$ together with two secret matrix basis $S_1, S_2 \in \{-d, \cdots, 0, \cdots, d\}^{m \times k_1}$. Original signer computes $T_1 = AS_1 mod q$, and keep $PK_o = (A, T_1)$ as his own public key, $SK_o = S_1$ as secret key. Proxy signer computes $T_2 = AS_2 mod q$, and keep $PK_p = (A, T_2)$ as his own public key, $SK_p = S_2$ as secret key.
3. DelGen: Original signer computes $t = H_3(ID_p)$, $S_3 = S_1 t \in Z^{m \times k_2}$, $T_3 = T_1 t \in Z^{n \times k_2}$, and send (S_3, T_3) to Proxy signer via an safe authenticated channel, Proxy signer keep the delegation key $PK_D = S_3$, proxy signer can use it to generate valid proxy signatures stand for original signer, and the corresponding public key is $PK_D = T_3$.
4. DelVer: The Proxy signer receives (S_3, T_3) from original signer, and checks if $AS_3 = T_1 t$ holds, where $t = H_3(ID_p)$.
5. Psign: The Proxy signer with identity ID_p do the following:
   - Divide the message u into two parts $u = u_1 \parallel u_2$ where $\mid u_1 \mid = l_2$, if$\mid u \mid < l_2$, let $u_2 = \perp$.
   - Select a random $y \in D_\sigma^m$, and compute $\alpha = H_1(Ay)$.
   - Let $u_1' = F_1(u_1) \parallel (F_2(F_1(u_1)) \bigoplus u_1)$, $r = \alpha \bigoplus u_1'$.
   - Compute $C = H_2(r, u_2)$.
   - Compute $Z = (S_2 \| S_3)C + y \in Z_q^m$.
   - The proxy signature on the message u is (r, Z, u_2) with probability $min(\frac{D_\sigma^m(Z)}{MD_{(S_2 \| S_3)C,\sigma}(Z)}, 1)$.

6. Pver: Given (r, Z, u_2), a verifier does as explained in the succeeding text:
   - Compute $\alpha = H_1(AZ - (T_2 \parallel T_3)H_2(r, u_2))$.
   - Compute $u_1^{'} = r \bigoplus \sigma$, $u_1 = |u_1^{'}|_{l_2} \bigoplus F_2(|u_1^{'}|^{l_1})$, and then recover message $u = u_1 \parallel u_2$.
   - Check if $\|Z\| < 2\sigma\sqrt{m}$ and $F_1(u_1) = |u_1^{'}|^{l_1}$ hold at same time, if so, accept signature from the proxy signer, otherwise, reject it.

Theorem 2. *Our lattice-based proxy signature scheme with message recovery is correctness.*

Proof. From the above detailed construction, we can easily have following equations where u message

$$\begin{aligned} & AZ - (T_2 \parallel T_3)H_2(r, u_2) \\ & = A((S_2 \| S_3)C + y) - (T_2 \parallel T_3)H_2(r, u_2) \\ & = Ay \end{aligned}$$

the distribution of $(S_2 \| S_3)C + y$ is statically closed to the distribution D_σ^m, and from the Theorem 1, $\parallel (S_2 \| S_3)C + y \parallel \leq 2\sigma\sqrt{m}$ with probability at least $1 - 2^{-m}$. On the other hand, $u_1^{'} = F_1(u_1) \parallel (F_2(F_1(u_1)) \bigoplus u_1)$, we can recover $u_1 = |u_1^{'}|_{l_2} \bigoplus F_2(|u_1^{'}|^{l_1})$ with $F_1(u_1) = |u_1^{'}|^{l_1}$ hold.

5 Security Analysis

As a signature scheme, security is the most important factor, we prove that our lattice-based proxy signature scheme with message recovery is secure enough (unforceability) under the hardness assumption of SIS in this section.

When we proving the unforceability of proxy signature scheme with message recovery, we should take two types adversary into consideration:

Type(i) Adversary A can not only have the public key PK_o of original signer and public key PK_p of proxy signer, but also have the original signer's secret key SK_o.

Type(ii) Adversary A has neither the original signer's secret key SK_O nor the proxy signer's secret key SK_P.

It's obvious that adversary in **Type**(i) knows more information than the adversary in **Type**(ii), so we only need to take **Type**(i) adversary into consideration. We suppose there is a polynomial time, adversary A forge a valid proxy signature by at most q_{H_1}times H_1 query, q_{H_2}times H_2 query, q_{F_1}times F_1 query, q_{F_2}times F_2 query, and q_s times signature query with non-negligible probability, it means there exist an algorithm C (Challenger C) which can solve a $SIS_{q,n,m,C}$ problem "with the help of" Adversary A in polynomial time. Algorithm C (Challenger C) has a game with A, the following is the simulation. **Queries**. The adversary A issues the following types of queries adaptively, A has random oracle $H_1 - query$ before any other queries.

- $H_3 - query$. Challenger C maintains a list $L_0 = (ID_{p_i}, PK_{p_i}, SK_{p_i})$, and the initial value is null, when adversary A issues a query on ID_{p_i}, Challenger C search it in list first, if there exist corresponding tuple $(ID_{p_i}, PK_{p_i}, SK_{p_i})$, return PK_{p_i}; otherwise, Challenger C randomly chooses matrix $S \in Z_q^{m \times k_2}$, then Challenger C computes $PK = AS$, Update the list L_0 as $L_0 = (L_0, (ID, PK, SK))$, return PK.
- $KeyGen - query$. When adversary A issues a query on ID_p, Challenger C look it up in L_0, find a match tuple (ID, PK, SK), and output SK as response.
- $DelGen - query$. On receiving the secret key SK_P, Challenger C outputs SK_D as response.
- $H_1 - query$. Challenger C maintains a list $L_1 = (Ay, \alpha_i)$, and the initial value is null. When the adversary A issue a query for $Aymodq$, Challenger C search it in list first, if there exist corresponding tuple (Ay, α), return α; otherwise, randomly chooses $\alpha \in \{0,1\}^{k_1+k_2}$. Update the list L_1 as $L_1 = (L_1, (Ay, \alpha))$, then return α.
- $F_1 - query$. Challenger C maintains a list $L_2 = (u_1, F_1(u_1))$, and the initial value is null. When the adversary A issue a query for u_1, Challenger C search it in list first, if there exist corresponding tuple $(u_1, F_1(u_1))$, return $F_1(u_1)$, otherwise, randomly chooses $F_1(u_1) \in \{0,1\}^{l_1}$, Update the list $L_2 = (L_2, (u_1, F_1(u_1)))$.
- $F_2 - query$. Challenger C maintains a list $L_3 = (F_1(u_1), F_2(F_1(u_1))$, and the initial value is null. When the adversary A issue a query for $F_1(u_1)$, Challenger C search it in list first, if there exist corresponding tuple $(F_1(u_1), F_2(F_1(u_1))$, return $F_2(F_1(u_1)$, otherwise, randomly chooses $F_2(F_1(u_1) \in \{0,1\}^{l_2}$, Update the list $L_3 = (L_3, (F_1(u_1), F_2(F_1(u_1))$.
- $H_2 - query$. Challenger C maintains a list $L_4 = (r_i, u_i, z_i, c_i)$. When the adversary A issues a query for $(r, u = u_1 \parallel u_2)$, the Challenger C search it in list first, if there exist corresponding tuple (r, u, c, z), return C; otherwise, randomly chooses vector $Z \in D_\sigma^m, C \in \{v : v \in \{-1,0,1\}^m\}$, $H_1 - query$ $AZ - (T_2 \parallel T_3) modq$ for α, let $u_1^{'} = \alpha \bigoplus r$, and according to $u_1^{'} = F_1(u_1) \parallel (F_2(F_1(u_1)) \bigoplus u_1)$, and Update $L_3 = (L_3, (F_1(u_1), F_2(F_1(u_1)) \bigoplus u_1)$, $L_2 = (L_2, (u_1, F_1(u_1)))$. Update $L_1 = (r, u, c, z)$ where r, u satisfied $H_2(r, u_2) = C$, then return C.
- $Psign - query$. To obtain a proxy signature on message u, adversary search it in L_4, if Challenger C find a match tuple(r, u, c, z), then output(r,z) as a response; otherwise, randomly choose vector $C \in \{-1,0,1\}^{k_1+k_2}$, $Z \in D_\sigma^m$, adversary A $H_1 - query$ $AZ - T_2 \parallel T_3 C$ for α, F_1-query and F_2-query for $(u_1, F_1(u_1))$ and $(F_1(u_1), (F_2(F_1(u_1)) \bigoplus u_1))$, let $r = \alpha \bigoplus u_1^{'}$. Update $L_1 = (L_1, (r, u, c, z))$ where $H(r, u_2) = C$. then return (r, u_2) and u_2.

Forgery. The adversary finally outputs a valid forgery (r, z, u_2^*) on message $u^* = u_1^* \parallel u_2^*$.

The specific example SIS problem: In order to solve SIS problem, adversary A should find a small vector $x \in \Lambda_q^{\perp}(A)$, Challenger responses to adversary A with different results when adversary A repeats his queries. According to General Forking Lemma [24], adversary A finally gets a valid forgery $(r^{\#}, Z^{\#}, u_2^*)$ on

same message $u^* = u_1^* \parallel u_2^*$ with non-negligible probability, and the following equation satisfied, where $C^{\#} = H_2(r^{\#}, u_2^*), H_2(r^*, u_2^*) = C^*$.

According to the above construction, we can see that for any message u:

- $H_2(r^{\#}, u_2^*) \neq H_2(r^*, u_2^*)$
- $AZ^{\#} - (T_2 \parallel T_3)H_2(r^{\#}, u_2^*) - (AZ^* - (T_2 \parallel T_3)H_2(r^*, u_2^*)) = A(Z^{\#} - Z^* - (S_2 \parallel S_3)C^{\#} + (S_2 \parallel S_3)C^*)$
- $\parallel Z^{\#} \parallel \leq 2\sigma\sqrt{m}, \parallel Z^* \parallel \leq 2\sigma\sqrt{m}, \parallel (S_2 \parallel S_3)C^{\#} \parallel \leq d_{k_1+k_2}\sqrt{m}$, and $\parallel (S_2 \parallel S{+}_3)C^* \parallel d_{k_1+k_2}\sqrt{m}$
- $\parallel Z^{\#} - Z^* - (S_2 \parallel S{+}_3)C^{\#} + (S_2 \parallel S{+}_3)C^* \parallel \leq (4\delta + 2d_{k_1+k_2})\sqrt{m}$

According to [21], $Z^{\#} - Z^* + (S_2 \parallel S_3C^{\#} - S_2 \parallel S_3)C^* \neq 0$ with probability at least 1/2, so $Z^{\#} - Z^* + ((S_2 \parallel S_3)C^{\#} - (S_2 \parallel S_3)C^*) \neq 0$ with non-negligible ability. Our lattice-based proxy signature scheme with message recovery is Unforgeability.

6 Efficiency Analysis

When we have a efficiency analysis of our scheme, We take signature size, computation time and energy cost into consideration. In this section, we analyse our scheme's efficiency by comparing it with some existing proxy signature schemes from the length of secret delegation key, proxy signature message and total time cost. In order to simplify the presentation, We define R = Rejection sampling algorithm computation cost, T = TrapGen algorithm computation cost, S = SamplePre algorithm computation cost, B = BasisDel algorithm computation cost, E = ExtBasis algorithm computation cost, M = Multiplication of matrix, M = Multiplication of vector, M= $m = O(n \log n)$, $q = O(n^2)$, $M = \omega(\sqrt{\log m})$, $S_1, S_2 \in \{-1, \cdots, 0, \cdots, 1\}^{m \times k_1}$, $\sigma = 12k_2\sqrt{m}$. Table 1 are given the detailed sizes and time of the comparison.

Table 1. Comparison of related proxy signature schemes

Proxy signature scheme	Delegation key length	Signature proxy	Time
[17]	$4m^2 \log(LM\sqrt{2m})$	$\|u\| + 2m \log(LM^2 2m)$	2T + 2S + E
[18]	$m^2 \log(LM\sqrt{2m})$	$\|u\| + m \log(LM^5 m^2)$	T + 3S + 3B
[19]	$ml' \log q$	$\|u\| + k + l + 2m \log(12\delta)$	R + M + V
Our's	$ml'' \log q$	$\|u_2\| + l_1 + l_2 + m \log(12\delta)$	$\approx R + (M + V)/2$

From Table 1, it is obvious that the total length (signed message and signature) of our message recovery scheme is less than other proposed schemes which is the foremost advantage of a lattice-based proxy signature scheme with message recovery, besides, we can find that in proxy signature [17,18], they mainly use TrapGen algorithm, SamplePre algorithm, BasisDel algorithm and ExtBasis algorithm which are very time consuming, while in our scheme and [14], based on

Lyubashevsky's rejection sampling algorithm, we mainly use the multiplication of matrix and vector, and due to different technique used in verification, we are almost twice as fast as [14].

When we let $k + l = l_1 + l_2$ and take security parameters mentioned in [21] into consideration $(n = 512, q = 2^{57}, d = 1)$, we can have a direct comparison with [19] as following

$$\triangle_{Lengthof ProxySignatureandMessage} \approx l_2 + mlog(12\delta) = (l_2 + 163000)bits \quad (2)$$

Refer to the comparison proposed in [22], 1 bit transmission cost more energy than 32 bits simple operation, in that case, even we increase simple computation (like hash and XOR) in message recovery technology, our scheme still cost much less energy than [19]'s in pracRef1tical situation. According to the above analysis, our message recovery proxy signature scheme is more efficient than these existing schemes in signature size, time and energy cost.

7 Conclusion

With the development of quantum computers, constructing an efficient quantum-secure proxy signature scheme enjoys priority. Lattice-based signature occupies a position of particular interest, as it relies on well-studied problems and comes with uniquely strong security guarantees, such as worst-case to average case reductions. In this work, we proposed an efficient lattice-based proxy signature with message recovery which is possible to be the first proxy signature scheme with message recovery that can resist known quantum attack, and we give a formal proof of it's security in the random oracle model. In addition, compared with some existing proxy signature schemes, our scheme is more efficient than others in signature length, signature time and energy cost. Contribution to rich theoretical foundation of Lattice Cryptography, we will design more efficient lattice-based signature schemes with message recovery in the future.

Acknowledgements. This work was supported by NSFC (61402030), the Major Program of National Natural Science Foundation of China (11290141), and Fundamental Research of Civil Aircraft no. MJ-F-2012-04.

References

1. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures: delegation of the power to sign messages. IEICE Trans. Fundam. A **79**(9), 1338–1354 (1996)
2. Kumar, R., Verma, H.K., Dhir, R.: Analysis and design of protocol for enhanced threshold proxy signature scheme based on rsa for known signers. Wirel. Pers. Commun. **80**(3), 1281–1345 (2015)
3. Xiao, Y.M.: Improvement of an Eliptic curve based threshold proxy signature scheme (2016)
4. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In: Quantum Entanglement and Quantum Information-Proceedings of Ccast, pp. 303–332 (1999)

5. Tang, S., Xu, L.: Proxy signature scheme based on isomorphisms of polynomials. In: Xu, L., Bertino, E., Mu, Y. (eds.) NSS 2012. LNCS, vol. 7645, pp. 113–125. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34601-9_9
6. Yang, C., Qiu, P., Zheng, S., Wang, L.: An efficient lattice-based proxy signature scheme without trapdoor. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 189–194 (2016)
7. Chen, Y.Z., Liu, Y., Wen, X.J.: A quantum proxy weak blind signature scheme. Chin. J. Quantum Electron. **54**(4), 1325–1333 (2011)
8. Zhang, L., Ma, Y.: A lattice-based identity-based proxy blind signature scheme in the standard model. Math. Probl. Eng. **2014**(1) (2014)
9. Wang, T.Y., Wei, Z.L.: Analysis of forgery attack on one-time proxy signature and the improvement. Int. J. Theor. Phys. **55**(2), 1–3 (2015)
10. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. In: IEEE Symposium on Foundations of Computer Science, pp. 372–381 (2004)
11. Nyberg, K., Rueppel, R.A.: A new signature scheme based on the DSA giving message recovery. In: Proceedings of the ACM Conference on Computer and Communications Security, CCS 1993, Fairfax, Virginia, USA, pp. 58–61. November 1993
12. Simoens, P., Vankeirsbilck, B., Deboosere, L., Ali, F.A., Turck, F.D., Dhoedt, B., Demeester, P.: Upstream bandwidth optimization of thin client protocols through latency-aware adaptive user event buffering. Int. J. Commun. Syst. **24**(5), 666–690 (2011)
13. Liu, C.X., Liu, Y., Zhang, Z.J., Cheng, Z.Y.: High energy-efficient and privacy-preserving secure data aggregation for wireless sensor networks. Int. J. Commun. Syst. **26**(3), 380–394 (2013)
14. Padhye, S., Tiwari, N.: ECDLP-Based Certificateless Proxy Signature Scheme with Message Recovery. Wiley, Hoboken (2015)
15. Zhou, C.: An improved ID-based proxy signature scheme with message recovery. Int. J. Secur. Appl. **9**(9), 151–164 (2015)
16. Asaar, M.R., Salmasizadeh, M., Susilo, W.: A Short ID-Based Proxy Signature Scheme. Wiley, Hoboken (2016)
17. Xia, F., Yang, B., Ma, S., Sun, W.W., Zhang, M.W.: Lattice-based proxy signature scheme. J. Hunan Univ. **38**(6), 84–88 (2011)
18. Kim, K.S., Hong, D., Jeong, I.R.: Identity-based proxy signature from lattices. J. Commun. Netw. **15**(1), 1–7 (2013)
19. Jiang, M.M., Yupu, H.U., Baocang, Y.U., Lai, J.J.: Efficient lattice-based proxy signature. J. Beijing Univ. Posts Telecommun. (2014)
20. Tian, M., Huang, L.: Lattice-based message recovery signature schemes. Int. J. Electron. Secur. Digit. Forensics **5**(3/4), 257–269 (2013)
21. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_43
22. Barr, K.C.: Energy-aware lossless data compression. ACM Trans. Comput. Syst. **24**(3), 250–291 (2006)
23. Ajtai, M.: Generating hard instance of lattice problems. In: Twenty-Eighth ACM Symposium on Theory of Computing ACM, pp. 99–108 (1996)
24. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, pp. 390–399, October 30–November 03 2006