Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Post-quantum blind signcryption scheme from lattice*

Huifang YU†‡, Lu BAI

School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

†E-mail: yuhuifang@xupt.edu.cn

Received Mar. 7, 2020; Revision accepted Sept. 7, 2020; Crosschecked Apr. 1, 2021

Abstract: Blind signcryption (BSC) can guarantee the blindness and untrackability of signcrypted messages, and moreover, it provides simultaneous unforgeability and confidentiality. Most traditional BSC schemes are based on the number theory. However, with the rapid development of quantum computing, traditional BSC systems are faced with severe security threats. As promising candidate cryptosystems with the ability to resist attacks from quantum computing, lattice-based cryptosystems have attracted increasing attention in academic fields. In this paper, a post-quantum blind signcryption scheme from lattice (PQ-LBSCS) is devised by applying BSC to lattice-based cryptosystems. PQ-LBSCS inherits the advantages of the lattice-based cryptosystem and blind signcryption technique. PQ-LBSCS is provably secure under the hard assumptions of the learning with error problem and small integer solution problem in the standard model. Simulations are carried out using the Matlab tool to analyze the computational efficiency, and the simulation results show that PQ-LBSCS is more efficient than previous schemes. PQ-LBSCS has extensive application prospects in e-commerce, mobile communication, and smart cards.

Key words: Lattice-based cryptosystem; Blind signcryption; Post-quantum computing; Learning with error assumption; Small integer solution assumption

https://doi.org/10.1631/FITEE.2000099 **CLC number:** TP309

1 Introduction

Ajtai (1996) discovered a link between the worst-case complexity and the average-case complexity of some well-known lattice problems. Ajtai and Dwork (1997) proposed a public key cryptosystem based on the lattice theory, thereby creating a new field in public key cryptosystems. Scholars proposed some public key cryptosystems from lattice, such as GGH (Garg et al., 2013), NTRU (Hoffstein et al., 1998), and Regev (Regev, 2009) systems. In recent years, lattice-based cryptosystems have attracted increasing attention of many scholars because of their ability to resist quantum computing attacks and the low computational complexity (Li et al., 2013).

Signcryption can complete signature and public key encryption operations at the same time with lower calculation and communication cost compared with traditional approaches. Yan et al. (2015) proposed an identity-based signcryption scheme from lattice. Lu et al. (2016) proposed a lattice-based signcryption scheme without trapdoors. Gerard and Merckx (2018) devised a post-quantum signcryption scheme from lattice. Researchers also proposed some other signcryption algorithms from lattice (Sato and Shikata, 2018; Liu et al., 2019). However, as an important but cryptographic primitive, the development of lattice-based signcryption is relatively slow.

Blind signcryption can be obtained by applying the signcryption technique to a blind signature system. At present, there are several blind signature schemes from lattice (Yuen and Wei, 2005; Tian et al., 2016; Ye et al., 2018); however, there is no blind signcryption scheme from lattice. Because traditional blind signcryption schemes are constructed based on the

‡ Corresponding author

* Project supported by the Key Project of Natural Science Foundation Basic Research Program of Shaanxi Province, China (No. 2020JZ-54) and the Innovation Foundation of Postgraduate of Xi'an University of Posts and Telecommunications, China (No. CXJJLY2018075)

ORCID: Huifang YU, https://orcid.org/0000-0003-4711-3128

number theory, they are no longer secure in the environments of quantum computing. Hence, it is necessary to construct a secure and efficient blind signcryption scheme from lattice which can resist quantum computing attacks.

As a response to security requirements, a post-quantum blind signcryption scheme from lattice (PQ-LBSCS) is proposed, and it is based on the lattice-based cryptosystems (Yan, 2015) and conventional blind signcryption (Zia and Ali, 2019). A trapdoor generation method and a blind signature function are added based on the signcryption from lattice. PQ-LBSCS can resist quantum computing attacks and has the security properties of blindness, untraceability, confidentiality, and unforgeability. PQ-LBSCS has lower computational complexity and higher communicational efficiency than previous schemes. In the future, these merits will promote its applications in engineering.

2 Preliminaries

2.1 Notations

Notations used in this paper are listed in Table 1.

Table 1 Notations used in this paper

Notation	Meaning
s	Gaussian parameter for encryption
s_s	Gaussian parameter $s_s = O\left(\sqrt{\lambda nk}\right)\omega\left(\sqrt{\log n}\right)^2 \beta$
H_F	Chameleon hash function
A_i	Public key of the user
T_i	Private key of the user
M	Plaintext in our scheme
$\boldsymbol{c}$	Ciphertext in our scheme
k	An integer
l	A random integer
H_i	Hash function
$\boldsymbol{e}$	A vector in lattice
$\boldsymbol{Z}$	Center of the discrete Gaussian distribution
$\boldsymbol{x}$	A vector selected from the Gaussian distribution
$\boldsymbol{v}$	A verification matrix in the unsigncryption function
Z_q	Residue class ring by module q
$U(\cdot)$	A uniform distributed function
k'	Encryption algorithm parameter
w	Length of a sequence
$\hbar(\cdot)$	Mapping in the security certificate

2.2 Lattice definition

Definition 1 (Integer lattice) Given an invertible matrix of m rows and m columns $\boldsymbol{B}=[\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m]\in Z^{m\times m}$, where Z represents the residue class ring and vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m\in Z^m$ are linearly independent, an m-dimensional integer lattice generated by matrix $\boldsymbol{B}$ is expressed as follows:

$$\boldsymbol{\Lambda}=\left\{\boldsymbol{y}\in Z^m : \boldsymbol{y}=\boldsymbol{BZ}=\sum_{i=1}^{m}\boldsymbol{Z}b_i, \boldsymbol{Z}\in Z^m\right\}, \quad (1)$$

where $\boldsymbol{B}$ is the set of bases of lattice $\boldsymbol{\Lambda}$.

Definition 2 (q-module lattice) Given a matrix $\boldsymbol{A}\in Z_q^{n\times m}$, where $Z_q^{m\times n}$ is a q-module residual class matrix ring of n rows and m columns (here, m, n, and q are positive integers), the q-module lattice is defined as

$$\boldsymbol{\Lambda}^{\perp}(\boldsymbol{A})=\{\boldsymbol{e}\in Z^m : \boldsymbol{Ae}=\boldsymbol{0} \bmod q\}, \quad (2)$$

$$\boldsymbol{\Lambda}_{\boldsymbol{u}}^{\perp}(\boldsymbol{A})=\{\boldsymbol{e}\in Z^m : \boldsymbol{Ae}=\boldsymbol{u} \bmod q\}, \quad (3)$$

where $\boldsymbol{u}\in Z_q^n$, n is the rank of the lattice, and m is the dimension of the lattice.

2.3 Discrete Gaussian distribution

Definition 3 (Discrete Gaussian distribution) For any real parameter $s>0$, the discrete Gaussian distribution density function of lattice $\boldsymbol{\Lambda}$ centered on $\boldsymbol{Z}\in Z^m$ is defined as

$$\forall \boldsymbol{y}\in\boldsymbol{\Lambda}, \rho_{s,\boldsymbol{Z}}(\boldsymbol{y})=\exp\left(-\pi\frac{\|\boldsymbol{y}-\boldsymbol{Z}\|^2}{s^2}\right). \quad (4)$$

For $\rho_{s,\boldsymbol{Z}}(\boldsymbol{\Lambda})=\sum_{\boldsymbol{y}\in\boldsymbol{\Lambda}}\rho_{s,\boldsymbol{Z}}(\boldsymbol{y})$, the Gaussian distribution on $\boldsymbol{\Lambda}$ is defined as follows:

$$\forall \boldsymbol{y}\in\boldsymbol{\Lambda}, D_{\boldsymbol{\Lambda},s,\boldsymbol{Z}}(\boldsymbol{y})=\frac{\rho_{s,\boldsymbol{Z}}(\boldsymbol{y})}{\rho_{s,\boldsymbol{Z}}(\boldsymbol{\Lambda})}. \quad (5)$$

Lemma 1 (Property of Gaussian distribution) Given a prime number $q\geq 3$, the positive integers n and m satisfying $m\geq 2n\log q$, a vector $\boldsymbol{T}$ of bases for $\boldsymbol{\Lambda}_q^{\perp}(\boldsymbol{A})$, and Gaussian parameter $s\geq\|\boldsymbol{T}\|\omega\left(\sqrt{\log m}\right)$, for any vector $\boldsymbol{w}$, we have

(1) $\Pr\left[\boldsymbol{x}\leftarrow D_{\boldsymbol{\Lambda},s,\boldsymbol{Z}}:\|\boldsymbol{x}\|>s\sqrt{m}\right]\le \mathrm{negl}(n)$;

(2) For a random matrix $\boldsymbol{A}\in Z_q^{n\times m}$, if $\boldsymbol{e}\leftarrow D_{Z^m,s}$, then the distribution of $\boldsymbol{y}=\boldsymbol{Ae} \bmod q$ is close to the uniform distribution on Z_q^n.

Theorem 1 Given an m-dimensional lattice $\boldsymbol{\Lambda}$, $\boldsymbol{v}$ is a point in the space formed by the lattice basis vector and n is a positive integer. For any random real numbers $\varepsilon>0$ and $s>\eta(\boldsymbol{\Lambda})$, there is

$$\Pr_{X\sim D_{\boldsymbol{\Lambda},s,\boldsymbol{v}}}\left\{\|\boldsymbol{x}-\boldsymbol{v}\|>s\sqrt{n}\right\}\le 2^{-m}\frac{1+\varepsilon}{1-\varepsilon}. \tag{6}$$

The probability of obtaining the same vector twice under a discrete Gaussian distribution is approximately zero; i.e., the entropy of the distribution is large and the probability of preimage collision is approximately zero.

2.4 Difficult problems on lattice

Definition 4 (Learning with error (LWE) problem) Assume that $n\ge 1$ is a positive integer and that χ is the Gaussian noise distribution in Z_q^m, where the modulus $q\ge 2$. The probability distribution $\boldsymbol{A}_{s,\chi}$ is obtained in the following way: matrix $\boldsymbol{A}\in Z_q^{n\times m}$ and vector $\boldsymbol{s}\in Z_q^n$ are randomly and uniformly selected, noise vector $\boldsymbol{e}\in Z_q^m$ is randomly extracted from the Gaussian noise distribution χ, and $(\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e})$ is output. Thus, there are two types of LWE problems:

(1) Search-type LWE (SLWE) problem. Given m independent samples $(\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e})$ obeying the distribution $\boldsymbol{A}_{s,\chi}$, $\boldsymbol{s}\in Z_q^n$ is output with a negligible probability.

(2) Decision-type LWE (DLWE) problem. For each distinguishing sample $(\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e})$ obtained by the above algorithm, it is necessary to determine whether the sample is randomly chosen from the distribution $\boldsymbol{A}_{s,\chi}$ or from the uniform distribution of $Z_q^{n\times m}\times Z_q^m$.

Lemma 2 (Hardness of the LWE problem) Given the prime numbers $\alpha=\alpha(n)\in\{0, 1\}$ and $q\le 2^n$, where $\alpha q>2\sqrt{n}$, if there is an effective algorithm that can solve the LWE problem, then there exists an effective quantum algorithm that can approximately solve the shortest independent vector (SIV) problem with an approximate factor of $\tilde{O}(n/\alpha)$.

Definition 5 (Small integer solution (SIS) problem) Assume that n and m are positive integers and that q is a prime number. Given that $\beta>0$ is a small real number and that $\boldsymbol{A}$ is randomly chosen on $Z_q^{n\times m}$, the SIS problem is to find a short vector $\boldsymbol{v}$ on lattice $\boldsymbol{\Lambda}_q^{\perp}(\boldsymbol{A})$ such that $\boldsymbol{Av}=\boldsymbol{0} \bmod q$, where the norm of this short vector satisfies $\|\boldsymbol{v}\|\le\beta$.

Lemma 3 (Hardness of the SIS problem) For any β and prime number $q\ge\beta\omega(\sqrt{n\log n})$ defined by polynomial poly(n), the hardness of the SIS problem is equivalent to that of the SIV problem when the approximation factor is $\gamma=\beta\tilde{O}(\sqrt{n})$.

2.5 Chameleon hash function on lattice

Definition 6 (Chameleon hash function) $H=\{h_i: K\times R\to y\}$ is a hash function which maps message space K and random number space R to range y. The hash function satisfies four properties as follows:

(1) Anti-collision.

(2) Inputting message M and random number r, the hash function value $h_i(M, r)$ can be efficiently calculated, and this shows that the hash function has the nature of forward calculation.

(3) The hash function value is statistically close to a uniform distribution, for $h_i\leftarrow H$, $M\leftarrow K$, $r\leftarrow R$, and $(h_i, h_i(M, r))$ being uniformly distributed on (H, y).

(4) A Chameleon hash function has the ability to find a collision and hash preimage through trapdoors because each hash function h_i is associated with a pair of public and private keys, and here the public keys are from the same distribution. For any h_i, M_1, M_2, and r_1, the Chameleon hash function can use the private key corresponding to h_i to obtain a pair of collisions; i.e., it is possible to find r_2 satisfying $h_i(M_1, r_1)=h_i(M_2, r_2)$.

Definition 7 (Universal hash function) For different x, $x'\in X$ and $\Pr_{h\leftarrow H}[h(x)=h(x')]=1/|\boldsymbol{y}|$, the function family $H=\{h: X\to\boldsymbol{y}\}$ is said to be a universal hash function.

2.6 Preimage sampling algorithm

Theorem 2 (Preimage sampling theorem) Assume

that m, n, and q are integers and $q>2$ and $m>n$. Input matrix $\boldsymbol{A}\in Z_q^{n\times m}$, trapdoor $\boldsymbol{T}_A$ on lattice $\Lambda^{\perp}(\boldsymbol{A})$, vector $\boldsymbol{y}\in Z^m$, and real number $s>\|\tilde{\boldsymbol{T}}_A\|\omega\left(\sqrt{\log m}\right)$, output a preimage $\boldsymbol{x}\in Z^m$ of $\boldsymbol{y}$ using the preimage sampling algorithm SamplePre($\boldsymbol{A}$, $\boldsymbol{T}_A$, $\boldsymbol{y}$, s), and ignore the statistical distance between $\boldsymbol{x}$ and $D_{\Lambda_q^{\boldsymbol{y}}(\boldsymbol{A}),s}$.

3 Formal definition

The post-quantum blind signcryption scheme from lattice (PQ-LBSCS) consists of four probabilistic polynomial time algorithms: Setup, KeyGen, BlindS, and Unsigncrypt.

1. Setup. A trusted authority (TA) executes this algorithm which inputs the security parameter 1^n and outputs the public parameter named "params."

2. KeyGen. Every user in the system runs the key generation algorithm once, inputting params and outputting each user's public and private key pair ($\boldsymbol{A}_i$, $\boldsymbol{T}_i$). Users publicize their public keys and retain the private keys. Here, the blind signcrypter's public and private key pair is ($\boldsymbol{A}_\mathrm{s}$, $\boldsymbol{T}_\mathrm{s}$) and the receiver's public and private key pair is ($\boldsymbol{A}_\mathrm{r}$, $\boldsymbol{T}_\mathrm{r}$).

3. BlindS. The message owner and blind signcrypter execute this algorithm together. The message owner sends the blinded message to the blind signcrypter who does not know the content of message. Given the message M and the public key $\boldsymbol{A}_\mathrm{r}$ of the receiver along with the public and private key pair ($\boldsymbol{A}_\mathrm{s}$, $\boldsymbol{T}_\mathrm{s}$) of the blind signcrypter, this blind signcryption algorithm outputs a signcrypted ciphertext $\boldsymbol{c}$.

4. Unsigncrypt. Given the ciphertext $\boldsymbol{c}$ and the public and private key pair ($\boldsymbol{A}_\mathrm{r}$, $\boldsymbol{T}_\mathrm{r}$) of the receiver along with the public key $\boldsymbol{A}_\mathrm{s}$ of the sender, this algorithm outputs a plain text M or a symbol $\perp$.

PQ-LBSCS must satisfy the requirements of confidentiality (indistinguishability under adaptive ciphertext-chosen attacks) and unforgeability (strong unforgeability under adaptive ciphertext-message attacks). For the confidentiality requirement, we refer to the security model of YML (Yan et al., 2013). For the unforgeability requirement, we refer to the security model of YLM (Yan et al., 2019). For simplicity, we omit details of the security models of PQ-LBSCS.

4 PQ-LBSCS

4.1 Setup

A TA carries out this algorithm and generates the system parameters params=($\boldsymbol{G}$, α, s', H_0, H_1, H_2, $H_{\boldsymbol{F}}$, H_3, B, s_s). The concrete steps are as follows:

Step 1: pick a matrix $\boldsymbol{G}\in Z_q^{n\times nk}$, where q=poly(n) is a large prime number and $k=O(\log q)=O(\log n)$.

Step 2: $m_0=O(nk)$, $m=m_0+nk$, $m_1=m_0+2nk$.

Step 3: $\alpha\in(0, 1)$ is the error rate of LWE and satisfies $1/\alpha=O(nk)\omega\left(\sqrt{\log n}\right)$. $s'\geq\sqrt{5}\omega\left(\sqrt{\log n}\right)$ is a Gaussian parameter which will be used for encryption.

Step 4: define hash functions: $H_0:\{0,1\}^*\to Z_q^n$, $H_1:\{0,1\}^*\times Z_q^{n\times m}\to\{0,1\}^{\lambda}$, $H_2:Z_q^{m_1}\to\{0,1\}^l$, $H_3:\{0,1\}^{nk}\times Z_{2q}^m\to\{0,1\}^{nk'}$, $H_{\boldsymbol{F}}:\{0,1\}^l\times Z_q^m\to R^*$, where H_0–H_3 are universal hash functions, $H_{\boldsymbol{F}}$ is selected from a family of universal hash functions, $\boldsymbol{F}$ is a matrix on $Z_q^{n/(l+m)}$, $\boldsymbol{F}=\left[\boldsymbol{F}^{(0)}\middle\|\boldsymbol{F}^{(1)}\right]\in Z_q^{nl}\times Z_q^{nm}$, and $\boldsymbol{F}^{[0]}$(resp.$\boldsymbol{F}^{(1)}$) and $\boldsymbol{N}^{[0]}$(resp.$\boldsymbol{N}^{(1)}$) are taken from the same distribution.

Step 5: define a set $L=\left\{\boldsymbol{B}^{(0)},\cdots,\boldsymbol{B}^{(\lambda)}\right\}$, where $\boldsymbol{B}^{(i)}\xleftarrow{\$}U\left(Z_q^{n\times(nk)}\right)$.

4.2 KeyGen

In this phase, each user generates his/her public key $\boldsymbol{A}_i$ and private key $\boldsymbol{T}_i$ (i=s, r) as follows:

Step 1: pick $\boldsymbol{A}_i^{(0)}\xleftarrow{\$}Z_q^{nm_0}$, $\boldsymbol{T}_i\xleftarrow{\$}D_{Z,\omega(\sqrt{\log n})}^{m_0\times nk}$.

Step 2: calculate $\boldsymbol{A}_i^{(1)}=-\boldsymbol{A}_i^{(0)}\boldsymbol{T}_i$.

Step 3: set $\boldsymbol{A}_i=\left[\boldsymbol{A}_i^{(0)}\middle\|\boldsymbol{A}_i^{(1)}\right]$, where $\boldsymbol{A}_i\in Z_q^{n\times m}$.

4.3 BlindS

Assuming that M is a message, the interaction between the blind signcrypter S and the message owner Alice is shown in Fig. 1, and the details are as follows:

Step 1: Alice calculates $\boldsymbol{g}=H_0(M)$ and randomly chooses vector $\boldsymbol{Z}=(c_1, c_2, .., c_m)$ in a discrete normal distribution $D_{Z^m,a\omega(\sqrt{\log n})}$; $\|\boldsymbol{Z}\|\leq a\omega\left(\sqrt{\log n}\right)\sqrt{m}$ holds with the extreme probability. Because the

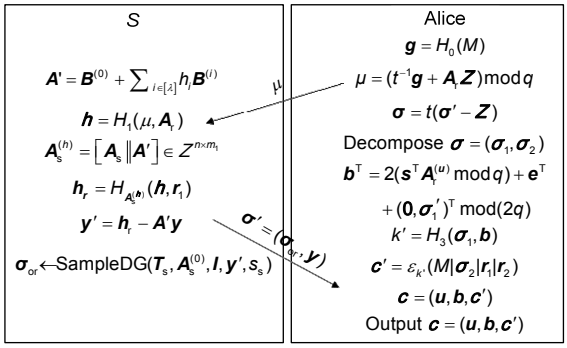

Fig. 1 Blind signcryption scenarios

sample $\boldsymbol{Z}$ satisfies Lemma 2, $\boldsymbol{A}_r\boldsymbol{Z}$ approximately obeys a uniform distribution.

Step 2: Alice randomly selects $t\in Z$ to calculate $\mu=(t^{-1}\boldsymbol{g}+\boldsymbol{A}_r\boldsymbol{Z})\bmod q$ and delivers μ to S, where $1<t<x<\|\boldsymbol{B}\|-1$.

Step 3: S calculates $\boldsymbol{A}'=\boldsymbol{B}^{(0)}+\sum_{i\in[\lambda]}h_i\boldsymbol{B}^{(i)}$ and $\boldsymbol{h}=H_1(\mu,\boldsymbol{A}_r)$, and constructs $\boldsymbol{A}_s^{(h)}=[\boldsymbol{A}_s\|\boldsymbol{A}']\in Z^{n\times m_1}$, where $\boldsymbol{h}=(h_1,h_2,\ldots,h_\lambda)\in\{0,1\}^\lambda$ is the binary representation of $\boldsymbol{h}$.

Step 4: S constructs a Chameleon hash function $H_{A_s^{(h)}}$ according to the method mentioned in Section 2.5.

Step 5: S chooses $\boldsymbol{r}_1\in D_{Z^{nk},s_s}$ and $\boldsymbol{y}\leftarrow D_{Z^{nk},s_s}$, and calculates $\boldsymbol{h}_r=H_{A_s^{(h)}}(h,\boldsymbol{r}_1)$ and $\boldsymbol{y}'=\boldsymbol{h}_r-\boldsymbol{A}'\boldsymbol{y}$.

Step 6: S selects $\boldsymbol{\sigma}_{or}\leftarrow\mathrm{SampleDG}(\boldsymbol{T}_s,\boldsymbol{A}_s^{(0)},\boldsymbol{I},\boldsymbol{y}',s_s)$, and delivers $\boldsymbol{\sigma}'=(\boldsymbol{\sigma}_{or},\boldsymbol{y})$ to Alice, where $\boldsymbol{I}\in Z^{n\times n}$ is a unit matrix.

Step 7: Alice calculates $\boldsymbol{\sigma}=t(\boldsymbol{\sigma}'-\boldsymbol{Z})$, where $\boldsymbol{\sigma}$ is a signature of message M.

Step 8: Alice decomposes $\boldsymbol{\sigma}=(\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2)$ and $w=H_2(\boldsymbol{\sigma})$, where $\boldsymbol{\sigma}_1\in\{0,1\}^{nk}$ and the rest is written as $\boldsymbol{\sigma}_2$.

Step 9: Alice extracts $\boldsymbol{e}_0\leftarrow D_{Z,\alpha q}^{m_0}$ to calculate $s_r=\sqrt{\|\boldsymbol{e}_0\|^2+m_0(\alpha q)^2}$, and then extracts $\boldsymbol{e}_1\leftarrow D_{Z,s_r}^{nk}$ and sets $\boldsymbol{e}=(\boldsymbol{e}_0,\boldsymbol{e}_1)$.

Step 10: Alice extracts $\boldsymbol{r}_2\leftarrow D_{Z^{nk},s'}$ to calculate $\boldsymbol{u}=H_{\boldsymbol{F}}(w,\boldsymbol{r}_2)$ and constructs $\boldsymbol{A}_r^{(u)}=[\boldsymbol{A}_r\|\boldsymbol{A}_r^{(1)}+\hbar(\boldsymbol{u})\boldsymbol{G}]$.

Step 11: Alice uniformly chooses a vector $\boldsymbol{s}\xleftarrow{\$}Z_q^n$, and calculates $\boldsymbol{\sigma}_1'=\mathrm{encode}(\boldsymbol{\sigma}_1)\in Z^{nk}$ and $\boldsymbol{b}^{\mathrm{T}}=2(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}_r^{(u)}\bmod q)+\boldsymbol{e}^{\mathrm{T}}+(\boldsymbol{0},\boldsymbol{\sigma}_1')^{\mathrm{T}}\bmod(2q)$.

Step 12: Alice calculates $k'=H_3(\boldsymbol{\sigma}_1,\boldsymbol{b})$ and $\boldsymbol{c}'=\varepsilon_{k'}(M|\boldsymbol{\sigma}_2|\boldsymbol{r}_1|\boldsymbol{r}_2)$, and then outputs the ciphertext $\boldsymbol{c}=(\boldsymbol{u},\boldsymbol{b},\boldsymbol{c}')$.

4.4 Unsigncrypt

After receiving $\boldsymbol{c}=(\boldsymbol{u},\boldsymbol{b},\boldsymbol{c}')$, the receiver Bob carries out this unsigncryption algorithm as follows:

Step 1: decrypt $(\boldsymbol{u},\boldsymbol{b})$ to obtain

$$\boldsymbol{\sigma}_1=\boldsymbol{Q}^{-1}\left(\boldsymbol{v}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{T}_r\\ \boldsymbol{I}\end{bmatrix}\bmod(2q)\right). \tag{7}$$

Output $\perp$ if $\boldsymbol{u}=\boldsymbol{0}$, and continue otherwise.

Step 2: employ $\mathrm{Invert}^{\wp}(\boldsymbol{T}_r,\boldsymbol{A}_s^{(u)},\boldsymbol{b},\boldsymbol{u})$ (Yan et al., 2013) to obtain $(\boldsymbol{z},\boldsymbol{e})$, where $\boldsymbol{e}=(\boldsymbol{e}_0,\boldsymbol{e}_1)\in Z^{m_0}\times Z^{nk}$ and $\boldsymbol{z}\in Z_q^n$. Output $\perp$ if $\|\boldsymbol{e}_1\|\geq\alpha q\sqrt{2m_0nk}\,\omega\left(\sqrt{\log n}\right)$ or $\|\boldsymbol{e}_0\|\geq\alpha q\sqrt{m_0}$, and continue otherwise.

Step 3: set $\boldsymbol{v}=\boldsymbol{b}-\boldsymbol{e}\bmod q$ and $\boldsymbol{v}=(\boldsymbol{v}_1,\boldsymbol{v}_2)\in Z_{2q}^{m_0}\times Z_{2q}^{nk}$. Output $\perp$ if $\boldsymbol{v}_1\notin 2\boldsymbol{\Lambda}((\boldsymbol{A}_s^{(0)})^{\mathrm{T}})$, and continue otherwise.

Step 4: calculate $D_{k'}(\boldsymbol{c}')=(M|\boldsymbol{\sigma}_2|\boldsymbol{r}_1|\boldsymbol{r}_2)$, where $k'=H_3(\boldsymbol{\sigma}_1,\boldsymbol{b})$.

Step 5: calculate $w=H_1(\boldsymbol{\sigma})$, where $\boldsymbol{\sigma}=(\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2)$. Output $\perp$ if $\boldsymbol{u}\neq\boldsymbol{H}_{\boldsymbol{F}}(w,\boldsymbol{r}_2)$, and continue otherwise. Output $\perp$ if $\|\boldsymbol{r}_2\|\geq s'\sqrt{m}$, and continue otherwise.

Step 6: verify whether the sender is authentic or not:

(1) Calculate $\boldsymbol{h}=H_1(M,\boldsymbol{A}_r)$ and construct $\boldsymbol{A}_s^{(h)}=[\boldsymbol{A}_s\|\boldsymbol{B}^{(0)}+\sum_{i\in[\lambda]}h_i\boldsymbol{B}^{(i)}]$. Output $\perp$ if $\|\boldsymbol{\sigma}\|>s_s\sqrt{m_1}$, and continue otherwise.

(2) Calculate $\boldsymbol{h}_r=H_{A_s^{(h)}}(\boldsymbol{h},\boldsymbol{r}_1)$. Output M if $\boldsymbol{A}_s^{(h)}\boldsymbol{\sigma}=\boldsymbol{h}_r$ and $\perp$ otherwise.

Theorem 3 PQ-LBSCS is correct under the hard assumptions of SIS and LWE problems.

Proof Under the LWE and SIS assumptions, PQ-LBSCS can be correctly unsigncrypted with the probability of $1-2^{-\Omega(n)}$. Assuming that $\boldsymbol{c}=(\boldsymbol{u},\boldsymbol{b},\boldsymbol{c}')$ is a valid ciphertext, we now analyze the process of the

unsigncryption algorithm. First, we prove that $\boldsymbol{\sigma}_1$ can be obtained with an overwhelming probability in the unsigncryption algorithm. Assuming that $\boldsymbol{T}_\mathrm{r}$ is the $\boldsymbol{G}$ trapdoor of $\boldsymbol{A}_\mathrm{r}^{(\boldsymbol{u})}$ and $\boldsymbol{R}_I=[\boldsymbol{T}_\mathrm{r}\quad \boldsymbol{I}]^\mathrm{T}$, $\boldsymbol{Q}$ is a base of $\Lambda(\boldsymbol{G}^\mathrm{T})$. Then, there must exist a matrix $\boldsymbol{Q}'\in Z_q^{n\times(nk)}$ satisfying $\boldsymbol{Q}=\boldsymbol{G}^\mathrm{T}\boldsymbol{Q}'$. Herewith, we have

$$\begin{aligned}
&\hat{\boldsymbol{b}}^\mathrm{T} \bmod q\\
&=\boldsymbol{b}^\mathrm{T}\boldsymbol{R}_I\\
&=\left[2(\boldsymbol{s}^\mathrm{T}\boldsymbol{A}_\mathrm{r}^{(\boldsymbol{u})} \bmod q)+\boldsymbol{e}^\mathrm{T}+(\boldsymbol{0},\boldsymbol{\sigma}')\right]\boldsymbol{R}_I\\
&=\left\{2\left[\boldsymbol{s}^\mathrm{T}\left(\boldsymbol{A}_\mathrm{r}^{(\boldsymbol{0})}\|-\boldsymbol{A}_\mathrm{r}^{(\boldsymbol{0})}\boldsymbol{T}_\mathrm{r}+\hbar(\boldsymbol{u})\boldsymbol{G}\right)\bmod q\right]\right.\\
&\quad\left.+(\boldsymbol{e}_0,\boldsymbol{e}_1)^\mathrm{T}+\left(\boldsymbol{0},(\boldsymbol{Q}\boldsymbol{\sigma}_1)^\mathrm{T}\right)\right\}\boldsymbol{R}_I\\
&=2\left(\boldsymbol{s}^\mathrm{T}\hbar(\boldsymbol{u})\boldsymbol{G}\bmod q\right)+(\boldsymbol{e}_0^\mathrm{T}\boldsymbol{T}_\mathrm{r}+\boldsymbol{e}_1^\mathrm{T})+(\boldsymbol{G}^\mathrm{T}\boldsymbol{Q}'\boldsymbol{\sigma}_1)^\mathrm{T}\\
&=\left(2\boldsymbol{s}^\mathrm{T}+\boldsymbol{\sigma}_1^\mathrm{T}\boldsymbol{Q}'\hbar(\boldsymbol{u})^{-1}\right)\hbar(\boldsymbol{u})\boldsymbol{G}+(\boldsymbol{e}_0^\mathrm{T}\boldsymbol{T}_\mathrm{r}+\boldsymbol{e}_1^\mathrm{T}).
\end{aligned}\tag{8}$$

Obviously, $\hat{\boldsymbol{b}}^\mathrm{T}=\boldsymbol{s}^\mathrm{T}\boldsymbol{G}+\boldsymbol{e}^\mathrm{T}$. Because $\boldsymbol{T}_\mathrm{r}$ is a $\boldsymbol{G}$ trapdoor for $\boldsymbol{A}_\mathrm{r}$, it can return a vector $\boldsymbol{e}\in Z^m$.

Next, we show that the effective noise vector $\boldsymbol{e}$ can be obtained using the function $\mathrm{Invert}^{\wp}$ with an overwhelming probability. If $\wp(\hat{\boldsymbol{b}})$ in $\mathrm{Invert}^{\wp}$ wants to obtain a desired value, $\mathrm{Invert}^{\wp}$ returns a correct $\boldsymbol{e}$, where the algorithm $\wp(\hat{\boldsymbol{b}})$ is realized using Invert$\boldsymbol{G}$. As long as the error vector $\boldsymbol{e}_0^\mathrm{T}\boldsymbol{T}_\mathrm{r}+\boldsymbol{e}_1^\mathrm{T}\in P_{1/2}(q\boldsymbol{S}_k^{-\mathrm{T}})$, the conditions of the proof algorithm can be satisfied, where $\boldsymbol{S}_k$ is the trapdoor matrix of $\Lambda(\boldsymbol{G}^\mathrm{T})$. For $s_\mathrm{r}=\sqrt{\|\boldsymbol{e}_0\|^2+m_0(\alpha q)^2}\,\omega\left(\sqrt{\log n}\right)$ (here, $\boldsymbol{e}_0\sim D_{Z,\alpha q}^{m_0}$ and $\boldsymbol{e}_1\sim D_{Z,s_\mathrm{r}}^{nk}$), based on the distance between the center and the vector obeying the discrete Gaussian distribution, $\|\boldsymbol{e}_1\|<\alpha q\sqrt{2m_0nk}\,\omega(\log n)$ and $\|\boldsymbol{e}_0\|<\alpha q\sqrt{m_0}$ both hold except in the case where the probability is $2^{-\Omega(n)}$. Then, the maximum singular value of $\boldsymbol{T}_\mathrm{r}$ satisfies $s(\boldsymbol{T}_\mathrm{r})<O\left(\sqrt{nk}\right)\omega\left(\sqrt{\log n}\right)$ with the probability of $1–2^{-\Omega(n)}$. Since $1/\alpha=O\left(\sqrt{nk}\right)\cdot\omega\left(\sqrt{\log n}\right)$ is small enough, we can obtain

$$\begin{aligned}
\left\|\boldsymbol{e}^\mathrm{T}\boldsymbol{R}_I\right\|&\le\left\|\boldsymbol{e}_0^\mathrm{T}\boldsymbol{T}_\mathrm{r}\right\|+\left\|\boldsymbol{e}_1^\mathrm{T}\right\|\\
&<\alpha qO\left(\sqrt{nk}\right)\omega\left(\sqrt{\log n}\right)\\
&<O(q)\in P_{1/2}\left(q\left(S_k^{-1}\right)^\mathrm{T}\right),
\end{aligned}\tag{9}$$

and inequality (9) holds with the probability of $1–2^{-\Omega(n)}$, where $\boldsymbol{e}=(\boldsymbol{e}_0,\boldsymbol{e}_1)$.

Finally, we provide the proof that the signature can be verified with an overwhelming probability. For this reason, we now illustrate that $(\boldsymbol{\sigma},\boldsymbol{r}_1)$ is a valid signature of μ. We need to calculate the probability of $\|\boldsymbol{\sigma}\|\le s_\mathrm{s}\sqrt{m_1}$, where

$$\boldsymbol{\sigma}=\begin{bmatrix}\boldsymbol{\sigma}_\mathrm{or}\\ \boldsymbol{y}\end{bmatrix}.\tag{10}$$

Among them, $\boldsymbol{\sigma}_\mathrm{or}$ is obtained using the SampleDG algorithm, which is an efficient preimage sampling algorithm under $\boldsymbol{G}$ trapdoor. On the basis of Theorem 2, $\boldsymbol{A}_\mathrm{s}^{(0)}\boldsymbol{\sigma}_\mathrm{or}=\boldsymbol{y}'$ and $\|\boldsymbol{\sigma}_\mathrm{or}\|\le s_\mathrm{s}\sqrt{m}$ hold with the probability of $1–2^{-\Omega(n)}$. In a similar way, $\|\boldsymbol{y}\|\le s_\mathrm{s}\sqrt{nk}$ holds with the probability of $1–2^{-\Omega(n)}$; here, $\|\boldsymbol{\sigma}\|\le s_\mathrm{s}\sqrt{m_1}$ holds with the probability of $1–2^{-\Omega(n)}$. Because

$$\begin{aligned}
\boldsymbol{A}_\mathrm{s}^{(\boldsymbol{h})}\boldsymbol{\sigma}&=\left[\boldsymbol{A}_\mathrm{s}\|\boldsymbol{A}'\right]\begin{bmatrix}\boldsymbol{\sigma}_\mathrm{or}\\ \boldsymbol{y}\end{bmatrix}\\
&=\boldsymbol{y}'+\boldsymbol{A}'\boldsymbol{y}=\boldsymbol{h}_\mathrm{r}\\
&=H_{\boldsymbol{A}_\mathrm{s}^{(\boldsymbol{h})}}(H_0(\mu,\boldsymbol{A}_\mathrm{r}),\boldsymbol{r}),
\end{aligned}\tag{11}$$

the signature holds with the probability of $1–2^{-\Omega(n)}$.

5 Security analysis

5.1 Confidentiality

Theorem 4 PQ-LBSCS has IND-CCA2 security (Yu and Wang, 2019) for the inside adversary $\mathcal{A}$ because it is hard for $\mathcal{A}$ to solve the LWE problem under $\alpha'=\alpha/3\ge 2\sqrt{n}/q$.

Proof In game G_0, challenger C runs the Setup and KeyGen algorithms. Lastly, C outputs params and the user's public key $\boldsymbol{A}$ to $\mathcal{A}$.

Phase 1: A polynomial bounded number of adaptive queries are performed by $\mathcal{A}$. C outputs a ciphertext $\boldsymbol{c}$ by a call to the blind signcryption algorithm. Then, C outputs M or $\perp$ by a call to the unsigncryption algorithm.

Challenge: A challenge query is performed by $\mathcal{A}$

for equal-length messages M_0 and M_1. C selects a random number $b\in\{0, 1\}$ to calculate a ciphertext $\boldsymbol{c}^*$ of M_b. Hereafter, C outputs $\boldsymbol{c}^*$ to the adversary.

Phase 2: Another series of adaptive queries are performed by $\mathcal{A}$ as those in phase 1. Here, $\mathcal{A}$ cannot query a signcryption of $\boldsymbol{c}^*$.

Guess: $\mathcal{A}$ outputs b' to guess b and succeeds in the game if $b'=b$. $\mathcal{A}$ wins G_0 with the advantage $\mathrm{Adv}(\mathcal{A})=\left|\Pr\left[b=b'\right]-1/2\right|$.

In game G_1, C changes the recipient's public key query along with the unsigncryption query. Then, the process of generating the recipient's public key is as follows: C selects $\boldsymbol{A}_{\mathrm{r}}^{(0)}$ and $\boldsymbol{T}_{\mathrm{r}}$ in the same way as in G_0. Assuming that $\boldsymbol{T}=\boldsymbol{T}_{\mathrm{r}}$, C selects $\boldsymbol{u}_0\leftarrow R$ to calculate $\boldsymbol{A}_{\mathrm{r}}=\left[\boldsymbol{A}_{\mathrm{r}}^{(0)}\left\|-\boldsymbol{A}_{\mathrm{r}}^{(0)}\boldsymbol{T}-\hbar(\boldsymbol{u}_0)\boldsymbol{G}\right.\right]$ and sends $\boldsymbol{A}_{\mathrm{r}}$ to $\mathcal{A}$. $\mathcal{A}$ considers $\boldsymbol{A}_{\mathrm{r}}$ as the recipient's public key. $\mathcal{A}$ requests an unsigncryption query for $(\boldsymbol{c}, \boldsymbol{A}_{\mathrm{s}}, \boldsymbol{T}_{\mathrm{s}})=((\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{c}'), \boldsymbol{A}_{\mathrm{s}}, \boldsymbol{T}_{\mathrm{s}})$. The remaining steps are run normally, except the first three steps of the unsigncryption algorithm. C executes the unsigncryption algorithm of $(\boldsymbol{u}, \boldsymbol{b})$ as follows:

Step 1: output $\perp$ and then terminate if $\boldsymbol{u}=\boldsymbol{0}$ or $\boldsymbol{u}=\boldsymbol{u}_0$, and continue otherwise.

Step 2: use $\mathrm{Invert}^{\wp}(\boldsymbol{T}_{\mathrm{r}}, \boldsymbol{A}_{\mathrm{r}}^{(\boldsymbol{u})}, \boldsymbol{b}, \hbar(\boldsymbol{u}-\boldsymbol{u}_0))$ to obtain $(\boldsymbol{z}, \boldsymbol{e})$, where $\boldsymbol{z}\in Z_q^n$ and $\boldsymbol{e}=(\boldsymbol{e}_0,\boldsymbol{e}_1)\in Z^{m_0}\times Z^{nk}$.

Step 3: output $\perp$ and then terminate if $\|\boldsymbol{e}_0\|\geq\alpha q\sqrt{m_0}$ or $\|\boldsymbol{e}_1\|\geq\alpha q\sqrt{2m_0nk}\,\omega(\sqrt{\log n})$, and continue otherwise.

Step 4: set $\boldsymbol{v}=\boldsymbol{b}-\boldsymbol{e}\bmod(2q)$ and decompose $\boldsymbol{v}=(\boldsymbol{v}_1,\boldsymbol{v}_2)\in Z_{2q}^{m_0}\times Z_{2q}^{nk}$.

Step 5: output $\perp$ and then terminate if $\boldsymbol{v}_1\notin 2\boldsymbol{\Lambda}\left((\boldsymbol{A}_{\mathrm{r}}^0)^{\mathrm{T}}\right)$, and continue otherwise.

Step 6: calculate

$$\boldsymbol{\sigma}_1=\boldsymbol{Q}^{-1}\left(\boldsymbol{v}^{\mathrm{T}}\begin{bmatrix}\hat{\boldsymbol{T}}\\ \boldsymbol{I}\end{bmatrix}\bmod(2q)\right), \tag{12}$$

where $\hat{\boldsymbol{T}}$ is the solution of $\boldsymbol{A}_{\mathrm{r}}^{(0)}\hat{\boldsymbol{T}}=\boldsymbol{A}_0\boldsymbol{T}-\hbar(\boldsymbol{u}_0)\boldsymbol{G}$.

In game G_2, C changes the method of generating ciphertext $(\boldsymbol{u}^*, \boldsymbol{b}^*, \boldsymbol{c}^*)$ along with the hash function $H_{\boldsymbol{F}}$. For $H_{\boldsymbol{F}}$, the change is as follows: without leaking the trapdoor information, C replaces the hash function $H_{\boldsymbol{F}}$ with a Chameleon hash function $H_{\boldsymbol{E}}$, where matrix $\boldsymbol{E}$ is as follows:

$$\boldsymbol{E}=\left[\boldsymbol{E}^{(0)}\left\|\boldsymbol{E}^{(1)}\right.\right]\in Z_q^{n\times l}\times Z_q^{n\times m}, \tag{13}$$

where $\boldsymbol{E}^{(0)}$(resp.$\boldsymbol{E}^{(1)}$) and $\boldsymbol{F}^{(0)}$(resp.$\boldsymbol{F}^{(1)}$) have the same distribution.

Generate the challenge ciphertext. $\mathcal{A}$ chooses two messages M_0 and M_1 ($M_0\neq M_1$) with an equal length along with the pair $(\boldsymbol{A}_{\mathrm{s}}^*, \boldsymbol{T}_{\mathrm{s}}^*)$ of public and private keys of the sender. Then, C chooses $b\in\{0, 1\}$ randomly and obtains the signature $(\boldsymbol{\sigma}, \boldsymbol{r}_1)$. Next, C sets $\boldsymbol{u}^*=\boldsymbol{u}_0$ and uses the trapdoor of $H_{\boldsymbol{E}}$ to select $\boldsymbol{r}_2$ satisfying $\boldsymbol{u}^*=H_{\boldsymbol{E}}(H_1(\boldsymbol{\sigma}), \boldsymbol{r}_2)$. Signcryption steps are the same as those in G_1.

In game G_3, C realizes the change of ciphertext $(\boldsymbol{u}^*,\boldsymbol{b}^*,\boldsymbol{c}^*)$ by changing $\boldsymbol{b}^*$. C randomly selects $\boldsymbol{s}\xleftarrow{\$}Z_q^n$ and $\boldsymbol{e}_0\leftarrow D_{Z^{m_0},\alpha q}$. C calculates

$$\boldsymbol{b}_0^{\mathrm{T}}=2(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{r}}^{(0)}\bmod q)+\boldsymbol{e}_0^{\mathrm{T}}\bmod(2q). \tag{14}$$

Hereafter, C chooses $\boldsymbol{e}\xleftarrow{\$}D_{Z^{nk},\alpha q\sqrt{m}\omega(\log n)}$ to set $\boldsymbol{b}_1^{\mathrm{T}}=\boldsymbol{b}_0^{\mathrm{T}}\boldsymbol{T}_{\mathrm{r}}+\boldsymbol{e}^{\mathrm{T}}+(\boldsymbol{Q}\sigma_1)^{\mathrm{T}}\bmod(2q)$ and $(\boldsymbol{b}^*)^{\mathrm{T}}=(\boldsymbol{b}_0^{\mathrm{T}},\boldsymbol{b}_1^{\mathrm{T}})$. The remaining steps are the same as those in G_2.

In game G_4, C changes the method of generating ciphertext $(\boldsymbol{u}^*, \boldsymbol{b}^*, \boldsymbol{c}^*)$. C randomly chooses $\boldsymbol{b}_0\xleftarrow{\$}Z_{2q}^{m_0}$, and the remaining steps are the same as those in G_3.

The security of PQ-LBSCS lies in two aspects: (1) Games G_i and G_{i+1} (i=0, 1, 2, 3) are indistinguishable; (2) The adversary in the last game G_4 has no advantage.

G_i and G_{i+1} are statistically indistinguishable; the proof is as follows:

(1) Given $\boldsymbol{u}_0\in R$ (here, R represents the finite field), $\hbar(\boldsymbol{u}_0)\in\boldsymbol{G}\log q$ is a fixed matrix. In addition, because the statistical distance between a residual hash lemma $\boldsymbol{A}_{\mathrm{r}}^{(0)}\boldsymbol{T}_{\mathrm{r}}$ and a uniform distribution can be ignored, the statistical distance between $\boldsymbol{A}_{\mathrm{r}}^{(0)}\boldsymbol{T}_{\mathrm{r}}-\hbar(\boldsymbol{u}_0)\boldsymbol{G}$ and a uniform distribution is negligible. Therefore, the value $\boldsymbol{u}_0$ is statistically hidden from the adversary; in other words, the public keys in G_0 and G_1 are statistically indistinguishable.

(2) As far as the adversary is concerned, G_2 and G_1 are statistically indistinguishable. Two matrices with the same distribution are used to construct the hash functions H_E and H_F. When the hash function is replaced, G_2 and G_1 are statistically indistinguishable. Because H_E is uniform and $\boldsymbol{u}_0$ is randomly chosen, even if the way of generating ($\boldsymbol{u}^*$, $\boldsymbol{b}^*$, $\boldsymbol{c}^*$) has been changed, the adversary still does not know $\boldsymbol{u}_0$, w, or $\boldsymbol{r}_2$; i.e., the adversary cannot distinguish between $\boldsymbol{u}^*=\boldsymbol{u}_0$ and $\boldsymbol{u}^*=H_E(w, \boldsymbol{r}_2)$.

(3) G_3 and G_2 are statistically indistinguishable. In G_3, the challenge ciphertext changes only in the public key encryption phase. In other words, only $\boldsymbol{b}=(\boldsymbol{b}_0,\boldsymbol{b}_1)\in Z_{2q}^{m_0}\times Z_{2q}^{nk}$ is different. Among them, the distribution of $\boldsymbol{b}_0$ in G_2 is the same as that in G_3. In G_2, $\boldsymbol{b}_1^{\mathrm{T}}=2(-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{r}}^{(0)}\boldsymbol{T}_{\mathrm{r}} \bmod q)+\boldsymbol{e}_1^{\mathrm{T}}+(\boldsymbol{Q}\boldsymbol{\sigma}_1)^{\mathrm{T}} \bmod(2q)$, where $\boldsymbol{e}_1\leftarrow D_{Z^{m_0},s}$ and the Gaussian parameter $s=\sqrt{\|\boldsymbol{e}_0\|^2+m_0(\alpha q)^2}\,\omega\left(\sqrt{\log n}\right)$. In G_3,

$$\begin{aligned}\boldsymbol{b}_1^{\mathrm{T}} &= -\boldsymbol{b}_0^{\mathrm{T}}\boldsymbol{T}_{\mathrm{r}}+\hat{\boldsymbol{e}}^{\mathrm{T}}+(\boldsymbol{Q}\boldsymbol{\sigma}_1)^{\mathrm{T}} \bmod(2q)\\ &= 2(-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}_{\mathrm{r}}^{(0)}\boldsymbol{T}_{\mathrm{r}} \bmod q)\\ &\quad +(\boldsymbol{e}_0^{\mathrm{T}}\boldsymbol{T}_{\mathrm{r}}+\hat{\boldsymbol{e}}^{\mathrm{T}})+(\boldsymbol{Q}\boldsymbol{\sigma}_1)^{\mathrm{T}} \bmod(2q).\end{aligned} \tag{15}$$

Now prove that the statistical distance between $\boldsymbol{e}_0^{\mathrm{T}}\boldsymbol{T}_{\mathrm{r}}+\boldsymbol{e}^{\mathrm{T}}$ and $\boldsymbol{e}_1^{\mathrm{T}}$ is negligible. Assume that $\boldsymbol{T}_{\mathrm{r}}=(\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_{nk})\in Z^{m_0\times nk}$, where $\boldsymbol{t}_i\leftarrow D_{Z^{nk},\omega\left(\sqrt{\log n}\right)}$. For $\boldsymbol{e}\leftarrow D_{Z,\alpha q\sqrt{m}\omega(\log n)}$, the statistical distance between ($\boldsymbol{e}_0$, $\boldsymbol{t}_i$)+$\boldsymbol{e}_i$ and D_{Z,s_1} is negligible (Regev, 2009). In other words, the statistical distance between $\boldsymbol{b}_1$ in G_3 and $\boldsymbol{b}_1$ in G_2 can be ignored. Therefore, ($\boldsymbol{u}^*$, $\boldsymbol{b}^*$, $\boldsymbol{c}^*$) in G_2 and G_3 are statistically indistinguishable.

(4) G_4 and G_3 are computationally indistinguishable. In G_4, if the LWE problem is hard for $\alpha'=\alpha/3\geq 2\sqrt{n}/q$, the probability that the adversary succeeds in the game is negligible. Here, we need to show the indistinguishability by discretizing LWE. $A_{s,\alpha'}\in Z_q^n\times\boldsymbol{T}$ is a non-discrete instance of LWE. $\boldsymbol{b}\rightarrow 2q\boldsymbol{b}+D_{Z-2q\boldsymbol{b},s}$ is the mapping and $(\alpha,\boldsymbol{b}=(\boldsymbol{s},\alpha)/q+\boldsymbol{e} \bmod 1)$ is an instance of O_{s}. If $\boldsymbol{s}\in Z_q^n$, O_{s} can be converted to $(\alpha,2(\boldsymbol{s},\alpha) \bmod q+\boldsymbol{e} \bmod(2q))\in Z_q^n\times Z_{2q}$. From the above mapping, a uniform instance O_{s}' on $Z_q^n\times\boldsymbol{T}$ is mapped to a uniform distribution on $Z_q^n\times Z_{2q}$.

In game G_3, $(\boldsymbol{A}_{\mathrm{r}}^{(0)},\boldsymbol{b}_0)$ is an instance of O_{s}. In game G_4, $(\boldsymbol{A}_{\mathrm{r}}^{(0)},\boldsymbol{b}_0)$ is a uniformly random instance on $Z_q^{n\times m_0}\times Z_{2q}^n$. LWE is pseudo-random when the above-mentioned discretized distribution satisfies $\alpha'=\alpha/3\geq 2\sqrt{n}/q$. Hence, G_3 and G_4 are indistinguishable under the hard assumption of discretizing LWE. Then, we analyze the advantage of $\mathcal{A}$ in G_4. In view of the residual hash lemma, when $\boldsymbol{T}_{\mathrm{r}}$ is chosen in G_4, the distance between the uniform distribution and $(\boldsymbol{A}_{\mathrm{r}}^{(0)},\boldsymbol{b}_0,\boldsymbol{A}_0\boldsymbol{T}_{\mathrm{r}},\boldsymbol{b}_0^{\mathrm{T}}\boldsymbol{T}_{\mathrm{r}})$ is negligible. As the messages are different, the statistical distance between the challenge ciphertexts is also negligible. Hence, the advantage of $\mathcal{A}$ in G_4 can be ignored.

5.2 Unforgeability

Theorem 5 PQ-LBSCS satisfies the UF-CMA security requirement (Yu and Wang, 2019) for inside adversaries because for a large enough number $\beta=O\left(\lambda(nk)^{3/2}\right)\omega\left(\sqrt{\log n}\right)^3$, it is hard to solve the SIS problem.

Proof Assume that the forger can forge a ciphertext. C runs Setup and KeyGen algorithms to obtain params and public/private key pair ($\boldsymbol{A}_{\mathrm{s}}^*$,$\boldsymbol{T}_{\mathrm{s}}^*$). C delivers $\boldsymbol{A}_{\mathrm{s}}^*$ and params to the forger. Other queries are the same as those in G_1.

In the forgery phase, the forger outputs a forged ciphertext $\boldsymbol{c}^*=(M, \boldsymbol{b}, \boldsymbol{c}')$, where the public key of the blind signcrypter is $\boldsymbol{A}_{\mathrm{s}}^*$. The attack of the forger is an inside attack, and this shows that the forger knows the recipient's private key $\boldsymbol{T}_{\mathrm{r}}^*$; because the ciphertext $\boldsymbol{c}^*$ is valid, the forger unsigncrypts ($\boldsymbol{u}$, $\boldsymbol{b}$) using $\boldsymbol{T}_{\mathrm{r}}^*$ and obtains $\boldsymbol{\sigma}_1$. Then, the forger uses $H_2(\boldsymbol{\sigma}_1, \boldsymbol{b})$ to unsigncrypt $\boldsymbol{c}'$ and obtains ($M|\boldsymbol{\sigma}_2|\boldsymbol{r}_1|\boldsymbol{r}_2$). Here, $\boldsymbol{\sigma}$ is the combination of $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$.

In the process of blind signcryption, we use the Chameleon hash function (Micciancio and Peikert, 2012). If $\beta=O\left(\lambda(nk)^{3/2}\right)\omega\left(\sqrt{\log n}\right)^3$ is large enough, it is hard to solve the SIS problem. PQ-LBSCS satisfies the UF-CMA security

requirement if the Chameleon hash function holds; in other words, $\boldsymbol{c}^*=(M, \boldsymbol{b}, \boldsymbol{c}')$ is an invalid ciphertext.

5.3 Blindness

Theorem 6 Let (μ_0, c_0) and (μ_1, c_1) denote the blinded message pair and ciphertext pair, respectively, which are obtained by the user in the blind signcryption phase. Choose μ_b and c_b, where $b\in\{0, 1\}$. If the blind signcrypter or distinguisher outputs b' in any polynomial time and the probability of $b=b'$ does not exceed $1/2+1/n^{\partial}$, where ∂ is a sufficiently large constant, it is said that (μ_0, c_0) and (μ_1, c_1) are indistinguishable for the blind signcrypter or distinguisher (Okamoto, 2006).

Proof On the basis of Lemma 1, $\boldsymbol{A}_r\boldsymbol{Z}$ in $\mu=(t^{-1}w+\boldsymbol{A}_r\boldsymbol{Z}) \bmod q$ obeys a uniform distribution over Z_q^n, where $\mu=(t^{-1}w+\boldsymbol{A}_r\boldsymbol{Z}) \bmod q$ is a blinded message received by the blind signcrypter, w is a secure hash function which approximately obeys a uniform distribution, and t is a random integer. For the blind signcrypter, μ obeys a distribution which is indistinguishable from the uniform distribution over Z_q^n. Assume that the blind signcrypter wants to recover a real message w using the random vector and t, because

$$\begin{aligned}&\Delta(t(\mu-\boldsymbol{Z}),w)\\&=\sum_{t<x}\left|p\left[t(\mu-c_1)=H_0(M)\right]-p\left[w=H_0(M)\right]\right|\\&=\sum_{t<x}\left|(1/q)^m-(1/q)^m\right|=0.\end{aligned}\tag{16}$$

This shows that the distribution of the results of such a choice is indistinguishable from the uniform distribution. Hence, PQ-LBSCS has blindness.

6 Comparison

In this section, we compare PQ-LBSCS with SZ (Sun and Zheng, 2018), YLM (Yan et al., 2019), YWM (Yan et al., 2015), and YHW (Yang et al., 2019) on the basis of communication overhead and computational efficiency in the blind signcryption and unsigncryption parts. The schemes compared are all recently proposed signcryption algorithms and the results are convincing.

The length of ciphertext, the size of params, the number of hash functions, and the size of the public key are shown in Table 2. PQ-LBSCS has a smaller ciphertext length than other schemes, and the size of params or the size of the public key of PQ-LBSCS is also smaller than those of other schemes. Moreover, in comparison with SZ, YLM, YWM, and YHW, PQ-LBSCS has anonymity. Here, the calculation cost of some hash functions is ignored in PQ-LBSCS, SZ, YLM, YWM, and YHM. However, the calculation cost of some special hash functions in PQ-LBSCS still needs to be considered.

Table 3 illustrates the computational efficiency of PQ-LBSCS, SZ, YLM, YWM, and YHM. The computational efficiency of PQ-LBSCS is higher than those of other schemes.

The computation cost shown in Table 4 is obtained using the simulation setting as follows:

CPU: Intel CORE i7;

Memory: 16 GB;

Operating system: Win10, 64-bit;

Operating platform: Matlab 2018a.

Figs. 2–4 are obtained using the Matlab tool to process the data in Table 4. Table 4 shows the operation time of the blind signcryption and unsigncryption algorithms of SZ, YLM, YWM, YHM, and PQ-LBSCS. Fig. 2 shows the time comparison for the signcryption algorithm, Fig. 3 shows the time comparison for the unsigncryption algorithm, and Fig. 4 shows the time comparison for both blind signcryption and unsigncryption algorithms.

As shown in Figs. 2–4, the time consumption of PQ-LBSCS is relatively small; moreover, the computational efficiency of PQ-LBSCS is higher than

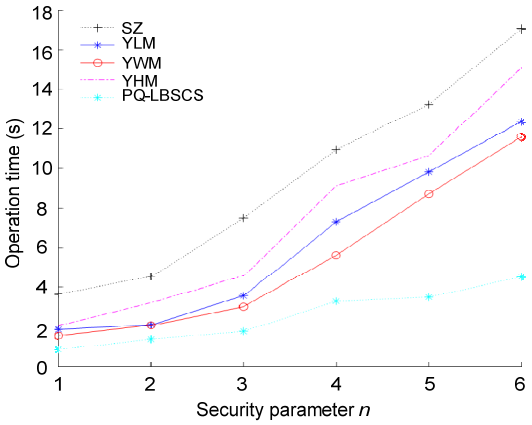

Fig. 2 Signcryption time comparison

Table 2 Size of some parameters

Scheme	Length of ciphertext	Size of params	Number of hash functions	Size of the public key
SZ	$2mk^2n(\log q)^2$	$(d+3)(\log q)^2$	2	$(mk+1)\log q$
YLM	$mkn^2(\log q)^2$	$mnk\log q$	6	$k(k+1)^2m^2\lambda(\log q)^2$
YWM	$lmn^2(\log q)^2$	$m^2nl\log q$	3	$2m^2\log q$
YHW	$2m^2n(\log q)^2$	$km^3\log q$	3	$nm^2\log q$
PQ-LBSCS	$mn^2(\log q)^2$	$2ln^2\log q$	5	$2n^2\log q$

q: a large prime number; m and k: real numbers; n: a security parameter; l: number of columns of matrix $\boldsymbol{F}$; d: number of vectors $\boldsymbol{B}_i$ in Sun and Zheng (2018)

Table 3 Computational efficiency of schemes

Scheme	Computational efficiency	
	Signcryption	Unsigncryption
SZ	PIS+5D_s+3C_s	4D_s+6C_s
YLM	PIS+5D_s+3C_s	7D_s+7C_s
YWM	PIS+5D_s+9C_s	6D_s+4C_s
YHW	PIS+4D_s+7C_s	3D_s+6C_s
PQ-LBSCS	PIS+4D_s+2C_s	2D_s+4C_s

C_s denotes the multiplication operation, D_s denotes the addition operation, and PIS represents the preimage sampling

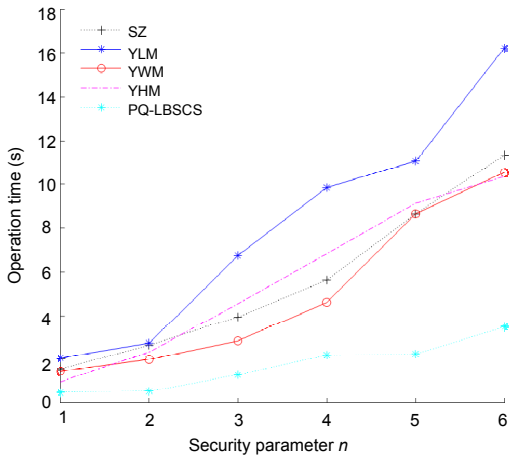

Fig. 3 Unsigncryption time comparison

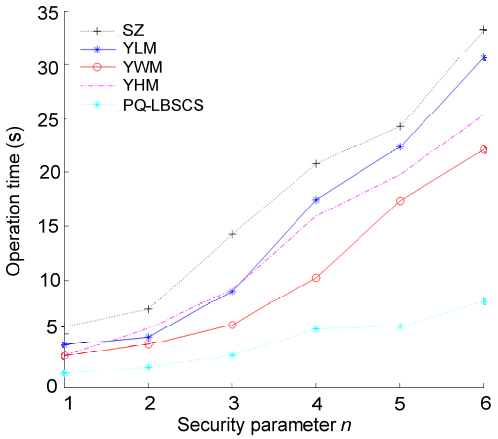

Fig. 4 Time comparison for the whole scheme

those of other schemes in the blind signcryption and unsigncryption algorithms.

7 Conclusions

In this paper, a post-quantum blind signcryption scheme (PQ-LBSCS) has been proposed, and it has been proved to meet the IND-CCA2 and UF-CMA security requirements. Analysis results have shown

Table 4 Operation time of signcryption and unsigncryption

Scheme	Algorithm	Operation time (s)					
		Security parameter=1	2	3	4	5	6
SZ	Signcryption	3.637	4.534	7.508	10.934	13.250	17.050
	Unsigncryption	2.042	2.735	6.759	9.832	11.071	16.230
YLM	Signcryption	1.832	2.064	3.547	7.320	9.830	12.450
	Unsigncryption	1.530	1.725	4.021	6.850	7.531	11.340
YWM	Signcryption	1.523	2.056	2.984	5.642	8.721	11.587
	Unsigncryption	1.537	2.645	3.954	5.642	8.634	11.347
YHM	Signcryption	2.043	3.184	4.576	9.127	10.649	15.083
	Unsigncryption	0.954	2.326	4.562	6.839	9.127	10.346
PQ-LBSCS	Signcryption	0.834	1.235	1.744	3.259	3.469	4.512
	Unsigncryption	0.454	0.522	1.257	2.180	2.229	3.476

that PQ-LBSCS has lower calculation complexity and can resist the attacks from quantum computing. PQ-LBSCS has wide application prospects in some projects. Our future plan is to study the certificateless threshold signcryption algorithm from lattice.

Contributors

Huifang YU designed the research. Huifang YU and Lu BAI processed the data. Lu BAI drafted the manuscript. Huifang YU helped organize the manuscript. Huifang YU and Lu BAI revised and finalized the paper.

Compliance with ethics guidelines

Huifang YU and Lu BAI declare that they have no conflict of interest.

References

Ajtai M, 1996. Generating hard instances of lattice problems (extended abstract). Proc 28th Annual ACM Symp on Theory of Computing, p.99-108. https://doi.org/10.1145/237814.237838

Ajtai M, Dwork C, 1997. A public-key cryptosystem with worst-case/average-case equivalence. Proc 29th Annual ACM Symp on Theory of Computing, p.284-293. https://doi.org/10.1145/258533.258604

Garg S, Gentry C, Halevi S, 2013. Candidate multilinear maps from ideal lattices. Proc 32nd Annual Int Conf on the Theory and Applications of Cryptographic Techniques, p.1-17. https://doi.org/10.1007/978-3-642-38348-9_1

Gerard F, Merckx K, 2018. Post-quantum signcryption from lattice-based signatures. *J IACR Cryptol Eprint Arch*, 9(15):56.

Hoffstein J, Pipher J, Silverman JH, 1998. NTRU: a ring-based public key cryptosystem. Proc 3rd Int Algorithmic Number Theory Symp, p.267-288. https://doi.org/10.1007/BFb0054868

Li FG, Bin Muhaya FT, Khan MK, et al., 2013. Lattice-based signcryption. *Concurr Comput Pract Exp*, 25(14):2112-2122. https://doi.org/10.1002/cpe.2826

Liu Z, Han YL, Yang XY, 2019. A signcryption scheme based learning with errors over rings without trapdoor. Proc 37th National Conf of Theoretical Computer Science, p.168-180. https://doi.org/10.1007/978-981-15-0105-0_11

Lu XH, Wen QY, Wang LC, et al., 2016. A lattice-based signcryption scheme without trapdoors. *J Electron Inform Technol*, 38(9):2287-2293 (in Chinese). https://doi.org/10.11999/JEIT151044

Micciancio D, Peikert C, 2012. Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval D, Johansson T (Eds.), Advances in Cryptology-EUROCRYPT. Springer, Berlin, Heidelberg, Germany, p.700-718. https://doi.org/10.1007/978-3-642-29011-4_41

Okamoto T, 2006. Efficient blind and partially blind signatures without random oracles. Proc 3rd Theory of Cryptography Conf, p.80-99. https://doi.org/10.1007/11681878_5

Regev O, 2009. On lattices, learning with errors, random linear codes, and cryptography. *J ACM*, 56(6):34. https://doi.org/10.1145/1568318.1568324

Sato S, Shikata J, 2018. Lattice-based signcryption without random oracles. Proc 9th Int Conf on Post-Quantum Cryptography, p.331-351. https://doi.org/10.1007/978-3-319-79063-3_16

Sun YR, Zheng WM, 2018. An identity-based ring signcryption scheme in ideal lattice. *J Netw Intell*, 3(3):152-161.

Tian HB, Zhang FG, Wei BD, 2016. A lattice-based partially blind signature. *J Secur Commun Netw*, 9(12):1820-1828. https://doi.org/10.1002/sec.1439

Yan JH, 2015. Research on Key Technologies of Lattices Signcryption. PhD Thesis, Beijing University of Posts and Telecommunications, Beijing, China (in Chinese).

Yan JH, Wang LC, Li WH, et al., 2013. Efficient lattice-based signcryption in standard model. *Math Probl Eng*, 2013:702539. https://doi.org/10.1155/2013/702539

Yan JH, Wang LC, Dong MX, et al., 2015. Identity-based signcryption from lattices. *Secur Commun Netw*, 8(18): 3751-3770. https://doi.org/10.1002/sec.1297

Yan JH, Wang LC, Li MZ, et al., 2019. Attribute-based signcryption from lattices in the standard model. *IEEE Access*, 7(1):56039-56050. https://doi.org/10.1109/ACCESS.2019.2900003

Yang XP, Cao H, Li WC, et al., 2019. Improved lattice-based signcryption in the standard model. *IEEE Access*, 7:155552-155562. https://doi.org/10.1109/ACCESS.2019.2949429

Ye Q, Zhou J, Tang YL, 2018. Partial blind signature scheme based on identity-based anti-quantum attack. *J Inform Netw Secur*, 5(3):46-53.

Yu HF, Wang ZC, 2019. Certificateless blind signcryption with low complexity. *IEEE Access*, 7:115181-115191. https://doi.org/10.1109/ACCESS.2019.2935788

Yuen TH, Wei VK, 2005. Fast and proven secure blind identity-based signcryption from pairings. Proc Cryptographers' Track at the RSA Conf, p.305-322. https://doi.org/10.1007/978-3-540-30574-3_21

Zia M, Ali R, 2019. Cryptanalysis and improvement of blind signcryption scheme based on elliptic curve. *Electron Lett*, 55(8):457-459. https://doi.org/10.1049/el.2019.0032