entropy

MDPI

Article

Practical NTRU Signcryption in the Standard Model

Jianhua Yan [1], Xiuhua Lu [2], Muzi Li [3,*], Licheng Wang [4], Jingxian Zhou [5] and Wenbin Yao [6]

[1] School of Information and Electric Engineering, Ludong University, Yantai 264025, China
[2] School of Cyber Science and Engineering, Qufu Normal University, Qufu 273165, China
[3] Archive Library, Ludong University, Yantai 264025, China
[4] School of Cyberspace Security, Beijing Institute of Technology, Beijing 100081, China
[5] Institute of Science and Technology Innovation, Civil Aviation University of China, Tianjin 300300, China
[6] School of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: mzlild@ldu.edu.cn

Abstract: Based on the NTRU trapdoor used in NIST's Falcon, a signcryption scheme following the sign-then-encrypt paradigm is constructed. The existing partitioning technique based on Waters hash over the lattice can not complete the security reduction in the standard model for the signature part due to the "partiality" of the pre-image generated with the NTRU trapdoor. To address this, a variant of Waters hash over small integers is proposed and, the probability of the successful reduction is analyzed. The resulting signcryption achieves existential unforgeability under the adaptive chosen-message attacks. By utilizing the uniqueness of the secret and the noise in an NTRU instance, the tag used in encryption is eliminated. Furthermore, a method to construct tamper-sensitive lattice public key encryption is proposed. This approach implants the ciphertext-sensitive information into the lattice public key encryption and binds it to the encrypted information. The malleability to the public key ciphertext triggers the change of the message–signature pair so that the IND-CCA2 security of the entire ciphertext can be guaranteed by the signature for the message. Thanks to the rational design and the efficiency of the NTRU trapdoor, the computational overhead of the proposed scheme is reduced significantly compared to the existing lattice-based signcryption scheme, reaching orders of magnitude improvement in efficiency. The experiment shows that the proposed scheme is efficient.

Keywords: lattice; NTRU signcryption; IND-CCA2; partitioning technique; quantum-resistant

Citation: Yan, J.; Lu, X.; Li, M.; Wang, L.; Zhou, J.; Yao, W. Practical NTRU Signcryption in the Standard Model. *Entropy* **2023**, *25*, 1651. https://doi.org/10.3390/e25121651

Received: 18 October 2023
Revised: 3 December 2023
Accepted: 6 December 2023
Published: 13 December 2023

1. Introduction

Network interaction in complex scenarios provides better services and more convenience for people to live, work, and study. The information security issues in complex scenarios have also become a thorny issue for network workers. In general, a common requirement for information security in complex scenarios is to ensure the comprehensive security of information: confidentiality, integrity, authentication, and non-repudiation. The signcryption due to Zheng [1] can simultaneously guarantee the above-integrated security at a low cost. Designing a secure signcryption scheme has become a research hotspot. Cryptographers have conducted a lot of in-depth research on signcryption based on the number theory and have achieved a series of good results. However, the rapid development of quantum computing [2] has posed a serious threat to the security foundation of traditional cryptography based on number theory. Lattice-based cryptography is becoming the backbone of quantum-resistant cryptography, due to its advantages in efficiency, flexibility, and security in the average case. It has become a common concern to construct a signcryption scheme based on the lattice.

1.1. Related Works

To resist quantum attacks, Li et al. proposed the first lattice-based signcryption [3]. After that, the lattice signcryption is developed in three directions: different security models,

advanced primitives, and specific applications. Regarding the applications, the signcryption applied to the management of health records has been studied in depth [4,5]. In terms of advanced signcryption primitives, the identity-based signcryption scheme [6], attribute-based signcryption scheme [7] (to support non-interactive fine-grained access control), and multi-recipient scheme [8] were constructed. For the security model, the schemes under the random oracle [9–13], in which [11] is the most efficient due to drawing lessons from the signcryption on Schnorr signature from [14] and the compression technique [15,16], and some schemes under the standard model were constructed.

In 2013, Yan et al. constructed a secure lattice signcryption scheme [17] under the standard model (YWW+13 [17] for short) based on the MP trapdoor [18]. This scheme follows the sign-then-encrypt (StE) paradigm, and the security of the ciphertext must be shifted to a reliance on the unforgeability of the signature with the help of the message authentication code (MAC). In fact, the MAC itself has the ability to improve IND-CCA1 security to IND-CCA2 security. Since the tag used for encryption is generated with the signature value, the lattice-based chameleon hash function is required in the process to vanish the trapdoor for completing the security reduction, which increases the computational overhead and the size of the public key. In 2018, Sato et al. made great progress in proposing a secure signcryption scheme [19] (SS18 [19] for short) under the standard model also based on the MP trapdoor. The lattice-based public key encryption (PKE) is actually an instance of learning with errors (LWE), $\mathbf{c}' = [\mathbf{A}\|\mathbf{u}]^t\mathbf{s} + \mathbf{e}$, plus the encoding of the message $[\mathbf{0}\|\mu]\lfloor q/2 \rfloor$. The malleability of the ciphertext lies in the homomorphic computational property of the LWE instance and that of the message. That is, it will affect the decryption, adding $[\mathbf{A}\|\mathbf{u}]^t\mathbf{s}'$ to the LWE instance, appending a small error $\mathbf{e}'$ to the LWE instance or superimposing additional information to the message. To eliminate the malleability of the ciphertext, the LWE instance is signed along with the original message in SS18 [19]. Meanwhile, the homomorphic malleability to the plaintext code will of course break the signatures. In this sense, the scheme belongs to the encrypt-then-sign (EtS) paradigm instead of the StE paradigm, as they claim. In 2019, Yang et al. constructed a signcryption scheme [20] (YCL+19 [20] for short) under the standard model based on ring learning with errors (RLWE) [21]. YCL+19 uses the key exchange [22,23] rather than the public key encryption to generate the key for the symmetric encryption, which reduces the size of the public key encryption. The hint information of the lattice-based key exchange is naturally immune to tampering due to its sensitivity to key recovery. However, it incurs a security risk to expose another part of the key exchange. Liu et al. proposed an NTRU-based signcryption [24] (LTTM19 [24] for short) by adopting NTRU-based key encapsulation [25] and an NTRU-based signature [26]. However, the unsigncryption queries can not be implemented in the security reduction under the standard model, so the scheme is not IND-CCA2 secure as they claimed. In the sign-then-encrypt (StE) paradigm, the signature seems more natural than that in encrypt-then-sign (EtS) paradigm since it is signed only for the message. Moreover, the construction of signcryption under the StE paradigm is more concerned with cryptographers [27,28]. It is also a problem of great importance to design a secure lattice-based signcryption scheme following the StE paradigm.

To address the quantum threat to cryptography based on number theory, in 2017, the National Institute of Standards and Technology (NIST) began collecting the post-quantum public key cryptography algorithms through an open, competitive process. The post-quantum cryptographic (PQC) algorithm should meet the following five requirements: secure under the existing computing conditions and quantum computers, fast operation, reasonable communication overhead, can be used as a direct replacement for the existing algorithms and protocols, and broad application scenarios. After many rounds of rigorous screening, NIST announced the screening results of the third round of post-quantum cryptographic algorithm standardization in July 2022. In the four post-quantum algorithms, there are two lattice-based signature algorithms, CRYSTALS-Dilithium [29] and Falcon [30], and they are more efficient than the hash-based signature. In fact, in May 2022, the scientists from Google published the latest research results [31] in the journal "Nature" to illustrate

the importance of post-quantum cryptography (PQC) and appeal to transition to PQC. Thus, it is a natural question: ***Can an efficient signcryption scheme following the StE paradigm be designed based on the NIST standard, which is secure in the standard model and does not require MAC transferring?***

1.2. Proposed Design

To resist quantum attacks, we construct a signcryption scheme based on NTRU, referred to as SC-NTRU. Our contribution can be summarized in two main points. First, we introduce an approach to improve the security of the encryption segment using the signature segment for the messages in the signcryption. The signature security can be appropriately decreased compared with that in the EtS paradigm. Second, we construct a new abort-resistant hash to adapt to the approximate pre-image scenario, and utilize it to build an NTRU signature secure in the standard model. The reasonable design and efficient trapdoor of SC-NTRU lead to a significant reduction in computational overhead, surpassing existing lattice-based signcryption methods by several orders of magnitude.

We have developed a method to achieve IND-CCA2 security in signcryption by combining three techniques. Firstly, we leverage the uniqueness of the secret and noise used in lattice-based encryption to transform tag-based encryption into general encryption. Secondly, we embed sensitive information related to the ciphertext itself into the ciphertext, binding it to the information to be encrypted using public key encryption (PKE). As the entire encryption is a hybrid encryption, the plaintext hidden in PKE serves as a key for symmetric encryption. Any modification to the ciphertext will consequently modify the key of the symmetric encryption due to their interdependence. Thirdly, we exploit the one-to-one property of symmetric encryption, such that a modified public key ciphertext will produce a modified message–signature pair. Subsequently, the unforgeability of the signature can be utilized to check the malleability and enhance the IND-CCA2 security of the complete ciphertext. It is important to note that the signature here does not need to achieve strong unforgeability while the strong unforgeability for a signature is necessary in the general construction of the IND-CCA2 secure encryption scheme. Since the message–signature pair here is encrypted and concealed from potential adversaries, any attempt to forge a signature would result in a new signature. In summary, the requirement for the signature to enhance encryption to IND-CCA2 security is diminished. Even a strong forgery of the signature supplies no help to unsigncrypting.

A common approach to constructing a secure lattice signature scheme in the standard model is through a partitioning technique based on Waters hash [32]. In [33], this hash function takes the form $H(\nu_0, \nu_1, \cdots, \nu_{\aleph}) = \sum_{i=0}^{\aleph}(-1)^{\nu_i} p_i$, with $p_i \in \mathbb{Z}_q$ for $i = 0, 1, \cdots, \aleph$; it is also referred to as an abort-resistant hash function. The probability of the hash not aborting is demonstrated using the concept of the hyperplane in [34]. However, we find that this hash proposed in [34] can not help us complete the security reduction for the signature component involved in the signcrytion. The pre-image generated by the NTRU trapdoor exhibits a certain level of "partiality". The entire NTRU trapdoor does not fall into the category of approximate trapdoors, such as that constructed by Chen [35] based on [18], and the pre-image generated by NTRU trapdoor can be exact in its entirety. However, the checkout polynomial only operates on a subset of the pre-image, which is reflected in its form $\mathbf{s_1} + \mathbf{s_2} * \mathbf{h_f} = \mathbf{0}$ (refer to Algorithm 2). In other words, the pre-image corresponding to the checkout polynomial is merely an approximate pre-image, as there exists a small error vector $\mathbf{x}' = \mathbf{y} - \mathbf{h_f}\mathbf{x}$. The range of the existing abort-resistant hash is $\mathbb{Z}_q$. When the hash value operates on the checkout polynomial, the product of the hash value and the short error vector x' results in a vector close to the uniform distribution over $\mathbb{Z}_q^n$. Consequently, this product vector makes it impossible to simulate the signature in security reduction. To address this issue, we modify the hash range to a space with a small value. However, this modification leads to a significant increase in the abort probability, which hinders secure reduction. To overcome this issue, we introduce a new random variable that cyclically selects random numbers when the abort condition is met. This helps to avoid

premature abandonment. Subsequently, we need to evaluate the probability of successfully completing reduction when an adversary forges a signature. However, this evaluation is not trivial since the hyperplane model for the abort-resistant hash defined over a ring ($\mathbb{Z}_q$) is inadequate for the hash over small integers.

2. Preliminaries

In this paper, the notations are as follows. $\mathbb{Z}$: the set of integers; $\mathbb{Z}^+$: the set of positive integers; $\mathcal{R}$: the ring $\mathcal{R} := \mathbb{Z}[x]/(1+x^n)$; for a prime q, $\mathcal{R}_q := \mathcal{R}/q$; the bold lowercase letter: polynomial or the vector composed of the coefficients of the polynomial; the bold uppercase letter: matrix; $\widetilde{\mathbf{B}}$: Gram–Schmidt orthogonalization of the matrix $\mathbf{B}$; $\|\mathbf{x}\|$: the two-norm of a vector named $\mathbf{x}$; $\|\mathbf{X}\|$: the maximum of the column vectors, $\|\mathbf{X}\| = max_i\{\|\mathbf{X}_i\|\}$.

2.1. NTRU Lattice and Hard Problem

Definition 1 (NTRU Lattice). *Let $\mathcal{R} := \mathbb{Z}[x]/(1+x^n)$ for some power-of-two integer n. Let $\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \mod q$ for $\mathbf{f}, \mathbf{g} \in \mathcal{R}$ and $q \in \mathbb{Z}^+$. The corresponding NTRU lattice to $\mathbf{h}, \mathbf{f}$ is defined as*

$$\Lambda_{\mathbf{h},q} = \{(\mathbf{u}, \mathbf{v}) \in \mathcal{R}^2 | \mathbf{u} + \mathbf{v} * \mathbf{h} = \mathbf{0} \mod q\}.$$

$\Lambda_{\mathbf{h},q}$ is a full-rank lattice over $\mathbb{Z}^{2n}$ linearly spanned by the row vectors of $\mathbf{A}_{\mathbf{h},q} = \begin{pmatrix} -A_N(\mathbf{h}) & \mathbf{I_n} \\ q\mathbf{I_n} & \mathbf{O_n} \end{pmatrix}$, where $A_N(\mathbf{h})$ denotes the anti-circulant matrix generated by the vector $\mathbf{h}$.

Definition 2 (Decisional Small Polynomial Ratio: DSPR [36]). *Let $\mathcal{R} := \mathbb{Z}[x]/(1+x^n)$, $\mathcal{R}_q = \mathcal{R}/q$. For $\mathbf{g}, \mathbf{f} \in \mathcal{R}$ with small coefficients and $\mathbf{f}$ invertible over $\mathcal{R}_q$, the distinguishing problem between the distribution of $\mathbf{h} = \mathbf{g}\mathbf{f}^{-1} \mod q$ and that of $\mathbf{h}' \xleftarrow{\$} \mathcal{R}_q$ is defined as the decisional small polynomial ratio problem.*

The hardness of the search version of DSPR has been studied in [37].

Definition 3 (Search Learning with Errors in a Ring of Integers). *Let Ψ be a family of distributions over $K_{\mathbb{R}}$ and $2 < q \in \mathbb{Z}$. The RLWE problem $RLWE_{q,\Psi}$ is to find $s \in R_q^\vee$ by allowing access to arbitrarily many samples from $A_{s,\Psi}$ for $\psi \in \Psi$.*

Proposition 1 (Hardness of RLWE [21]). *Let K denote an arbitrary number field with degree n. Let $\mathbb{Z} \ni q = q(n) \geq 2$ and arbitrary $\alpha = \alpha(n) \in (0,1)$ satisfying $\alpha q \geq \omega(\sqrt{\log n})$. A probabilistic polynomial time quantum reduction can be constructed from K-DGS$_\gamma$ to $\mathcal{O}_K$-LWE$_{q,\Psi_{\leq\alpha}}$, where $\gamma = \eta_\epsilon(\mathbf{I}) \cdot \omega(\sqrt{\log n})/\alpha$.*

2.2. Trapdoor Generation and Pre-Image Sample Algorithm

Proposition 2 (NTRU Key Generation). *Inputting dimension n and modulus q, there is an efficient keyGen algorithm to output public key $\mathbf{h}$ and the trapdoor $\mathbf{B} = \begin{pmatrix} A_N(\mathbf{g}) & -A_N(\mathbf{f}) \\ A_N(\mathbf{g}') & -A_N(\mathbf{f}') \end{pmatrix}$, such that $\mathbf{h}$ is computationally indistinguishable from the uniform distribution over $\mathcal{R}$, $\|\widetilde{\mathbf{B}}\| \leq 1.17\sqrt{q}$, and $\mathbf{g}, \mathbf{f} \leftarrow D_{\mathbb{Z}^n,\eta}$ for $\eta \leq 1.17\sqrt{q/(2n)}$.*

Proposition 3 (Pre-Image Sampling [38]). *Let $\epsilon = 2^{-\lambda}/(2n)$ for arbitrary $n, \lambda \in \mathbb{Z}^+$, Δ denote statistical distance. For any basis $\mathbf{B} \in \mathbb{Z}^{n\times n}$ of $\Lambda_{\mathbf{A},q}$ and arbitrary syndrome vector $\mathbf{u} \in \mathbb{Z}^n$, there is a pre-image sampling algorithm $\mathcal{GS}$, $\mathbf{x} = \mathcal{GS}(\mathbf{B}, \eta, \mathbf{u})$, such that $\mathbf{A}\mathbf{x} = \mathbf{u} \mod q$ and $\Delta(D_{\Lambda(\mathbf{B},\eta,\mathbf{u})}, \mathbf{x}) \leq 2^{-\lambda}$, when $\eta \geq \|\tilde{\mathbf{B}}\| \cdot \zeta'_\epsilon(\mathbb{Z})$ and $\zeta'_\epsilon(\mathbb{Z}) \approx \frac{1}{\pi}\sqrt{\frac{1}{2}ln\left(2+\frac{2}{\epsilon}\right)}$. ϵ can be relaxed to $2^{-\lambda/2}/(4\sqrt{2n})$, and $\zeta'_\epsilon(\mathbb{Z}) \approx \frac{1}{\sqrt{2\pi}}\sqrt{\ln 2^{(\lambda+7)/2}n}$.*

Proposition 4 (Discrete Gaussian Distribution). *Let Λ be an m-dimension lattice. Let real s be the Gauss parameter and $\mathbf{c} \in \mathbb{R}^m$ be the center. The discrete Gauss distribution has the following good properties.*

1. *(Lemma 4.4 of [39]) For any positive real k, $Pr[|x| > ks; x \xleftarrow{\$} D_{\mathbb{Z},s}] \leq 2e^{-k^2/2}$.*
2. *(Lemma 4.4 of [39]) When $s > 3/\sqrt{2\pi}$, $D^m_{\mathbb{Z},s}(\mathbf{x}) < 2^{-m}$ for all vector $\mathbf{x} \in \mathbb{Z}^m$.*
3. *(Lemma 4.4 of [39]) For any positive real k, $Pr[\|\mathbf{x}\| > ks\sqrt{m}; \mathbf{x} \xleftarrow{\$} D^m_{\mathbb{Z},s}] \leq k^m e^{(1-k^2)m/2}$.*
4. *(Lemma 4.3 of [39]) Let $t \in \mathbb{R}^+$, $\mathbf{v} \in \mathbb{R}^m$. Then $Pr[\langle \mathbf{z}, \mathbf{v} \rangle > t] \leq 2e^{-t^2/(2s^2\|\mathbf{v}\|^2)}$.*
5. *(Lemma 4.4 of [40]) $\underset{\mathbf{x} \sim D_{\Lambda,s,\mathbf{c}}}{Pr}[\|\mathbf{x} - \mathbf{c}\| \geq s\sqrt{m}] \leq \frac{1-\epsilon}{1+\epsilon} 2^{-m}$.*
6. *(Lemma 5.2 and Corollary 5.4 of [41]) Let n, m be integers, s real and q prime. When $s \geq \omega(\sqrt{\log m})$, $s \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ for $\epsilon \in (0, 1/2)$. When $m \geq 2n \log q$, $s \geq \omega(\sqrt{\log m})$, with $1 - 2q^{-n}$ probability, the syndrome $\mathbf{y} = \mathbf{A}\mathbf{x}$ is statistically close to uniform over $\mathbb{Z}_q^n$ for all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $x \xleftarrow{\$} D^m_{\mathbb{Z},s}$.*
7. *(Lemma 2.4 of [42] or lemma 4.4 [40]).*

$$\rho_s(\Lambda + c) \in \left[\frac{1-\epsilon}{1+\epsilon}, 1\right] \cdot \rho_s(\Lambda).$$

3. Signcryption: Syntax and Security Models

In this section, the syntax and security models of signcryption are presented.

3.1. Syntax of Signcryption

A signcryption scheme is composed of the following four algorithms:

- **Setup** (λ): This algorithm takes a security parameter λ as input, then returns the public parameter PP.
- **KeyGen** (λ, PP): Input the security parameter λ and the public parameter PP, and output the public key and private key pairs (PK, SK) for users.
- **SignCrypt** (μ, PK_s, SK_s, PK_r): Take a message μ, the sender's public key PK_s and private key SK_s, and the recipient's public key PK_r, generate a ciphertext c.
- **UnSignCrypt** (c, PK_s, SK_r): Inputting a ciphertext c, the sender's public key PK_s and the recipient's private key SK_r, this algorithm unsigncrypts the ciphertext and verifies the signature. If the signature can pass the verification, it returns the message μ, otherwise it returns $\perp$.

3.2. Security Models of Signcryption

To clarify the confidence of the signcryption, an **IND-CCA2** game is introduced (Table 1).

- Initial: The challenger $\mathcal{C}$ runs the setup and key generation algorithms to generate public parameters PP, the receiver's keys (Pk_r, SK_r), and the sender's keys (Pk_s, SK_s), followed by giving (PP, Pk_r, Pk_s, SK_s) to an adversary $\mathcal{A}$.
- Phase 1: The adversary $\mathcal{A}$ implements the unsigncryption queries in an adaptive manner bounded by polynomial times. If c is a valid ciphertext, $\mathcal{C}$ replies with the corresponding plaintext μ, otherwise it returns $\perp$.
- Challenge: $\mathcal{A}$ selects two plaintexts μ_0, μ_1 with equal length and gives them to $\mathcal{C}$. $\mathcal{C}$ tosses a fair coin $b \in \{0, 1\}$ and generates a challenge ciphertext $c^* = \text{signcrypt}(PK_s, SK_s, PK_r, \mu_b)$ followed by giving c^* to $\mathcal{A}$.
- Phase 2: $\mathcal{A}$ continues to perform unsigncryption queries as in Phase 1, except for not permitting to query unsigncryption on c^*.
- Guess: $\mathcal{A}$ gives b^* as the guess on b tossed by $\mathcal{C}$.

Then, the advantage of $\mathcal{A}$ to win **Game IND-CCA2** is defined as $Adv(\mathcal{A}) = |\Pr[b = b'] - \frac{1}{2}|$.

Table 1. Game IND-CCA2 between $\mathcal{C}$ and $\mathcal{A}$.

	$\mathcal{C}$	Communication	$\mathcal{A}$
Intial.	$\{(pk_s, sk_s), (pk_r, sk_r)\} \leftarrow \text{KeyGen}(1^k)$	$\xrightarrow{pk_s, pk_r, sk_s}$	
Phase 1.	$\mu = \text{UnSigncrypt}(c, pk_r, pk_s, sk_r)$	$\xleftarrow[\text{for Unsigncrypt}]{c}$ $\xrightarrow[\text{as reply}]{\mu}$	choose c
	...	...	...
Challenge	Toss a coin b, $c^* = \text{Signcrypt}(\mu_b, pk_r, pk_s, sk_s)$	$\xleftarrow[\text{for challenge}]{\mu_0, \mu_1}$ $\xrightarrow[\text{as reply}]{c^*}$	choose μ_0, μ_1 with equal length
Phase 2.	repeat Phase 1, except reply $\perp$ to the query for c^*	repeat	repeat
Guess		$\xleftarrow[\text{as reply}]{b'}$	guess b'

Definition 4 (Confidentiality of Signcryption)**.** *A signcryption scheme is said to be indistinguishable against inner choose ciphertext attacks (IND-CCA2) if there exists no probabilistic polynomial time inner adversary who can win* ***Game IND-CCA2*** *with a non-negligible advantage.*

To capture the unforgeability of the signcryption, an **EUF-CMA game** is introduced (Table 2).

- Initial: The challenger $\mathcal{C}$ runs the setup and key generation algorithms to generate public parameters, the keys for the receiver and sender are as in the **IND-CCA2 Game**. Subsequently, $\mathcal{C}$ gives (PP, Pk_s, Pk_r, SK_r) to $\mathcal{A}$.
- Signcrypt: $\mathcal{A}$ chooses messages μ and implements polynomially-bounded signcryption queries by an adaptive approach. $\mathcal{C}$ replies with the corresponding ciphertexts c.
- Forge: $\mathcal{A}$ outputs c^*, which contains a new signature for some μ that has not been previously queried.

Table 2. Game EUF-CMA between $\mathcal{C}$ and $\mathcal{A}$.

	$\mathcal{C}$	Communication	$\mathcal{A}$
Initial.	$\{(pk_s, sk_s), (pk_r, sk_r)\} \leftarrow \text{KeyGen}(1^k)$	$\xrightarrow{pk_s, pk_r, sk_r}$	
Queries	$c = \text{SC}(\mu, pk_r, pk_s, sk_s)$	$\xleftarrow[\text{for Signcrypt}]{\mu}$ $\xrightarrow[\text{as reply}]{c}$	choose μ
	...	...	...
Forgery		$\xleftarrow[\text{as reply}]{c^*}$	generate c^*

Then, the advantage of $\mathcal{A}$ to win **Game EUF-CMA** is defined as $Adv(\mathcal{A}) = \Pr[(\mu, \sigma) = \textbf{Unsigncrypt}(c^*) \text{ and } \mathfrak{N}]$, where $\mathfrak{N}$ denotes the fact that σ is a valid signature for μ not discoved by the unsigncryption queries.

Definition 5 (Existential Unforgeability of Signcryption)**.** *A signcryption scheme is said to be existentially unforgeable against inner chosen message attacks (EUF-CMA) if no probabilistic polynomial time forgery can win* ***Game UF-CMA*** *with a non-negligible advantage.*

4. Signcryption Based on NTRU

In this section, a signcryption scheme based on NTRU is proposed, followed by its correctness and parameter settings.

4.1. Construction

The symmetric encryption ENC involved in the proposed scheme is IND-OT secure and one-to-one. That is, for a plaintext μ and symmetric key k, there is only one ciphertext c satisfying $\mathrm{DEC}_k(c) = \mu$.

- Setup(1^λ): On inputting a security parameter 1^λ, generate the public parameters and hash functions.
  1. Choose hash functions: $H_0 : \{0,1\}^* \to \{0,1\}^{\aleph}$, $H_1 : R_q \to R_q$, $H_2 : R_q \times R_q \to \{0,1,2,3\}^n$.
  2. $\mathbf{y}, \mathbf{u} \xleftarrow{\$} R_q$.
  3. $\mathbf{z_i} \xleftarrow{\$} R_q$ for $i = 0, 1, \cdots, \aleph$.
  4. publish public parameters $(\mathbf{y}, \mathbf{u}, \{\mathbf{z_i}\}_{i=1}^{\aleph})$.
- KeyGen(1^ℓ): User i generates it own public key and private key.
  1. $(\mathbf{B}, \mathbf{h}) \leftarrow \mathrm{KeyGen}(n, q)$ to satisfy $\mathbf{B}\begin{pmatrix}\mathbf{i}\\ \mathbf{h}\end{pmatrix} = \mathbf{0}$ where $\mathbf{B} = \begin{pmatrix} A(\mathbf{g}) & -A(\mathbf{f}) \\ A(\mathbf{G}) & -A(\mathbf{F}) \end{pmatrix} \in \mathbb{Z}^{2n\times 2n}$, The coefficient vector of $\mathbf{i}$ is $(1, 0, 0, \cdots, 0)^t$.
  2. $\mathbf{b}, \mathbf{d} \xleftarrow{\$} R_q$.
  3. Publish $(\mathbf{b}, \mathbf{d}, \mathbf{h})$ as public key and keep $\mathbf{B}$ as private key. Namely, the sender's (resp. receiver's) public and private key are $((\mathbf{b_s}, \mathbf{d_s}, \mathbf{h_s}), \mathbf{B_s})$ (resp. $((\mathbf{b_r}, \mathbf{d_r}, \mathbf{h_r}), \mathbf{B_r})$).
- SignCrypt$(\mu, \mathbf{h_r}, \mathbf{B_s}, \{\mathbf{z_i}\}_{i=0}^{\aleph})$.
  1. $\mathbf{k} \xleftarrow{\$} \{0,1,2,3\}^n$.
  2. $\nu := H_0(\mu, \mathbf{b_r}, \mathbf{k})$.
  3. $\mathbf{z} := \mathbf{z_0} + \sum_{i=1}^{\aleph}(-1)^{\nu_i}\mathbf{z_i}$.
  4. $\mathbf{x_1} \leftarrow D_{\mathbb{Z}^n,\eta_1}$.
  5. $\mathbf{t} := \mathbf{y} - \mathbf{z}\mathbf{x_1}$.
  6. $(\mathbf{x_0'}, \mathbf{x_0}) := (\mathbf{t}, \mathbf{0}) - \mathrm{SamplePre}(\mathbf{B_s}, \eta_1, (\mathbf{t}, \mathbf{0}))$.
  7. $\mathbf{x} := (\mathbf{x_0}, \mathbf{x_1})^t$.
  8. $\mathbf{r} \xleftarrow{\$} \{-1,0,1\}^n$, $\mathbf{e_0}, \mathbf{e_2} \xleftarrow{\$} \{-\varrho, \cdots, \varrho\}^n$, $\mathbf{e_1} \xleftarrow{\$} ([-\varrho\sqrt{n}, \varrho\sqrt{n}] \cap \mathbb{Z})^n$.
  9. $\mathbf{c_0} := \mathbf{r}\mathbf{h_r} + \mathbf{e_0}$.
  10. $\mathbf{c_1} := \mathbf{r}(\mathbf{b_r} + H_1(\mathbf{c_0})\mathbf{d_r}) + \mathbf{e_1}$.
  11. $\mathbf{c_2'} := \lfloor(\mathbf{r}\mathbf{u} + \mathbf{e_2})/2^\ell\rfloor$.
  12. $\mathbf{c_2} := \mathbf{c_2'} + \lfloor((H_2(\mathbf{c_1}, \mathbf{c_2'}) \oplus \mathbf{k})\lfloor q/8\rfloor)/2^\ell\rfloor$.
  13. $c_3 := \mathtt{Enc}_{H_3(\mathbf{k})}(\mathbf{x}, \mu)$.
  14. output $(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}, c_3)$.
- UnSignCrypt$(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}, c_3, \mathbf{B_r}, \mathbf{h_s}, \{\mathbf{z_i}\}_{i=0}^{\aleph})$.
  1. $\mathbf{s_1} \leftarrow D_{\mathbb{Z}^n,\eta_2}$.
  2. $\mathbf{t} := \mathbf{u} - \mathbf{s_1}(\mathbf{b_r} + H_1(\mathbf{c_0})\mathbf{d_r})$.
  3. $(\mathbf{s_0'}, \mathbf{s_0}) = (\mathbf{t}, \mathbf{0}) - \mathrm{SamplePre}(\mathbf{B_r}, \eta_2, (\mathbf{t}, \mathbf{0}))$.
  4. $\mathbf{w} := \mathbf{c_2} * 2^\ell - \mathbf{s_1}\mathbf{c_1} - \mathbf{s_0}\mathbf{c_0}$.
  5. $\tilde{\mathbf{k}} := \lfloor\frac{\mathbf{w}}{\lfloor q/8\rfloor}\rceil$.
  6. $\mathbf{c_2'} = \mathbf{c_2} - \lfloor\tilde{\mathbf{k}}\lfloor q/8\rfloor/2^\ell\rfloor$.
  7. $\mathbf{k} = \tilde{\mathbf{k}} \oplus H_2(\mathbf{c_1}, \mathbf{c_2'})$.
  8. $(\mathbf{x}, \mu) := \mathtt{Dec}_{H_3(\mathbf{k})}(c_3)$.
  9. $\nu := H_0(\mu, \mathbf{b_r}, \mathbf{k})$.
  10. $\mathbf{z} := \mathbf{z_0} + \sum_{i=1}^{\aleph}(-1)^{\nu_i}\mathbf{z_i}$.
  11. If $\|(\mathbf{h_s}, \mathbf{z})\mathbf{x} - \mathbf{y}\| \le \frac{1}{2\pi^3}\aleph n^{5/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}$ return μ, otherwise return $\perp$.

4.2. Correctness

In the proposed scheme, we draw lessons from [43] to use the ciphertext generated in the previous step (i.e., $\mathbf{c_0}$) as the tag to produce the subsequent ciphertext. On the one hand, it can transform a tag encryption to a normal encryption. On the other hand, when $\mathbf{c_0}$ is determined, the rest of the ciphertexts are relatively determined except for the influence of errors and the value to be encrypted, which is important for the security reduction (reference to Theorem 1). We give the unique witness for NTRU as follows.

Lemma 1 (Unique Witness). *Let* $\varrho = 20\log n$, $q = \frac{5}{\pi^3}\varrho\aleph n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}$ *for all but a negligible fraction of* $\mathbf{h} \in \mathcal{R}$ *and any syndrome* $\mathbf{c} \in \mathcal{R}$ *be the number of the pair* $(\mathbf{r}, \mathbf{e}) \in \{-1, 0, 1\}^n \times \{-\varrho, \cdots, \varrho\}^n$ *to satisfy that* $\mathbf{c} = \mathbf{rh} + \mathbf{e}$ *is no more than one.*

Proof. For each $\mathbf{c} \in \mathcal{R}$, suppose that there exist two different $(\mathbf{r}, \mathbf{e}) \neq (\mathbf{r}', \mathbf{e}') \in \{-1, 0, 1\}^n \times \{-1, 0, 1\}^n$ satisfying $\mathbf{r} + \mathbf{he} = \mathbf{r}' + \mathbf{he}' = \mathbf{c}$. Then, $(\mathbf{r} - \mathbf{r}')\mathbf{h} = \mathbf{e}' - \mathbf{e} \in \{-2\varrho, \cdots, 2\varrho\}^n$. To complete this lemma, it only needs to argue that $\|\tilde{\mathbf{r}}\mathbf{h}\|_\infty > 4\varrho + 1$ with overwhelming probability, for $\tilde{\mathbf{r}} \in \{-2, \cdots, 2\}^n$. Let $\pounds$ denote the n-dimension open ℓ_∞ cube with the radius of each dimension being $4\varrho + 1$ (edge length $8\varrho + 2$). Then, for any fix $\tilde{\mathbf{r}} \in \{-2, \cdots, 2\}^n$, the probability that $\tilde{\mathbf{r}}\mathbf{h}$ falls into some cubic $\pounds$ is $(8\varrho + 2)^n/q^n$. According to the union bound, for all $\tilde{\mathbf{r}} \in \{-2, \cdots, 2\}^n$, the probability of $\tilde{\mathbf{r}}\mathbf{h}$ falling into some $\pounds$ is no more than

$$5^n\left(\frac{8\varrho+2}{q}\right)^n = \left(\frac{40\varrho+10}{q}\right)^n < \left(\frac{41\varrho}{\frac{5}{\pi^3}\varrho\aleph n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}}\right)^n = \left(\frac{41\pi^3}{5\aleph n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}}\right)^n,$$

$$5^n\left(\frac{8\varrho+2}{q}\right)^n = \left(\frac{40\varrho+10}{q}\right)^n < \left(\frac{41\varrho}{\frac{6\sqrt{2}}{\pi^3}\varrho\aleph n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}}\right)^n = \left(\frac{41\pi^3}{6\sqrt{2}\aleph n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}}\right)^n,$$

which is negligible. That is, for all but a negligible fraction of $\mathbf{h}$, $\|\tilde{\mathbf{r}}\mathbf{h}\|_\infty > 4\varrho + 1$. □

By the way, in the scheme, the converted ciphertext with each dimensional error less than $\frac{4}{\sqrt{2}\pi^2}\varrho n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}$, the unique witness also holds. The corresponding probability is $(\frac{8}{\sqrt{2}\pi^2}\varrho n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}/q)^n 5^n \approx (5/36)^n$, which is negligible.

In order to guarantee the correctness and security of the proposed scheme, the following requirements should be satisfied.

- The RLWE should be hard $\alpha q \geq \omega(\sqrt{\log n})$ (Proposition 1).
- The pre-image sample algorithm works well, $\eta \geq \|\tilde{\mathbf{B}}\| \cdot \zeta'_\epsilon(\mathbb{Z})$ for $\zeta'_\epsilon(\mathbb{Z}) \approx \frac{1}{\pi}\sqrt{\frac{1}{2}ln\left(2 + \frac{2}{\epsilon}\right)}$ and $\epsilon = 2^{-\lambda/2}/(4\sqrt{2}n)$ [38], where $\mathbf{B}$ is the corresponding basis of a lattice (Proposition 3).
- In the security reduction, a simulated trapdoor (Algorithm 1) can be used for sampling (Proposition 3).
- The correctness of the unsigncryption requires that both the error in the public key encryption and the error in the signature are small enough to guarantee security.

Therefore, the Gaussian parameter and the modulus can be set as follows. η_2: Gaussian parameter for decryption; η_1: Gaussian parameter for signature. The correctness of the proposed scheme is implied in Lemmas 11 and 16.

$$\begin{aligned} &\eta_1 = \frac{1}{2\pi^3}\aleph n^2(\ln(2^{(7+\lambda)/2}n))^{3/2}, \quad \eta_2 = \frac{1}{\sqrt{2}\pi^2}n(\ln(2^{(7+\lambda)/2}n))^{3/2}, \quad \varrho = 10(\log n)^{3/4}, \\ &q = \frac{6\sqrt{2}}{\pi^3}\varrho\aleph n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}, \quad \eta_3 = \frac{1}{\sqrt{2}\pi^2}\sqrt{n}\ln(2^{(7+\lambda)/2}n) \end{aligned} \tag{1}$$

5. Security and Performance

The confidentiality and unforgeability of SC-NTRU are demonstrated in Section 5.1. The efficiency of the SC-NTRU is evaluated by analyzing the numbers of different kinds of computations in Section 5.2. An experiment for SC-NTRU is presented in Section 5.3.

5.1. Security

The IND-CCA2 security of the proposed scheme is based on a variant of the search RLWE implied in [38]. We formalize it as follows.

Definition 6 (Variant of Search RLWE). *Let ϱ, t be small integers, $\mathbf{s} \in R_q$ with $\|\mathbf{s}\|_\infty \leq \varrho$. Let ℓ denote a positive integer, and Ψ be a family of distributions over $K_{\mathbb{R}}$. For $i \in [\ell+1]$, $\mathbf{e_i} \leftarrow \Psi$, $\mathbf{a_i} \leftarrow R_q$. The search RLWE is to find $\mathbf{k} \in [2^{t-1}]^n$ from $\mathbf{c_i} = \mathbf{a_i s} + \mathbf{e_i}$ for $i \in [\ell]$ and $\mathbf{c}_{\ell+\mathbf{1}} = \mathbf{a}_{\ell+\mathbf{1}}\mathbf{s} + \mathbf{e}_{\ell+\mathbf{1}} + \mathbf{k}\lfloor q/2^t \rfloor$.*

Remark 1. *The variant is as hard as the standard search LWE. Suppose there exists an adversary $\mathcal{A}$ who can find the correct $\mathbf{k}$. Then, $\mathcal{A}$ can compute $\mathbf{c}'_{\ell+\mathbf{1}} = \mathbf{a}_{\ell+\mathbf{1}}\mathbf{s} + \mathbf{e}_{\ell+\mathbf{1}}$ due to $\mathbf{k}\lfloor q/2^t \rfloor$ hidden by $\mathbf{c}'_{\ell+\mathbf{1}}$. That is, $\mathcal{A}$ has the ability to compute $\mathbf{c}'_{\ell+\mathbf{1}}$ from $\{\mathbf{c_i}\}_{i=1}^{\ell}$. One method is that $\mathcal{A}$ learns $\mathbf{r}$ from $\{\mathbf{c_i}\}_{i=1}^{\ell}$, then $\mathcal{A}$ uses the same $\mathbf{r}$ to get $\mathbf{k}$ from $\mathbf{c}_{\ell+\mathbf{1}}$. This means that the variant of the search RLWE can be reduced to the standard search RLWE in polynomial time. The other method is that $\mathcal{A}$ can find the mapping from $\{\mathbf{a_i}\}_{i=1}^{\ell}$ to $\mathbf{a}_{\ell+\mathbf{1}}$. The mapping must have the form as $\mathbf{a}_{\ell+\mathbf{1}} = \sum_{i=1}^{\ell} \mathbf{x_i a_i} + \mathbf{y}$, where $\mathbf{x_i}, \mathbf{y}$ are sampled from R_q with small coefficients. Otherwise, $\mathbf{a}_{\ell+\mathbf{1}} = \sum_{i=1}^{\ell} \mathbf{x_i a_i}^{\kappa_i} + \mathbf{y}$, $1 \neq \kappa_i \in \mathbb{Z}$. When the mapping is applied to $\{\mathbf{c_i}\}_{i=1}^{\ell}$, the error will increase almost to the uniform distribution, which leads to failing to obtain $\mathbf{k}$ from $\mathbf{c}_{\ell+\mathbf{1}}$. In fact, it is also a search RLWE problem to find the mapping $\mathbf{a}_{\ell+\mathbf{1}} = \sum_{i=1}^{\ell} \mathbf{x_i a_i}^{\kappa_i} + \mathbf{y}$.*

Theorem 1 (IND-CCA2). *Under the parameter settings as in Equation (1), the proposed signcryption scheme is IND-CCA2 secure against inner adversaries in the standard model, as long as the $RLWE_{n,m,q}$ problem and DSPR problem are computationally intractable.*

Proof. The theorem can be proven by a series of games G_0, G_1, $\cdots$, G_{13}. In each game, the adversary $\mathcal{A}$'s probability of success is $Pr[A_i]$ for $i \in \{0, 1, \cdots, 13\}$. G_0 is the real IND-CCA2 security game. In $G_0 \cdots G_{10}$, the challenge ciphertexts are all hybrid encryption ciphertexts. It is proven that the successive games satisfy the indistinguishability or the game transitions based on failure events to $\mathcal{A}$. Thus, the difference between attacking in G_0 and attacking in G_{10} is guaranteed to be negligible. Due to the security of the symmetric encryption involved, the ciphertext of the symmetric encryption c_3 does not reveal any information about the plaintext and the corresponding signature.

Only using symmetric encryption, a direct attack on the message–signature pair is no less difficult than an attack on the symmetric key. According to the unforgeability of the signature and the security of the symmetric encryption, it is proven that it is impossible to manipulate c_3 to obtain a valid ciphertext to help the attack. Therefore, ignoring the ciphertext c_3 in the challenge ciphertext of G_{11}, the transformation does not increase the hardness of the adversary's attack. Following that, it is shown that the game transitions based on failure events are satisfied from game G_{11} to G_{13}. In G_{13}, the challenge ciphertext is an RLWE instance, and the probability that the adversary succeeds to obtain the information encrypted by the public key is negligible. Thus, the $\mathcal{A}$'s success probability to attack in G_0 is also negligible. This means that the proposed scheme is IND-CCA2 secure. The games are as follows.

- G_0: The game G_0 is the original IND-CCA2 game, namely, $\mathbf{h_r} \leftarrow \mathbf{KeyGen}, \mathbf{b_r}, \mathbf{d_r} \leftarrow \mathbb{Z}_q^n$.
  - Setup: Choose hash functions: $H_0 : \{0,1\}^* \rightarrow \{0,1\}^{\aleph}$, $H_1 : R_q \rightarrow R_q$, $H_2 : R_q \times R_q \rightarrow \{0,1,2,3\}^*$, and public parameters $\mathbf{y}, \mathbf{u} \xleftarrow{\$} R_q$.
  - KeyGen: Generate the public and private keys: $(\mathbf{B_s}, \mathbf{h_s}) \leftarrow \text{KeyGen}(n, q)$, $(\mathbf{B_r}, \mathbf{h_r}) \leftarrow \text{KeyGen}(n, q)$, $\mathbf{b_s}, \mathbf{d_s} \xleftarrow{\$} R_q, \mathbf{z_i} \xleftarrow{\$} R_q$ for $i = 0, 1, \cdots, \aleph$. Publish $(\mathbf{b_s}, \mathbf{d_s}, \mathbf{h_s}, \mathbf{h_r})$ as the public key and keep $\mathbf{B_s}$ and $\mathbf{B_r}$ as private keys.
  - Phase I: Upon receiving an unsigncryption query from $\mathcal{A}$, $\mathcal{C}$ uses the private key $\mathbf{B_r}$ to unsigncrypt as in the proposed scheme. If the signature satisfies the

constraint $\|(\mathbf{h_s}, \mathbf{z})\mathbf{x} - \mathbf{y}\| \leq \frac{1}{2\pi^3}\aleph n^{5/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}$, $\mathcal{C}$ returns the message μ, otherwise it returns $\perp$.

- Challenge: After a polynomial round of interaction with $\mathcal{C}$, $\mathcal{A}$ gives a satisfied signal to $\mathcal{C}$. $\mathcal{C}$ randomly chooses a message μ and generates the challenge ciphertext $\mathbf{c}^* = (\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*}, \mathbf{c_3^*})$ according to the steps of sincryption in the proposed scheme, then gives $\mathbf{c}^*$ to $\mathcal{A}$.
- Phase II: If the ciphertext for unsigncryption from $\mathcal{A}$ is $\mathbf{c}^*$, $\mathcal{C}$ replies with $\perp$ directly. Otherwise, $\mathcal{C}$ unsigncrypts the querying ciphertext as in Phase I.

- G_1: In the game G_1, only the producing method of $\mathbf{b_r}$ is changed. Let $\mathbf{b_r} = \mathbf{h_r p} + \mathbf{e}$ where $\mathbf{p} \leftarrow \{-1, 0, 1\}^n, \mathbf{e} \leftarrow \{-\varrho, \cdots, \varrho\}^n$. Since $\mathcal{C}$ has the normal private key, the unsigncryption approach is the same as in G_0.
- G_2: Before publishing the public keys, $\mathcal{C}$ generates a challenge ciphertext $\mathbf{c}^* = (\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*}, \mathbf{c_3^*})$ normally, namely $\mathbf{c_0^*} = \mathbf{r h_r} + \mathbf{e_0}$, $\mathbf{c_1^*} = \mathbf{r}(\mathbf{b_r} + H_1(\mathbf{c_0^*})\mathbf{d_r}) + \mathbf{e_1}$, $\mathbf{c_2'}$: $= \lfloor(\mathbf{ru} + \mathbf{e_2})/2^\ell\rfloor$, $\mathbf{c_2^*}$: $= \mathbf{c_2'} + \lfloor((H_2(\mathbf{c_1^*}, \mathbf{c_2'}) \oplus \mathbf{k})\lfloor q/8\rfloor)/2^\ell\rfloor$, c_3^*: $= \mathtt{Enc}_{H_3(\mathbf{k})}(\mathbf{x}, \mu)$ for $\mathbf{r} \xleftarrow{\$} \{-1, 0, 1\}^n, \mathbf{e_0}, \mathbf{e_2} \xleftarrow{\$} \{-\varrho, \cdots, \varrho\}^n, \mathbf{e_1} \xleftarrow{\$} ([-\varrho\sqrt{n}, \varrho\sqrt{n}] \bigcap \mathbb{Z})^n$.
- G_3: In the game G_3, $\mathcal{C}$ continues changing $\mathbf{b_r}$ as $\mathbf{b_r} = \mathbf{h_r p} + \mathbf{e} - H_0(\mathbf{c_0^*})\mathbf{d_r}$ where the method to generate $\mathbf{p}, \mathbf{e}$ is identical to that in G_2.
- G_4: The game G_4 is the same as G_3, except for the approach to produce $\mathbf{d_r}$. $\mathcal{C}$ generates $\mathbf{d_r}$ by KeyGen algorithm, $\mathbf{d_r} \leftarrow \mathbf{KeyGen}$, instead of $\mathbf{d_r} \xleftarrow{\$} R_q$.
- G_5: $\mathcal{C}$ answers with $\perp$ to the unsigncryption queries with the kind of ciphertext $(\mathbf{c_0^*}, \mathbf{c_1}, \mathbf{c_2}, c_3)$ in phase I. The others remain the same as in the game G_5.
- G_6: In this game, in phase I, $\mathcal{C}$ answers all the unsigncryption queries normally, but $\mathcal{C}$ replies with $\perp$ to the queries with ciphertext $(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2})$ satisfying $\mathbf{c_0} \neq \mathbf{c_0^*}$ but $H_1(\mathbf{c_0}) = H_1(\mathbf{c_0^*})$.
- G_7: In phase II, $\mathcal{C}$ answers with $\perp$ to unsigncryption queries with the kind of ciphertext $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*}, c_3)$ meeting $c_3 \neq \mathbf{c_3^*}$. Except for the cases above, $\mathcal{C}$ responds to the unsigncryption queries as in G_6.
- G_8: In phase II, $\mathcal{C}$ responds with $\perp$ to the unsigncryption queries with the form $(\mathbf{c_0^*}, \mathbf{c_1}, \mathbf{c_2}, c_3)$, satisfying $(\mathbf{c_1}, \mathbf{c_2}) \neq (\mathbf{c_1^*}, \mathbf{c_2^*})$. Replying to the other unsigncryption queries keeps the same as in G_7.
- G_9: In this game, $\mathcal{C}$ produces $\mathbf{h_r} \leftarrow \mathbb{Z}_q^n$ instead of $\mathbf{h_r} \leftarrow$ KeyGen. Moreover, $\mathcal{C}$ will use $\mathbf{d_r}$ to answer the unsigncryption queries. The others are identical to the game G_8.
- G_{10}: In this game, $\mathcal{C}$ produces the challenge ciphertext by collaborating with a signer, which for convenience will be called signature oracle $\mathcal{O}_{sign}$. First, $\mathcal{C}$ generates $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^{*\prime}})$ normally as in G_9, followed by giving $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^{*\prime}})$ to $\mathcal{O}_{sign}$. Then, $\mathcal{O}_{sign}$ randomly chooses a message μ and $\mathbf{k} \xleftarrow{\$} \{0, 1, 2, 3\}^n$, and produces a signature $(\mathbf{x}, \mathbf{k})$ for μ. Next, $\mathcal{O}_{sign}$ generates $c_3^* = \mathtt{Enc}_{\mathbf{k}}(\mu, \mathbf{x})$ and gives it to $\mathcal{C}$. Lastly, $\mathcal{C}$ gives $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*}, c_3^*)$ to $\mathcal{A}$ as the challenge ciphertext and waits to get $(\mu, \mathbf{k}, \mathbf{x})$ from $\mathcal{A}$.
- G_{11}: $\mathcal{C}$ changes the challenge ciphertext a little. $\mathcal{C}$ does not give $\mathbf{k}$ to $\mathcal{O}_{sign}$ and does not need to get c_3 from $\mathcal{O}_{sign}$. $\mathcal{C}$ just gives the ciphertext $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*})$ generated by itself to $\mathcal{A}$ as the challenge ciphertext. If $\mathcal{A}$ is able to obtain the plaintext $\mathbf{k}$ in $\mathbf{c_2^*}$ with non-negligible probability, $\mathcal{C}$ admits that $\mathcal{A}$ wins the game with the same probability.
- G_{12}: This game is identical with the game G_{11} except that the challenge ciphertext $\mathbf{c_1^*}$ is computed as $\mathbf{c_1^*} = \mathbf{c_0^* p} + \mathbf{e_1}$ where $\mathbf{e_1} \xleftarrow{\$} ([-\varrho\sqrt{n}, \varrho\sqrt{n}] \bigcap \mathbb{Z})^n$.
- G_{13}: $\mathcal{C}$ queries the variant RLWE oracle to fetch an instance $\left(\begin{pmatrix}\mathbf{a_1}\\ \mathbf{a_2}\end{pmatrix}, \begin{pmatrix}\mathbf{z_1}\\ \mathbf{z_2}\end{pmatrix}\right)$ Then, $\mathcal{C}$ publishes the public keys as $\mathbf{h_r} = \mathbf{a_1}, \mathbf{u} = \mathbf{a_2}$. $\mathcal{C}$ and sets the challenge ciphertext as follows $\mathbf{c_0^*} = \mathbf{z_1}, \mathbf{c_2^*} = \mathbf{z_2}$. The construction method for $\mathbf{c_1^*}$ remains identical with that in G_{12}, that is $\mathbf{c_1^*} = \mathbf{c_0^* p} + \mathbf{e_1}$.

□

Lemma 2. *The games G_1 and G_0 are computationally indistinguishable when $\varrho = 20 \log n$.*

Proof. This lemma can be proven by hybrids. First, $\mathcal{C}$ chooses $\mathbf{h}' \xleftarrow{\$} R_q$ and calculates $\mathbf{b}'_{\mathbf{r}} = \mathbf{h}'\mathbf{p} + \mathbf{e}$ where $\mathbf{p} \xleftarrow{\$} \{-1, 0, 1\}^n, \mathbf{e} \xleftarrow{\$} \{-\varrho, \cdots, \varrho\}^n$. The infinity norm of the error $\mathbf{e}$ is set at $20 \log n$, which satisfies the constraint for errors in RLWE, i.e., the Gaussian parameter $\alpha q \geq \omega(\sqrt{\log n})$ (see Proposition 1). According to the intractability of RLWE, $\mathbf{b}'_{\mathbf{r}}$ and $\mathbf{b}_{\mathbf{r}} \xleftarrow{\$} R_q$ are computationally indistinguishable. Next, $\mathcal{C}$ continues modifying $\mathbf{b}'_{\mathbf{r}}$ as $\mathbf{b}'_{\mathbf{r}} = \mathbf{h}_{\mathbf{r}}\mathbf{p} + \mathbf{e}$. Since $\mathbf{h}_{\mathbf{r}}$ and $\mathbf{h}'$ are statistically indistinguishable according to the uniform property of the algorithm KeyGen, the new $\mathbf{b}'_{\mathbf{r}}$ is computationally indistinguishable with $\mathbf{b}_{\mathbf{r}} \xleftarrow{\$} R_q$. Given all this, the games G_1 and G_0 are computationally indistinguishable. □

Lemma 3. *The games G_2 and G_1 are identical from the view of $\mathcal{A}$.*

Proof. Since the computation procedure for $(\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*)$ is completely hidden from $\mathcal{A}$. In the view of $\mathcal{A}$, it makes no difference whether or not some challenge ciphertext has been generated before the public key is published. That is, the difference between G_2 and G_1 is only the difference in concept. Consequently, the lemma holds. □

Lemma 4. *The game G_3 is statistically indistinguishable with the game G_2.*

Proof. The only difference between the games G_3 and G_2 is $\mathbf{b}_{\mathbf{r}}$. In G_3, $\mathcal{C}$ calculates $\mathbf{b}_{\mathbf{r}} = \mathbf{h}_{\mathbf{r}}\mathbf{p} + \mathbf{e} - H_0(\mathbf{c}_0^*)\mathbf{d}_{\mathbf{r}}$, while in G_2 $\mathbf{b}_{\mathbf{r}} = \mathbf{h}_{\mathbf{r}}\mathbf{p} + \mathbf{e}$. Since $\mathbf{c}_0^*$ is fixed and $\mathbf{d}_{\mathbf{r}}$ submits to the uniform distribution. Therefore, the $\mathbf{b}_{\mathbf{r}}$ is statistically indistinguishable with the $\mathbf{b}_{\mathbf{r}}$ in the game G_2. □

Due to the uniform property of the public keys generated by the KeyGen algorithm, we have the following lemma.

Lemma 5. *The game G_4 is statistically indistinguishable with the game G_3.*

It is difficult to directly prove the statistical indistinguishability between the games G_5 and G_4. However, the game sequences can be continued by the game transitions based on failure events.

Lemma 6. *Let E_5 denote the event that $\mathcal{A}$ makes the unsigncryption queries with the ciphertexts $(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}, c_3)$ meeting $\mathbf{c_0} = \mathbf{c}_0^*$ in phase I. From the point of view of $\mathcal{A}$, the games G_5 and G_4 are totally the same, when E_5 does not occur. Furthermore, $Pr[A_5|\neg \mathrm{E}_5] = Pr[A_4|\neg \mathrm{E}_5]$, and $Pr[\mathrm{E}_5]$ is negligible.*

Proof. First, in the game G_5, $\mathcal{C}$ may reply to all the unsigncryption queries normally except for the queried ciphertexts $(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}, c_3)$ with $\mathbf{c_0} = \mathbf{c}_0^*$. Therefore, the behavior of $\mathcal{C}$ is identical in games G_5 and G_4, when E_5 does not happen. That is, $\mathcal{A}$ learns exactly the same knowledge form $\mathcal{C}$, when not considering the event E_5.

Second, in the games G_5 and G_4, $\mathbf{c}_0^*$ is hidden from $\mathcal{A}$ in phase I. Whether $\mathbf{c}_0^*$ is obtained by evaluating $\mathbf{rh_r} + \mathbf{e_0}$ or by guessing, the probability that $\mathcal{A}$ computes a ciphertext with the form $(\mathbf{c}_0^*, \mathbf{c_1}, \mathbf{c_2}, c_3)$ is q^{-n}. That is to say, the probability that $\mathcal{C}$ can not correctly answer the valid unsigncryption queries is no more than q^{-n}, which is negligible. The difference is that $\mathcal{C}$ deliberately answers the query E_5 with $\perp$ in G_5 instead of unsigncrypting normally as in G_4. In summary, the reduction for the attack capability learned in G_5 compared to that in G_4 is negligible. □

Lemma 7. *Let E_6 denote the events that $\mathcal{A}$ makes the unsigncryption queries with the ciphertexts $(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}, c_3)$ meeting $\mathbf{c_0} \neq \mathbf{c}_0^*$ but $H_1(\mathbf{c_0}) = H_1(\mathbf{c}_0^*)$ in phase I. The games G_6 and G_5 are identical in the adversary $\mathcal{A}$'s view, when E_6 does not happen, $Pr[A_6|\neg \mathrm{E}_6] = Pr[A_5|\neg \mathrm{E}_6]$. Moreover, $Pr[\mathrm{E}_6]$ is negligible.*

Proof. First, in G_6, the approach of $\mathcal{C}$ to answer the unsigncryption queries is totally identical when the queried ciphertexts $(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}, c_3)$ satisfy $\mathbf{c_0} \neq \mathbf{c_0^*}$ and $H_1(\mathbf{c_0}) = H_1(\mathbf{c_0^*})$. Therefore, the knowledge learned from G_6 and G_5 is the same, when not considering the event E_6.

Second, for a known $\mathbf{c_0^*}$, $\mathcal{A}$ finds a $\mathbf{c_0}$ satisfying $\mathbf{c_0} \neq \mathbf{c_0^*}$ but $H_0(\mathbf{c_0}) = H_0(\mathbf{c_0^*})$, which means $\mathcal{A}$ discovers a hash collision. We set the hash function to satisfy the security intensity of the proposed signcryption system, namely, the probability that hash collision occurs is no more than a negligible probability $2^{-\lambda}$. In fact, $\mathbf{c_0^*}$ is hidden in phase I, then the probability that $\mathcal{A}$ just computes a ciphertext with $\mathbf{c_0} \neq \mathbf{c_0^*}$ and $H_0(\mathbf{c_0}) = H_0(\mathbf{c_0^*})$ is also no more than the above probability $2^{-\lambda}$. Thus, even though the attack power of $\mathcal{A}$ obtained in G_6 is less than that in G_5, the difference is negligible. □

Lemma 8. *Let E_7 denote the events that $\mathcal{A}$ set the unsigncryption queries with the type of ciphertext $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*}, c_3)$ meeting $c_3 \neq c_3^*$ in phase II. In the adversary $\mathcal{A}$'s view, the games G_7 and G_6 are identical, when E_7 does not happen. Namely, $Pr[A_7|\neg \mathrm{E}_7] = Pr[A_6|\neg \mathrm{E}_7]$. Furthermore, $Pr[\mathrm{E}_7]$ is negligible.*

Proof. The ciphertext elements $(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2})$ constitute the ciphertext of the public key encryption. Once the elements are determined, the plaintext $\mathbf{k}$ involved in the public key encryption is determined and unique. That is, the key used for the symmetric encryption is fixed. According to the one-to-one property of the symmetric encryption ENC, modifying c_3 will yield a distinct message–signature pair $(\mathbf{x}', \mu') = \mathtt{Dec}_{\mathbf{k}}(c_3)$ (relative to $(\mathbf{x}, \mu)$). The probability of the new message–signature pair successfully passing the signature verification is negligible. We prove this conclusion through a classification discussion. It can be divided into two cases.

- **Case 1**: $\mathcal{A}$ generates the ciphertext c_3 by its signature for some message μ. As an inner adversary, $\mathcal{A}$ has the ability to yield signatures by itself. However, the procedure to generate c_3 requires $\mathbf{k}^*$. The probability of obtaining $\mathbf{k}^*$ from $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*})$ is negligible due to the security of the public key encryption part. Please refer to Lemma 15 for more details.
- **Case 2**: $\mathcal{A}$ generates the ciphertext c_3 randomly. According to the one-to-one property of the symmetric encryption, a random message–signature pair $(\mu, \mathbf{x})$ is unsigncrypted from c_3. Since $\mathcal{A}$ does not possess knowledge of μ, it cannot generate a valid signature for it despite having the private key for signature generation. Consequently, the probability $(\mu, \mathbf{k}, \mathbf{x})$ passing signature verification is negligible.

□

Lemma 9. *Let E_8 denote the event that in phase II, $\mathcal{A}$ makes the unsigncryption queries with the form of the ciphertext as $(\mathbf{c_0^*}, \mathbf{c_1}, \mathbf{c_2}, c_3)$ and $(\mathbf{c_1}, \mathbf{c_2}) \neq (\mathbf{c_1^*}, \mathbf{c_2^*})$. In the adversary $\mathcal{A}$'s view, the games G_8 and G_7 are identical, when E_8 does not occur. Namely, $Pr[A_8|\neg \mathrm{E}_8] = Pr[A_7|\neg \mathrm{E}_8]$. Moreover, $Pr[\mathrm{E}_8]$ is negligible.*

Proof. This lemma can be argued by a classification discussion. Let $\mathbf{k}, \mathbf{k}^*$ be the keys concealed in $\mathbf{c_2}$, $\mathbf{c_2^*}$, respectively.

- **Case 1**: $\mathbf{k}^*$ remains unchanged, i.e., $\mathbf{k} = \mathbf{k}^*$. To guarantee the validity of the ciphertext, the information hidden by $\mathbf{c_2}$ should be $\lfloor (H(\mathbf{c_1}, \mathbf{c_2'}) \oplus \mathbf{k}^*) \lfloor q/8 \rfloor / 2^{\ell} \rfloor$. To fulfill this requirement, there are only two possible subcases.
  - $\mathcal{A}$ generates $\mathbf{c_2}$ through encryption. On one hand, $\mathcal{A}$ does not know the $\mathbf{k}^*$ concealed in $\mathbf{c_2^*}$. On the other hand, $\mathcal{A}$ chooses a $\mathbf{r}$ distinct from the secret $\mathbf{r}$ used in the $\mathbf{c_0^*}$, which will lead to an invalid public key ciphertext. Thus, the probability of this subcase occurring is negligible.
  - $\mathcal{A}$ produces $\mathbf{c_2}$ by falsifying $\mathbf{c_2^*}$. For this, $\mathcal{A}$ should have the ability to compute $\mathbf{c_2} - \mathbf{c_2^*} + \lfloor ((H(\mathbf{c_1}, \mathbf{c_2'}) \oplus \mathbf{k}^*) - (H(\mathbf{c_1^*}, \mathbf{c_2^{*\prime}}) \oplus \mathbf{k}^*)) \lfloor q/8 \rfloor / 2^{\ell} \rfloor$. The probability of this event is negligible since $\mathbf{c_2^{*\prime}}$ and $\mathbf{k}$ involved in $\mathbf{c_2^*}$ are hidden from $\mathcal{A}$.

- **Case 2**: $\mathbf{k} \neq \mathbf{k}^*$ and c_3 remains unchanged. In this case, due to the one-to-one property of the symmetric encryption, the message–signature pair $(\mu, \mathbf{k}, \mathbf{x})$ extracted from c_3 is completely random. As a result, the probability of $(\mu, \mathbf{k}, \mathbf{x})$ passing signature verification is negligible.
- **Case 3**: $\mathbf{k} \neq \mathbf{k}^*$ and $c_3 \neq c_3^*$. The case can be divided into two subcases.
  - In the procedure to generate c_3, the plaintext μ corresponding to it is not known to $\mathcal{A}$. Consequently, the probability $(\mathbf{c_1^*}, \mathbf{c_2}, \mathbf{c_3})$ being a valid ciphertext is negligible. The argument is the same as that of case 2 in the proof of Lemma 8.
  - The ciphertext c_3 is generated based on a valid message–signature pair produced by $\mathcal{A}$. It can be further divided into two subcases. (1) $\mathbf{c_2}$ is obtained though encrypting by $\mathcal{A}$. The argument to this subcase is identical to subcase 1 of case 1 in this Lemma 9. Due to the uniqueness of LWE (see Lemma 1), $\mathbf{r}$ has already been determined by $\mathbf{c_0^*}$. The probability of $\mathcal{A}$ choosing $\mathbf{r}$ used in $\mathbf{c_0^*}$ is negligible. The the distinct $\mathbf{r}$ yields an invalid ciphertext. (2) $\mathbf{c_2}$ is falsified from $\mathbf{c_2^*}$ by $\mathcal{A}$. This is similar to the demonstration in subcase 2 of case 1 of Lemma 9. $\mathcal{A}$ needs to compute $\mathbf{c_2} - \mathbf{c_2^*} + \lfloor((H(\mathbf{c_1}, \mathbf{c_2'}) \oplus \mathbf{k}) - (H(\mathbf{c_1^*}, \mathbf{c_2^{*\prime}}) \oplus \mathbf{k}^*))\lfloor q/8 \rfloor / 2^{\ell} \rfloor$ without knowing $\mathbf{k}^*$, and the probability of this event is negligible.

□

Algorithm 1 SimExtractE($\mathbf{h_r}, \mathbf{b_r}, \mathbf{d_r}, \tau, \tau^*, \mathbf{u}$)

Require:
public keys $\mathbf{h_r}, \mathbf{b_r}, \mathbf{d_r} \in R_q$, in which $\mathbf{d_r} \leftarrow$ `KeyGen`, $\mathbf{b_r} = \mathbf{h_r p} + \mathbf{e}$ for $\mathbf{p}, \mathbf{e} \leftarrow \{-\varrho, \cdots, \varrho\}^n$;
a pair of un-equivalent tags $\tau, \tau^* \leftarrow \{-\varrho, \cdots, \varrho\}^n$;
a syndrome $\mathbf{u} \in \mathcal{R}_q$.
Ensure:
A private key $(\mathbf{x_1}, \mathbf{x_2})$.

1: $\mathbf{x_0'} \leftarrow \{-1, 0, 1\}^n$
2: $\mathbf{x'} = (\mathbf{u} - \mathbf{h_r x_0'})/(\tau - \tau^*)$
3: $(\mathbf{x_1'}, \mathbf{x_1}) = (\mathbf{x'}, \mathbf{0}) - \text{SamplePre}(\mathbf{B_d}, \eta_2, (\mathbf{x'}, \mathbf{0}))$ $(\eta_2 = \frac{1}{\sqrt{2\pi^2}} n (\ln (2^{(7+\lambda)/2} n))^{3/2})$
4: $\mathbf{x_0} = \mathbf{x_0'} - \mathbf{p x_1}$
5: output $(\mathbf{x_0}, \mathbf{x_1})$ as the private key.

Lemma 10. *Under the parameter settings as in Equation (1), the* $(\mathbf{x_0}, \mathbf{x_1})$ *generated in Algorithm 1 is a valid private key. That is,* $\|\mathbf{u} - [\mathbf{h_r} \| \mathbf{b_r} + \tau \mathbf{d_r}](\mathbf{x_0}, \mathbf{x_1})\| \leq \frac{3}{\sqrt{2\pi^2}} \varrho n^{5/2} (\ln (2^{(7+\lambda)/2} n))^{3/2}$ *and* $\|(\mathbf{x_0}, \mathbf{x_1})\| \leq \frac{1}{\sqrt{2\pi^2}} \varrho n^{5/2} (\ln (2^{(7+\lambda)/2} n))^{3/2}$. *It ensures that this private key can be used correctly in unsigncryption.*

Proof. First, we argue that the multiplication of $[\mathbf{h_r} \| \mathbf{b_r} + \tau \mathbf{d_r}]$ and $(\mathbf{x_0}, \mathbf{x_1})$ is close enough to $\mathbf{u}$.

$$\begin{aligned}
[\mathbf{h_r} \| \mathbf{b_r} + \tau \mathbf{d_r}](\mathbf{x_0}, \mathbf{x_1}) &= \mathbf{h_r x_0} + (\mathbf{h_r p} + \mathbf{e} - \tau^* \mathbf{d_r} + \tau \mathbf{d_r}) \mathbf{x_1} \\
&= \mathbf{h_r}(\mathbf{x_0} + \mathbf{p x_1}) + \mathbf{e x_1} + (\tau - \tau^*) \mathbf{d_r x_1} \\
&= \mathbf{h_r x_0'} + \mathbf{e x_1} + (\tau - \tau^*)(\mathbf{x'} - \mathbf{x_1'}) \\
&= \mathbf{h_r x_0'} + \mathbf{e x_1} + (\mathbf{u} - \mathbf{h_r x_0'}) - (\tau - \tau^*) \mathbf{x_1'} \\
&= \mathbf{u} + \mathbf{e x_1} - (\tau - \tau^*) \mathbf{x_1'}
\end{aligned} \tag{2}$$

$$\begin{aligned}\|\mathbf{u}-[\mathbf{h_r}\|\mathbf{b_r}+\tau\mathbf{d_r}](\mathbf{x_0},\mathbf{x_1})\| =&\|(\tau-\tau^*)\mathbf{x_1'}-\mathbf{e}\mathbf{x_1}\| \le \|\mathbf{e}\mathbf{x_1}\|+\|(\tau-\tau^*)\mathbf{x_1'}\|\\ \le&\|\mathbf{e}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\sqrt{n}+\|\boldsymbol{\omega}-\boldsymbol{\omega}^*\|_\infty\|\mathbf{x_1'}\|\sqrt{n}\sqrt{n}\\ =&\varrho\eta_2\sqrt{n}n+2\eta_2\sqrt{n}n \le (\varrho+2)\frac{1}{\sqrt{2}\pi^2}n(\ln(2^{(7+\lambda)/2}n))^{3/2}n^{3/2}\\ =&\frac{1}{\sqrt{2}\pi^2}(\varrho+2)n^{5/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}.\end{aligned} \tag{3}$$

Second, we demonstrate that $(\mathbf{x_0},\mathbf{x_1})$ is short.

$$\begin{aligned}\|(\mathbf{x_0},\mathbf{x_1})\| =&\sqrt{\|\mathbf{x_0}\|^2+\|\mathbf{x_1}\|^2} \le \sqrt{\|\mathbf{x_0'}\|^2+\|\mathbf{p}\mathbf{x_1}\|^2+\|\mathbf{x_1}\|^2}\\ \approx&\|\mathbf{p}\mathbf{x_1}\| \le \|\mathbf{p}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\sqrt{n} \approx \varrho\eta_2\sqrt{n}n\\ \le&\frac{1}{\sqrt{2}\pi^2}n\varrho(\ln(2^{(7+\lambda)/2}n))^{3/2}n^{3/2}\\ =&\frac{1}{\sqrt{2}\pi^2}\varrho n^{5/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}\end{aligned} \tag{4}$$

Here, $\mathbf{x_0}$ takes up most of the length of the whole vector $[\mathbf{x_0},\mathbf{x_1}]$. That is,

$$\|\mathbf{x_0}\| \le \frac{1}{\sqrt{2}\pi^2}\varrho n^{5/2}(\ln(2^{(7+\lambda)/2}n))^{3/2},\ \ \|\mathbf{x_1}\| \le \frac{1}{\sqrt{2}\pi^2}n^{3/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}. \tag{5}$$

From the above, the private key and the error are of the same magnitude, differing only by a constant factor. In fact, we can use Lemma 3 of [43] to reduce the length of the private key and the error by the factor $\sqrt{n}$, so that the modulus q can also be reduced by $\sqrt{n}$. □

Lemma 11. *In the adversary $\mathcal{A}$'s view, the games G_9 and G_8 are totally identical. Furthermore, $Pr[F_9] = Pr[F_8]$.*

Proof. First, the public keys in the games G_9 and G_8 are identical. Second, $\mathcal{C}$ does not reply to the unsigncryption queries for the ciphertexts with $\mathbf{c_0} \neq \mathbf{c_0^*}$ in the games G_9 and G_8. Third, we argue that $\mathcal{C}$ can correctly unsigncrypt with the private key produced in Algorithm 1, when $\mathbf{c_0} \neq \mathbf{c_0^*}$. Let $\widetilde{\mathbf{c_2}} = \mathbf{r}\mathbf{u}+\mathbf{e_2}+(H_2(\mathbf{c_1},\mathbf{c_2'})\oplus\mathbf{k})\lfloor q/8\rfloor$.

$$\begin{aligned}2^\ell\mathbf{c_2}-[\mathbf{c_0}\|\mathbf{c_1}](\mathbf{x_0},\mathbf{x_1}) &= 2^\ell\mathbf{c_2}-\widetilde{\mathbf{c_2}}+\widetilde{\mathbf{c_2}}-[\mathbf{c_0}\|\mathbf{c_1}](\mathbf{x_0},\mathbf{x_1})\\ &= 1-2^\ell+\mathbf{r}\mathbf{u}+\mathbf{e_2}+\mathbf{k}\lfloor q/8\rfloor-[(\mathbf{r}\mathbf{h_r}+\mathbf{e_0})\|(\mathbf{r}(\mathbf{b_r}+H_1(\mathbf{c_0})\mathbf{d_r})+\mathbf{e_1})](\mathbf{x_0},\mathbf{x_1})\\ &\approx \mathbf{k}\lfloor q/8\rfloor+\mathbf{r}[\mathbf{u}-[\mathbf{h_r}\|(\mathbf{b_r}+H_1(\mathbf{c_0})\mathbf{d_r})](\mathbf{x_0},\mathbf{x_1})]+\mathbf{e_2}-\mathbf{e_0}\mathbf{x_0}-\mathbf{e_1}\mathbf{x_1}\end{aligned} \tag{6}$$

Let $\mathbf{e'} = \mathbf{u}-[\mathbf{h_r}\|(\mathbf{b_r}+H_1(\mathbf{c_0})\mathbf{d_r})](\mathbf{x_0},\mathbf{x_1})$. According to Lemma 10, $\|\mathbf{e'}\| \le \frac{1}{\sqrt{2\pi^2}}(\varrho^2+3\varrho)n^{5/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}$. Then,

$$\begin{aligned}&\|\mathbf{r}[\mathbf{u}-[\mathbf{h_r}\|(\mathbf{b_r}+H_1(\mathbf{c_0})\mathbf{d_r})](\mathbf{x_0},\mathbf{x_1})]+\mathbf{e_2}-\mathbf{e_0}\mathbf{x_0}-\mathbf{e_1}\mathbf{x_1}\|_\infty = \|\mathbf{r}\mathbf{e'}+\mathbf{e_2}-\mathbf{e_0}\mathbf{x_0}-\mathbf{e_1}\mathbf{x_1}\|_\infty\\ \le&\|\mathbf{r}\mathbf{e'}\|_\infty+\|\mathbf{e_2}\|_\infty+\|\mathbf{e_0}\mathbf{x_0}\|_\infty+\|\mathbf{e_1}\mathbf{x_1}\|_\infty \le \|\mathbf{r}\|_\infty\|\mathbf{e'}\|\sqrt{n}+\|\mathbf{e_2}\|_\infty+\|\mathbf{e_0}\|_\infty\|\mathbf{x_0}\|\sqrt{n}+\|\mathbf{e_1}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\\ \le&\|\mathbf{e'}\|\sqrt{n}+\|\mathbf{e_2}\|_\infty+\|\mathbf{e_0}\|_\infty\|\mathbf{x_0'}\|\sqrt{n}+\|\mathbf{e_0}\|_\infty\|\mathbf{p}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\sqrt{n}\sqrt{n}+\|\mathbf{e_1}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\\ \le&\frac{3}{\sqrt{2}\pi^2}\varrho n^{5/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}\sqrt{n}+\varrho+\varrho\sqrt{n}\sqrt{n}+\varrho^2\frac{1}{\sqrt{2}\pi^2}n(\ln(2^{(7+\lambda)/2}n))^{3/2}n^{1/2}n^{3/2}\\ &+\varrho\sqrt{n}\frac{1}{\sqrt{2}\pi^2}n\ln(2^{(7+\lambda)/2}n)^{3/2}n^{1/2}n^{1/2}\\ \approx&\frac{1}{\sqrt{2}\pi^2}(\varrho^2+3\varrho)n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}\end{aligned} \tag{7}$$

Since $\frac{1}{\sqrt{2}\pi^2}(\varrho^2+3\varrho)n^3(\ln(2^{(7+\lambda)/2}n))^{3/2} < \frac{1}{9}q$, the simulated private key can be used to correctly decrypt the public-key ciphertext. In summary, the games G_9 and G_8 are identical in $\mathcal{A}$'s view. Therefore, $\mathcal{A}$ has the same ability to attack the proposed schemes in G_9 and G_8. □

Lemma 12. *In the adversary $\mathcal{A}$'s view, the games G_{10} and G_9 are identical. It is reasonable that $\mathcal{C}$ needs $\mathcal{A}$ to return $(\mu, \mathbf{k}, \mathbf{x})$. Moreover, $Pr[F_{10}] = Pr[F_9]$.*

Proof. In the games G_{10} and G_9, the difference is that μ is chosen by $\mathcal{O}_{sign}$, and the signature $(\mathbf{k}, \mathbf{x})$ and the ciphertext $\mathbf{c_3^*}$ are also generated by $\mathcal{O}_{sign}$ in G_{10}, while they are all generated by $\mathcal{C}$ in G_9. However, the public keys and private keys for the signature are identical, and the procedure for generating $(\mu, \mathbf{k}, \mathbf{x})$ is executed strictly according to the signcryption algorithm. Therefore, the distribution of the challenge ciphertexts and the signatures involved in them are also identical in both games. In a word, the difference between the two games is only conceptual, and $\mathcal{A}$'s view in the games G_{10} and G_9 are completely the same. Therefore, what $\mathcal{A}$ can learn in the two games is identical.

Next, we demonstrate by contradiction that it is reasonable to require $\mathcal{A}$ to return $(\mu, \mathbf{k}, \mathbf{x})$ in G_{10}. Suppose that $\mathcal{A}$ does not know the information of $\mathbf{k}$, but it can recover $(\mu, \mathbf{x})$. That is, $\mathcal{A}$ computes the correct μ without knowing $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*})$. Due to $(\mu, \mathbf{x}) = DEC_{\mathbf{K}}(c_3)$, $\mathcal{A}$ obtaining $(\mu, \mathbf{x})$ from c_3 without $\mathbf{k}$ is contradictory to the security of the symmetric encryption. That is, $\mathcal{A}$ knows $\mathbf{k}$ with the same probability to recover $(\mu, \mathbf{x})$. Therefore, it is equivalent to computing $(\mu, \mathbf{x})$ and computing $(\mu, \mathbf{k}, \mathbf{x})$ from the challenge ciphertext. Thus, $\mathcal{C}$ requiring $\mathcal{A}$ to return $(\mu, \mathbf{k}, \mathbf{x})$ does not increase the hardness to attack the scheme. □

Lemma 13. *The modification to the game is rational, and the hardness of winning the game G_{11} is the same as that of winning G_{10}, in $\mathcal{A}$'s view. Furthermore, $Pr[F_{11}] = Pr[F_{10}]$.*

Proof. First, as proven in Lemma 12, if $\mathcal{A}$ can guess the $\mathbf{k}$ encrypted in $\mathbf{c_2}$, then it can compute the corresponding message and signature when given c_3. From this perspective, the crux of cracking the proposed scheme lies in obtaining $\mathbf{k}$.

Second, c_3 does not provide any assistance in obtaining $\mathbf{k}$. Firstly, it is impossible to directly obtain $\mathbf{k}$ from c_3. Since the symmetric encryption used in this scheme is IND-OT secure and one-to-one, c_3 does not leak any information of $H_3(\mathbf{k})$ to $\mathcal{A}$. Therefore, c_3 does not disclose any information of $\mathbf{k}$. Secondly, as proven in Lemma 8, the probability of obtaining assistance in computing $\mathbf{k}$ by changing c_3 is negligible.

Third, in the absence of c_3, it will not increase the probability of obtaining $\mathbf{k}$ by falsifying $(\mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*})$. Although the message and signature $(\mu, \mathbf{x})$ concealed in c_3 can help to verify the correctness of $\mathbf{k}$, the information of $\mathbf{k}$ can not also be changed without c_3. As proven in Lemma 7, the probability that $\mathcal{A}$ gets assistance to obtain $\mathbf{k}$ by changing $\mathbf{c_0^*}$ is negligible. The probability of $\mathcal{A}$ receiving assistance to obtain $\mathbf{k}$ by changing $(\mathbf{c_1^*}, \mathbf{c_2^*})$ is also negligible, as proven in Lemma 9.

In conclusion, the hardness of winning the games G_{11} and G_{10} is identical whether c_3 is known or not. □

Lemma 14. *In $\mathcal{A}$'s view, the games G_{12} and G_{11} are statistically indistinguishable. Furthermore, $Pr[F_{12}|\neg E_{12}] = Pr[F_{11}|\neg E_{11}]$ and $Pr[E_{12}] = Pr[E_{11}]$ is negligible.*

Proof. In the game G_{11}, $\mathbf{c_1^*} = \mathbf{r}(\mathbf{b_r} + H_1(\mathbf{c_0^*})\mathbf{d_r}) + \mathbf{e_1} = \mathbf{r}(\mathbf{h_r p} + \mathbf{e} - H_1(\mathbf{c_0^*})\mathbf{d_r} + H_1(\mathbf{c_0^*})\mathbf{d_r}) + \mathbf{e_1} = \mathbf{r h_r p} + \mathbf{re} + \mathbf{e_1} = \mathbf{c_0^* p} - \mathbf{e_0 p} + \mathbf{re} + \mathbf{e_1}$.

$$\begin{aligned}\mathbf{c_1^*} &= \mathbf{r}(\mathbf{b_r} + H_1(\mathbf{c_0^*})\mathbf{d_r}) + \mathbf{e_1} = \mathbf{r}(\mathbf{h_r p} + \mathbf{e} - H_1(\mathbf{c_0^*})\mathbf{d_r} + H_1(\mathbf{c_0^*})\mathbf{d_r}) + \mathbf{e_1} \\ &= \mathbf{r h_r p} + \mathbf{re} + \mathbf{e_1} = \mathbf{c_0^* p} - \mathbf{e_0 p} + \mathbf{re} + \mathbf{e_1}\end{aligned}$$

Let $\tilde{\mathbf{c}_1^*}$ denote the ciphertext $\mathbf{c}_1^*$ in the game G_{12}. Then, $\tilde{\mathbf{c}_1^*} = \mathbf{c}_0^*\mathbf{p} + \mathbf{e_1} = \mathbf{c}_1^* + \mathbf{e_0}\mathbf{p} - \mathbf{re}$, where $\mathbf{c}_1^*$ is the corresponding ciphertext in G_{11}. We need to argue that $(\mathbf{c}_0^*, \tilde{\mathbf{c}_1^*}, \mathbf{c}_2^*)$ is also a valid ciphertext. That is, $(\mathbf{c}_0^*, \tilde{\mathbf{c}_1^*}, \mathbf{c}_2^*)$ can be decrypted correctly with the valid private key. Suppose that $(\mathbf{s_0}, \mathbf{s_1})$ is a valid private key for the ciphertext, i.e., $(\mathbf{s_0}, \mathbf{s_1})$ is short enough. It might be good to set the key $(\mathbf{s_0}, \mathbf{s_1})$ to an equal element size to that of the simulated key. The simulated key can decrypt well, although in the two kinds of known keys, the element size of the simulated key is a little larger than that of the real key. That is, $\|\mathbf{s_0}\| \leq \frac{1}{\sqrt{2}\pi^2}\varrho n^{5/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}$, $\|\mathbf{s_1}\| \leq \frac{1}{\sqrt{2}\pi^2}\varrho n^{3/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}$.

$$\begin{aligned}\|\mathbf{c}_2^* - (\mathbf{c}_0^*\|\tilde{\mathbf{c}_1^*})(\mathbf{s_0}\|\mathbf{s_1})\|_\infty =& \|\mathbf{c}_2^* - (\mathbf{c}_0^*\|\mathbf{c}_1^*)(\mathbf{s_0}\|\mathbf{s_1}) - (\tilde{\mathbf{c}_1^*} - \mathbf{c}_1^*)\mathbf{s_1}\|_\infty \\ \leq& \|\mathbf{c}_2^* - (\mathbf{c}_0^*\|\mathbf{c}_1^*)(\mathbf{s_0}\|\mathbf{s_1})\|_\infty + \|(\tilde{\mathbf{c}_1^*} - \mathbf{c}_1^*)\mathbf{s_1}\|_\infty.\end{aligned} \tag{8}$$

$$\begin{aligned}\|\mathbf{c}_2^* - (\mathbf{c}_0^*\|\tilde{\mathbf{c}_1^*})(\mathbf{s_0}\|\mathbf{s_1})\|_\infty &= \|\mathbf{c}_2^* - (\mathbf{c}_0^*\|\mathbf{c}_1^*)(\mathbf{s_0}\|\mathbf{s_1}) - (\tilde{\mathbf{c}_1^*} - \mathbf{c}_1^*)\mathbf{s_1}\|_\infty \\ &\leq \|\mathbf{c}_2^* - (\mathbf{c}_0^*\|\mathbf{c}_1^*)(\mathbf{s_0}\|\mathbf{s_1})\|_\infty + \|(\tilde{\mathbf{c}_1^*} - \mathbf{c}_1^*)\mathbf{s_1}\|_\infty \\ \|\mathbf{c}_2^* - (\mathbf{c}_0^*\|\mathbf{c}_1^*)(\mathbf{s_0}\|\mathbf{s_1})\|_\infty &\leq \tfrac{1}{\sqrt{2}\pi^2}(\varrho^2 + 3\varrho)n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}.\end{aligned} \tag{9}$$

$$\begin{aligned}\|(\tilde{\mathbf{c}_1^*} - \mathbf{c}_1^*)\mathbf{s_1}\|_\infty = \|(\mathbf{e_0}\mathbf{p} - \mathbf{re})\mathbf{s_1}\|_\infty &\leq \|\mathbf{e_0}\mathbf{p}\mathbf{s_1}\|_\infty + \|\mathbf{r}\mathbf{e}\mathbf{s_1}\|_\infty \leq \|\mathbf{e_0}\|_\infty\|\mathbf{p}\mathbf{s_1}\|\sqrt{n} + \|\mathbf{r}\|_\infty\|\mathbf{e}\mathbf{s_1}\|_\infty \\ &\leq \varrho\|\mathbf{p}\|_\infty\|\mathbf{s_1}\|n^{3/2} + \|\mathbf{e}\|_\infty\|\mathbf{s_1}\|n^{3/2} \leq (\varrho^2 + \varrho)\frac{1}{\sqrt{2}\pi^2}n(\ln(2^{(7+\lambda)/2}n))^{3/2}\sqrt{n}n^{3/2} \\ &\leq \frac{1}{\sqrt{2}\pi^2}(\varrho^2 + \varrho)(\ln(2^{(7+\lambda)/2}n))^{3/2}n^3\end{aligned} \tag{10}$$

Therefore, $\|\mathbf{c}_2^* - (\mathbf{c}_0^*\|\tilde{\mathbf{c}_1^*})(\mathbf{s_0}\|\mathbf{s_1})\|_\infty \leq \frac{1}{\sqrt{2}\pi^2}(2\varrho^2 + 4\varrho)(\ln(2^{(7+\lambda)/2}n))^{3/2}n^3 < \lfloor q/16 \rfloor$. Thus, the challenge ciphertext is also a valid one. That is, if $\mathcal{A}$ can obtain the correct $\mathbf{k}$ from $(\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{c}_2^*)$, then it can also obtain the same $\mathbf{k}$ from $(\mathbf{c}_0^*, \tilde{\mathbf{c}_1^*}, \mathbf{c}_2^*)$. □

Lemma 15. *In $\mathcal{A}$'s view, the games G_{13} and G_{12} are computationally indistinguishable. Furthermore, $Pr[F_{13}|\neg E_{13}] = Pr[F_{12}|\neg E_{12}]$, and $Pr[E_{13}] = Pr[E_{12}]$ is negligible.*

Proof. Reduction from LWE. When the LWE oracle is $\mathcal{O} = \mathcal{O}_s$, namely the pseudorandom case, the challenge ciphertext has the same distribution as in the game G_{12}. First, the public keys used directly to encrypt the challenge ciphertext are $(\mathbf{h_r}, \mathbf{b_r} + H_1(\mathbf{c}_0^*)\mathbf{d_r}, \mathbf{u}) = (\mathbf{a_1}, \mathbf{a_1}\mathbf{p} + \mathbf{e} - H_1(\mathbf{c}_0^*)\mathbf{d_r} + H_1(\mathbf{c}_0^*)\mathbf{d_r}, \mathbf{a_2}) = (\mathbf{a_1}, \mathbf{a_1}\mathbf{p} + \mathbf{e}, \mathbf{a_2})$. Second, the challenge ciphertext is $\mathbf{c}_0^* = \mathbf{z_0} = \mathbf{r}\mathbf{a_1} + \mathbf{e_0}$ for some $\mathbf{r}, \mathbf{e_0} \leftarrow \{-1, 0, 1\}^n$, according to the definition of LWE. Due to $\mathbf{z_2} = \mathbf{r}\mathbf{a_2} + \mathbf{e_2}$, $\mathbf{c}_2^* = \mathbf{z_2} + \mathbf{k}\lfloor q/2 \rfloor = \mathbf{r}\mathbf{a_2} + \mathbf{e_2} + \mathbf{k}\lfloor q/2 \rfloor$. The $\mathbf{c}_1^*$ is as follows.

$$\mathbf{c}_1^* = \mathbf{c}_0^*\mathbf{p} + \tilde{\mathbf{e}} = (\mathbf{r}\mathbf{a_1} + \mathbf{e_0})\mathbf{p} + (\mathbf{r}\mathbf{e} - \mathbf{e_0}\mathbf{p} + \mathbf{e_1}) = \mathbf{r}(\mathbf{a_1}\mathbf{r} + \mathbf{e}) + \mathbf{e_1} = \mathbf{r}(\mathbf{b_r} + H_1(\mathbf{c}_0^*)\mathbf{d_r}) + \mathbf{e_1}.$$

Obviously, the challenge ciphertext $\mathbf{c}^*$ is exactly the challenge ciphertext in game G_{12}. In this case, $\mathcal{A}$'s advantage to distinguish the two games is zero.

When the LWE oracle is the real random case $\mathcal{O} = \mathcal{O}_\$$, the challenge ciphertext is uniformly distributed. In the challenge ciphertext, $\mathbf{c}_0^* = \mathbf{z_0}$ obeys the uniform distribution. Due to $\mathbf{z_1} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{c}_2^* = \mathbf{z_1} + \mathbf{k}\lfloor q/2 \rfloor$ is distributed uniformly. On the other hand, the generation method for $\mathbf{c}_1^*$ stays the same as in game G_{12}. $\mathbf{c}_1^* = (\mathbf{z_1}\mathbf{p} + \mathbf{e_1}) - \mathbf{e_0}\mathbf{p} + \mathbf{re}$. $(\mathbf{z_1}\mathbf{p} + \mathbf{e_1})$ is computationally indistinguishable from the uniform distribution over $\mathbb{Z}_q^n$, and $-\mathbf{e_0}\mathbf{p} + \mathbf{re}$ is restricted to a small range. Therefore, $\mathbf{c}_1^*$ is computationally indistinguishable from the uniform distribution over $\mathbb{Z}_q^n$. That is to say, in terms of distinguishing the games G_{12} and G_{11}, $\mathbf{c}_1^*$ is no more effective than $(\mathbf{c}_0^*, \mathbf{c}_2^*)$.

$\mathcal{C}$ uses the guess from $\mathcal{A}$ as the answer for the LWE oracle. Therefore, the advantage of $\mathcal{A}$ in distinguishing the two games is an adversary's advantage in attacking RLWE, which is negligible. □

In Theorem 2, the unforgeability of SC-NTRU will be proven by the interaction between the challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. To complete Theorem 2, some necessary conclusions are demonstrated in Lemmas 16 to 18.

Lemma 16. *In the query phase, the simulated signature generated by the simulated trapdoor in Algorithm 2 is indistinguishable from the real signature, and it can pass the signature verification.*

Proof. To argue that the signature generated by the simulated trapdoor in Algorithm 2 and that by the real trapdoor are identical, it is sufficient to demonstrate that the simulated trapdoor is short and has the same structure property as that of the real trapdoor. For the size property, we argue that the simulated trapdoor is short enough that it can use the same Gaussian parameter to sample as that used in the sample with the real trapdoor. For the structure property, we show that the size of the deviation and the length of the simulated trapdoor are the same, which is also the inherent character of the real trapdoor.

In Algorithm 2, for the case of $r = 0$:

$$\begin{aligned}
[\mathbf{h_s}\|\mathbf{f}](\mathbf{x_0},\mathbf{x_1}) &= [\mathbf{h_s}\|\mathbf{h_s}\mathbf{p}+\mathbf{e}+p\mathbf{h_f}](\mathbf{x_0'}-\mathbf{p}\mathbf{x_1},\mathbf{x_1}) = \mathbf{h_s}(\mathbf{x_0'}-\mathbf{p}\mathbf{x_1})+(\mathbf{h_s}\mathbf{p}+\mathbf{e}+p\mathbf{h_f})\mathbf{x_1}\\
&= p(\mathbf{h_s}\mathbf{x_0'}/p+\mathbf{h_f}\mathbf{x_1})+\mathbf{e}\mathbf{x_1} = \mathbf{e}\mathbf{x_1}-p\mathbf{x_1'}\\
\|\mathbf{e}\mathbf{x_1}-p\mathbf{x_1'}\| &\le p\|\mathbf{x_1'}\|+\|\mathbf{e}\mathbf{x_1}\| \le p\|\mathbf{x_1'}\|+\|\mathbf{e}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\sqrt{n} \le p\eta_3'\sqrt{n}+\varrho(\aleph+1)\eta_3'\sqrt{n}\sqrt{n}\sqrt{n}\\
&= \eta_3'\sqrt{n}(p+\varrho(\aleph+1)n) = \eta_3'\sqrt{n}(\aleph+1)(\xi+\varrho n) \approx \varrho(\aleph+1)\frac{1}{\sqrt{2}\pi^2}\sqrt{n}\ln(2^{(7+\lambda)/2}n)n^{3/2}\\
&= \frac{1}{\sqrt{2}\pi^2}\varrho(\aleph+1)n^2\ln(2^{(7+\lambda)/2}n)\\
\|(\mathbf{x_0},\mathbf{x_1})\| &= \|(\mathbf{x_0'}-\mathbf{p}\mathbf{x_1},\mathbf{x_1})\| = \sqrt{\|\mathbf{x_0'}-\mathbf{p}\mathbf{x_1}\|^2+\|\mathbf{x_1}\|^2} \le \sqrt{\|\mathbf{x_0'}\|^2+\|\mathbf{p}\mathbf{x_1}\|^2+\|\mathbf{x_1}\|^2}\\
&\approx \|\mathbf{p}\mathbf{x_1}\| \le \|\mathbf{p}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\sqrt{n} \le (\aleph+1)\eta_3 n^{3/2} = \frac{1}{\sqrt{2}\pi^2}(\aleph+1)n^2\ln(2^{(7+\lambda)/2}n)
\end{aligned} \tag{11}$$

In the case of $r = 1$:

$$\begin{aligned}
[\mathbf{h_s}\|\mathbf{f}](\mathbf{x_0},\mathbf{x_1}) &= [\mathbf{h_s}\|\mathbf{h_s}\mathbf{p}+\mathbf{e}+p\mathbf{h_f}](-\mathbf{p}\mathbf{x_1},\mathbf{x_1}) = \mathbf{h_s}(-\mathbf{p}\mathbf{x_1})+(\mathbf{h_s}\mathbf{p}+\mathbf{e}+p\mathbf{h_f})\mathbf{x_1}\\
&= p(\mathbf{x_1'}+\mathbf{h_f}\mathbf{x_1})+\mathbf{e}\mathbf{x_1}-p\mathbf{x_1'} = \mathbf{e}\mathbf{x_1}-p\mathbf{x_1'}\\
\|\mathbf{e}\mathbf{x_1}-p\mathbf{x_1'}\| &\le p\|\mathbf{x_1'}\|+\|\mathbf{e}\mathbf{x_1}\| \le p\|\mathbf{x_1'}\|+\|\mathbf{e}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\sqrt{n} \le p\eta_0\sqrt{n}+\varrho(\aleph+1)\eta_0\sqrt{n}n\\
&= \eta_0\sqrt{n}((\aleph+1)\xi+\varrho(\aleph+1)n) \approx \frac{1}{\sqrt{2}\pi}\varrho(\aleph+1)n^{3/2}\sqrt{\ln(2^{(7+\lambda)/2}n)}\\
\|(\mathbf{x_0},\mathbf{x_1})\| &= \|(-\mathbf{p}\mathbf{x_1},\mathbf{x_1})\| \le \sqrt{\|\mathbf{p}\mathbf{x_1}\|^2+\|\mathbf{x_1}\|^2} \approx \|\mathbf{p}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\sqrt{n} = (\aleph+1)\eta_0\sqrt{n}n\\
&\approx \frac{1}{\sqrt{2}\pi}(\aleph+1)n^{3/2}\sqrt{\ln(2^{(7+\lambda)/2}n)}
\end{aligned} \tag{12}$$

When using the simulated trapdoor to sign, the Gaussian parameter to sample is

$$\frac{1}{\sqrt{2}\pi^2}(\aleph+1)n^2\ln(2^{(7+\lambda)/2}n)\frac{1}{\sqrt{2}\pi}\sqrt{\ln(2^{(7+\lambda)/2}n)} = \frac{1}{2\pi^3}(\aleph+1)n^2(\ln(2^{(7+\lambda)/2}n))^{3/2}$$

For simplicity, it can be set as $\frac{1}{2\pi^3}(\aleph+1)n^2(\ln(2^{(7+\lambda)/2}n))^{3/2}$. The Gaussian parameter for the real trapdoor to sign is also this value, according to Equation (1). It is easy to see that the size and structure property of the simulated trapdoor are the same as those of the real trapdoor, respectively. □

Lemma 17. *Under the parameter settings in Equation (1), the vector* $\mathbf{x_0}+\mathbf{p}\mathbf{x_1}$ *constructed by* $\mathcal{C}$ *in the forgery phase of Theorem 2 derives a valid solution for the search RLWE instance* $(\mathbf{h_s},\mathbf{y})$.

Proof. Since the $\mathbf{x}$ output by $\mathcal{A}$ is a valid forgery, $\mathbf{x}$ obeys $D_{\mathbb{Z}^n,\mathbf{0},\eta}$ with $\eta = \frac{1}{2\pi^3}(\aleph+1)n^2(\ln(2^{(7+\lambda)/2}n))^{3/2}$ and $\|\mathbf{y}-(\mathbf{h_s},\mathbf{f})\mathbf{x}\|_\infty \le \frac{1}{2\pi^3}(\aleph+1)\varsigma n^2(\ln(2^{(7+\lambda)/2}n))^{3/2}$ for some constant $\varsigma > 1$. Then, it has

$$\begin{aligned}
&\|\mathbf{y}-\mathbf{h_s}(\mathbf{x_0}+\mathbf{px_1})\|_\infty = \|\mathbf{y}-[\mathbf{h_s}||\mathbf{h_sp}](\mathbf{x_0},\mathbf{x_1})\|_\infty = \|\mathbf{y}-[\mathbf{h_s}||\mathbf{h_sp}+\mathbf{e}](\mathbf{x_0},\mathbf{x_1})+\mathbf{ex_1}\|_\infty \\
\le&\|\mathbf{y}-[\mathbf{h_s}||\mathbf{h_sp}+\mathbf{e}](\mathbf{x_0},\mathbf{x_1})\|_\infty + \|\mathbf{ex_1}\|_\infty \le \frac{1}{2\pi^3}(\aleph+1)\varsigma n^2(\ln(2^{(7+\lambda)/2}n))^{3/2} + \|\mathbf{e}\|_\infty\|\mathbf{x_1}\|\sqrt{n} \\
\le&\frac{1}{2\pi^3}(\aleph+1)\varsigma n^2(\ln(2^{(7+\lambda)/2}n))^{3/2} + \varrho\frac{1}{2\pi^3}(\aleph+1)^2n^2(\ln(2^{(7+\lambda)/2}n))^{3/2}n \\
\approx&\frac{1}{2\pi^3}\varrho(\aleph+1)^2n^3(\ln(2^{(7+\lambda)/2}n))^{3/2}
\end{aligned} \tag{13}$$

Similarly, $\|\mathbf{y}-\mathbf{h_s}(\mathbf{x_0}+\mathbf{px_1})\| \le \frac{1}{2\pi^3}\varrho(\aleph+1)^2n^{7/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}$.

Furthermore, the size of the vector $\mathbf{x_0}+\mathbf{px_1}$ may be evaluated in the same approach.

$$\begin{aligned}
\|\mathbf{x_0}+\mathbf{px_1}\| \le&\|\mathbf{x_0}\| + \|\mathbf{px_1}\| \le \|\mathbf{x_0}\| + \|\mathbf{p}\|_\infty\|\mathbf{x_1}\|\sqrt{n}\sqrt{n} \\
\le&\frac{1}{2\pi^3}(\aleph+1)^2n^{7/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}.
\end{aligned} \tag{14}$$

Therefore, $\mathbf{y}-\mathbf{h_s}(\mathbf{x_0}+\mathbf{px_1})$ and $\mathbf{x_0}+\mathbf{px_1}$ is short enough, and they form a solution for the RLWE instance $(\mathbf{h_s},\mathbf{y})$ under the parameter settings in Equation (1). □

Lemma 18. *For the query times* $0 < Q < \sqrt{\frac{2}{3}\pi\aleph\xi(\xi+1)}$*, the upper bound* ξ *of* p_i *for* i *from* 1 *to* $\aleph$*. Both the signature queries and forgery in the simulation can be completed without aborts, with probability*

$$\frac{\sqrt{3}}{\pi\sqrt{2\aleph\xi(\xi+1)}}\left(1-\frac{\sqrt{3}Q}{\pi\sqrt{2\aleph\xi(\xi+1)}}\right) \le Pr[complete] \le \frac{\sqrt{3}}{\pi\sqrt{2\aleph\xi(\xi+1)}}, \tag{15}$$

no matter what strategy $\mathcal{A}$ *takes. In particular, when* $Q < \frac{\sqrt{3}\pi^4q}{20n^2\aleph^{1/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}}$*, the sucessful probability satisfies* $\frac{\sqrt{3}}{\pi\sqrt{20\aleph\xi(\xi+1)}} < Pr[complete] < \frac{\sqrt{3}}{\pi\sqrt{2\aleph\xi(\xi+1)}}$.

Proof. First, we evaluate the probability $Pr[1+\sum_{i=1}^{\aleph}(-1)^{\nu_i}p_i = 0|\nu_i \xleftarrow{\$} \{0,1\}, p_i \xleftarrow{\$} \{-\xi,-\xi+1,\cdots,\xi\}]$ since $(-1)^{\nu_i}p_i$ submits the uniform distribution over $\{-\xi,-\xi+1,\cdots,\xi\}$, $Pr[(-1)^{\nu_i}p_i = t|\nu_i \xleftarrow{\$} \{0,1\}, p_i \xleftarrow{\$} \{-\xi,-\xi+1,\cdots,\xi\}, t \in \{-\xi,-\xi+1,\cdots,\xi\}]$ $= 1/(2\xi+1)$. The mean and variance are $E[(-1)^{\nu_i}p_i] = 0$, $D[(-1)^{\nu_i}p_i] = \frac{1}{2\xi+1}\sum_{i=-\xi}^{\xi}i^2 = \frac{1}{3}\xi(\xi+1)$, respectively. Then, $E[\frac{1}{\aleph}\sum_{i=1}^{\aleph}(-1)^{\nu_i}p_i] = 0$, $D[\frac{1}{\aleph}\sum_{i=1}^{\aleph}(-1)^{\nu_i}p_i] = \frac{1}{3\aleph}\xi(\xi+1)$. When $\aleph$ becomes big enough, $\frac{1}{\aleph}\sum_{i=1}^{\aleph}(-1)^{\nu_i}p_i$ obeys the Gaussian distribution $\psi_{0,\sqrt{\frac{1}{3\aleph}\xi(\xi+1)}}$, according to the Central Limit Theorem. Then,

$$\begin{aligned}
&Pr[1+\sum_{i=1}^{\aleph}(-1)^{\nu_i}p_i = 0|\nu_i \xleftarrow{\$} \{0,1\}, p_i \xleftarrow{\$} \{-\xi,-\xi+1,\cdots,\xi\}] \\
=&Pr[\frac{1}{\aleph}\sum_{i=1}^{\aleph}(-1)^{\nu_i}p_i = -\frac{1}{\aleph}|\nu_i \xleftarrow{\$} \{0,1\}, p_i \xleftarrow{\$} \{-\xi,-\xi+1,\cdots,\xi\}] \\
=&\frac{1}{\sqrt{2\pi}\sqrt{\frac{\aleph}{3}\xi(\xi+1)}}e^{-1/(\frac{2}{3}\aleph^3\xi(\xi+1))} \approx \frac{\sqrt{3}}{\sqrt{2\pi\aleph\xi(\xi+1)}}
\end{aligned} \tag{16}$$

This probability is non-negligible.

Next, we evaluate the probability that the simulation normally completes the signature queries and forgery with the affine subspaces method, as in [34]. Since $\mathbf{f_i} = \mathbf{h_sp_i}+\mathbf{e_i}+p_i'\mathbf{h_f}$, $\mathbf{p_i},\mathbf{e_i} \xleftarrow{\$} \{-1,0,1\}^n$, $p_i' \xleftarrow{\$} \{-\xi,-\xi+1,\cdots,\xi\}$ from $i=0$ to $\aleph$ except $p_0' = 1$. At the beginning of the game, the values of p_i's are completely hidden from $\mathcal{A}$ by the LWE instances $\mathbf{h_sp_i}+\mathbf{e_i}$. Even though $\mathcal{A}$ knows the calculation form of $\mathbf{f_i}$, some information is

leaked due to signature queries. $\mathcal{A}$ can not infer enough information by Bayesian updating to observably increase the probability of forging. For ν^*, the event $1+\sum_{i=1}^{\aleph}(-1)^{\nu_i^*}p_i'=0$ corresponds a hyperplane V^* with $|V^*|=\frac{\sqrt{3}}{\sqrt{2\pi\aleph\xi(\xi+1)}}|V|$, where $|V|$ denotes the size of the total space determined by $(p_1,p_2,\cdots p_\aleph)$. Let V_j denote a hyperplane that can lead to an abort in the jth signature query. Then, $|V^*\setminus V_j|=|V^*-V^*\cap V_j|=\frac{\sqrt{3}}{\sqrt{2\pi\aleph\xi(\xi+1)}}(1-\frac{\sqrt{3}}{\sqrt{2\pi\aleph\xi(\xi+1)}})|V|$. The lower bound can be computed by the union bound,

$$Pr[complete]=Pr[(p_1,p_2\cdots,p_\aleph)\in(\mathbb{V}^*\setminus\cup_{j=1}^{Q}V_j)]\geq\frac{\sqrt{3}}{\sqrt{2\pi\aleph\xi(\xi+1)}}\left(1-\frac{\sqrt{3}Q}{\sqrt{2\pi\aleph\xi(\xi+1)}}\right) \quad (17)$$

The upper bound can be evaluated trivially, $Pr[complete]\leq\frac{\sqrt{3}}{\sqrt{2\pi\aleph\xi(\xi+1)}}$.

Specifically, when

$$Q<\frac{9\sqrt{2\pi\aleph\xi(\xi+1)}}{10\sqrt{3}}<\frac{9\sqrt{2\pi\aleph\varrho n(\varrho n+1)}}{10\sqrt{3}}\approx\frac{9\varrho n\sqrt{2\pi\aleph}}{10\sqrt{3}}\approx\frac{\sqrt{3}\pi^{7/2}q}{20n^2\aleph^{1/2}(\ln(2^{(7+\lambda)/2}n))^{3/2}},$$

$\frac{\sqrt{3}Q}{\sqrt{2\pi\aleph\xi(\xi+1)}}<\frac{9}{10}$, $\frac{\sqrt{3}}{10\sqrt{2\pi\aleph\xi(\xi+1)}}<Pr[complete]<\frac{\sqrt{3}}{\pi\sqrt{2\aleph\xi(\xi+1)}}$. □

Theorem 2. *If there exists an adversary $\mathcal{A}$ who can forge an existential signature for SC-NTRU with probability ϵ by carrying out adaptive chosen-message queries, then a probabilistic algorithm $\mathcal{C}$ can be constructed to solve the search $LWE_{n,m,q}$ problem with parameter settings in Equation (1) in almost the same time with probability* $\frac{\sqrt{3}\epsilon}{\pi\sqrt{2\aleph\xi(\xi+1)}}\left(1-\frac{\sqrt{3}Q}{\pi\sqrt{2\aleph\xi(\xi+1)}}\right)$.

Proof. Setup. The challenger $\mathcal{C}$ queries the RLWE oracle to fetch a random RLWE(q,n,m,β) instance $(\mathbf{h_s},\mathbf{y})$. Then, $\mathcal{C}$ generates the simulation environment for $\mathcal{A}$ as follows.

1. $\mathcal{C}$ generates a public and private key pair by running the KeyGen algorithm $(\mathbf{B_f},\mathbf{h_f})\leftarrow$ KeyGen(n,q).
2. $\mathcal{C}$ samples $\mathbf{p_i},\mathbf{e_i}\xleftarrow{\$}\{-1,0,1\}^n$, $p_i'\xleftarrow{\$}\{-\xi,-\xi+1,\cdots,\xi\}$ from $i=1$ to $\aleph$ but sets $p_i'=1$, then computes $\mathbf{f_i}=\mathbf{h_s}\mathbf{p_i}+\mathbf{e_i}+p_i'\mathbf{h_f}$ for i from 1 to $\aleph$.

$\mathcal{C}$ returns $(\mathbf{h_f},\{\mathbf{f_i}\}_{i=0}^{\aleph})$ to $\mathcal{A}$.

Queries. $\mathcal{A}$ submits a randomly chosen message $\mu\in\{0,1\}^*$ to $\mathcal{C}$ to query. $\mathcal{C}$ replies to the query by generating the corresponding signature as follows.

1. $\mathbf{k}\xleftarrow{\$}\{0,1\}^\lambda$.
2. ν: $=H_0(\mu,\mathbf{b_r},\mathbf{k})$.
3. $p'=p_0'+\sum_{i=1}^{\aleph}(-1)^{\nu_i}p_i'$. If $p'=0$, go to step 1.
4. $\mathbf{p}=\mathbf{p_0}+\sum_{i=1}^{\aleph}(-1)^{\nu_i}\mathbf{p_i}$.
5. $\mathbf{f}$: $=\mathbf{f_0}+\sum_{i=1}^{\aleph}(-1)^{\nu_i}\mathbf{f_i}$.
6. $\mathbf{e}$: $=\mathbf{e_0}+\sum_{i=1}^{\aleph}(-1)^{\nu_i}\mathbf{e_i}$.
7. Generate a basis $\mathbf{B}:=$ SimExtractS $(\mathbf{h_s},\mathbf{p},\mathbf{e},p',\mathbf{h_f},\mathbf{B_f})$ for $(\mathbf{h_s},\mathbf{f})$ (see Algorithm 2).
8. $(\mathbf{x}',\mathbf{x})$: $=(\mathbf{y},\mathbf{0})-$ SamplePre$(\mathbf{B},\eta,(\mathbf{y},\mathbf{0}))$.
9. return $\mathbf{x}$ as the signature.

Forgery. $\mathcal{C}$ receives a forgery $\mathbf{x}$ signature on a new message μ, then constructs the solution for ISIS as follows.

1. Compute $p'=p_0'+\sum_{i=1}^{\aleph}(-1)^{\nu_i}p_i'$. If $p'\neq 0$, abort.
2. $\mathbf{p}=\mathbf{p_0}+\sum_{i=1}^{\aleph}(-1)^{\nu_i}\mathbf{p_i}$.
3. Divide $\mathbf{x}$ as $(\mathbf{x_0},\mathbf{x_1})\in\mathbb{Z}_q^n\times\mathbb{Z}_q^n$.
4. Return $(\mathbf{x_0}+\mathbf{p}\mathbf{x_1},\mathbf{y}-\mathbf{h_s}(\mathbf{x_0}+\mathbf{p}\mathbf{x_1}))$ as the solution for $(\mathbf{h_s},\mathbf{y})$.

Firstly, according to Lemma 16, the simulated signatures generated during the query phase are valid and indistinguishable from the real signatures. That is, $\mathcal{C}$ creates a simulated environment, indistinguishable from reality for adversaries. This will lead $\mathcal{A}$ to complete the reduction game.

Secondly, according to Lemma 17, the coefficient vector of $\mathbf{y} - \mathbf{h_s}(\mathbf{x_0} + \mathbf{p}\mathbf{x_1})$ is short enough under the parameter settings in Equation (1). Therefore, the polynomials generated in step 4 of the forgery phase form a solution for the RLWE instance $(\mathbf{h_s}, \mathbf{y})$.

Finally, according to Lemma 18, the simulation can successfully complete both signature queries and forgery without aborting, with non-negligible probability.

In summary, $\mathcal{C}$ can obtain a solution for an RLWE instance with non-negligible probability, if $\mathcal{A}$ has the ability to forge a valid signature for the SC-NTRU scheme. □

Algorithm 2 SimExtractS($\mathbf{h_s}, \mathbf{p}, \mathbf{e}, p, \mathbf{h_f}, \mathbf{B_f}$)

Require:
public keys $\mathbf{h_s}, \mathbf{h_f} \in \mathcal{R}_q$ and syndrome $\mathbf{u} \in \mathcal{R}_q$, in which $\mathbf{h_f} \leftarrow KeyGen$, $\mathbf{b_r} = \mathbf{h_r}\mathbf{p} + \mathbf{e}$ for $\mathbf{p}, \mathbf{e} \leftarrow \{-1, 0, 1\}^n$;
a pair of un-equivalent tags τ, τ^* ;
a syndrome $\mathbf{u} \in \mathcal{R}_q$;
a basis $\mathbf{B_f}$ for $\mathbf{h_f}$.
Ensure:
A private key $(\mathbf{x_1}, \mathbf{x_2})$.

$\mathbf{f} = \mathbf{h_s}\mathbf{p} + \mathbf{e} + p\mathbf{h_f}$
$\Gamma = \{\}$
repeat
  $r \xleftarrow{\$} \{0, 1\}$
  if $r = 0$ **then**
    $\mathbf{x_0'} \leftarrow D_{\mathbb{Z}^n, \eta_0, \mathbf{0}}$ (where $\eta_0 = \frac{1}{\pi}\sqrt{\frac{1}{2}\ln\left(2^{(7+\lambda)/2}n\right)}$)
    $\mathbf{y'} = (-\mathbf{h_s}\mathbf{x_0'})/p$
    $(\mathbf{x_1'}, \mathbf{x_1}) := (\mathbf{y'}, \mathbf{0}) - \text{SamplePre}(\mathbf{B_f}, \eta_3, (\mathbf{y'}, \mathbf{0}))$ where $\eta_3 = \frac{1.17}{\pi}\sqrt{\frac{1}{2}\ln\left(2^{(7+\lambda)/2}n\right)q}$
    $\mathbf{x_0} = \mathbf{x_0'} - \mathbf{p}\mathbf{x_1}$
  else
    $(\mathbf{x_1'}, \mathbf{x_1}) := (\mathbf{g_{f_i}}, \mathbf{t_{f_i}})$ where $(\mathbf{g_{f_i}}, \mathbf{t_{f_i}})$ is a basis vector of $\Lambda_{\mathbf{f},q}$
    $\mathbf{x_0} = -\mathbf{p}\mathbf{x_1}$
  end if
  $\mathbf{b} = (\mathbf{x_0}, \mathbf{x_1})$
  if $\mathbf{b}$ is linearly independent with the vectors in Γ **then**
    $\Gamma = \Gamma \cap \mathbf{b}$
  end if
until $|\Gamma| = 2n$
covert $2n$ linearly independent vectors in Γ to a basis $\mathbf{B}$ of $(\mathbf{h_s} \| \mathbf{f})$ with the algorithm in Lemma 7.1 of [44].

5.2. Performance

In the existing lattice-based signcryption schemes, some building blocks are often used, such as the algorithm SampleD of [18], Inver of [18], SamplePre of [41], inverting matrix and solving system of linear equations, etc. To facilitate the performance comparison, we summarize some basic conclusions about the computational cost for these building blocks. In order to clearly state these conclusions, we introduce some notations for the basic mathematical operations. Let $\times_{\mathbb{Z}}$, $\times_{\mathbb{Z}_q}$, $\times_{\mathbb{R}}$ denote the multiplication operation over $\mathbb{Z}$, $\mathbb{Z}_q$, $\mathbb{R}$, respectively. Let $+_{\mathbb{Z}}$, $+_{\mathbb{Z}_q}$, $+_{\mathbb{R}}$ denote the addition operation over $\mathbb{Z}$, $\mathbb{Z}_q$, $\mathbb{R}$, respectively.

The computational overhead of matrix inversion and solving systems of linear equations can be evaluated by regular computation.

Proposition 5. *The computational cost to invert an n-dimensional matrix over* $\mathbb{Z}_q$ *is about* $\frac{2}{3}n(n^2 + 3n - 1)$ *multiplications over* $\mathbb{Z}_q$.

Proposition 6. *The computational cost of solving* $n \times m$*-dimension nonhomogeneous linear equations over* $\mathbb{Z}_q$ *is about* $\frac{1}{2}n(n+1)(m+1) + \frac{1}{6}(n-2)(n-1)(n+3)$ *multiplications and additions over* $\mathbb{Z}_q$*, and* $2nO(\log q)$ *inversion over* $\mathbb{Z}_q$*. Here, the equation substitution is* $mn - \frac{1}{2}n(n-1)$ *multiplications and* $mn + \frac{1}{2}n(n+1)$ *additions over* $\mathbb{Z}_q$ *and* $O(\log q)$ *inversion over* $\mathbb{Z}_q$.

Micciancio and Peikert presented a pre-image sample algorithm named SampleD$^{\mathcal{O}}$, specifically designed for the MP trapdoor [18]. For more details, please refer to [18]. The computational overhead of the algorithm can be broken down into the following steps:

- Step 1: Generating $(2n \log q)$-dimension DGS.
- Step 2: Performing $(n^2 \log^2 q + n^2 \log q) \times_{\mathbb{Z}}$ and $+_{\mathbb{Z}}$, as well as $(n^2 \log q) \times_{\mathbb{Z}_q}$ and $+_{\mathbb{Z}_q}$.
- Step 3: Conducting $(n^2) \times_{\mathbb{Z}}$ and $+_{\mathbb{Z}}$, $(n^2) \times_{\mathbb{Z}_q}$ and $+_{\mathbb{Z}_q}$, and utilizing $n \log q$-dimension DGS.
- Step 4: Involving $(n^2 \log^2 q) \times_{\mathbb{Z}}$ and $+_{\mathbb{Z}}$.

Therefore, the overall computational overhead of SampleD$^{\mathcal{O}}$ is summarized as follows.

Proposition 7. *The computational cost of Algorithm SampleD*$^{\mathcal{O}}$ *of [18] is about* $3n \log q$ *discrete Gaussian samples (DGS),* $2n^2 \log^2 q + n^2 \log q + n \log q$ *multiplications and additions over* $\mathbb{Z}$*,* $n^2 \log q + n^2$ *multiplications and additions over* $\mathbb{Z}_q$*. The overhead to compute the Gaussian parameter is* $3n^3 \log^3 q$*, which can be precomputed.*

Except for SampleD, the computational cost of signcryption of [17] is about $5n \log q$ DGS, $8n^2 \log q$ multiplications over $\mathbb{Z}_q$, $(\lambda + 8)n^2 \log q + n \log q - n$ additions over $\mathbb{Z}_q$, $n^2 \log^2 q + n \log q$ multiplications over $\mathbb{Z}$, $n^2 \log^2 q - n \log q$ additions over $\mathbb{Z}$.

In [18], an inversion algorithm called *Inver* is presented for the MP trapdoor. The computational overhead of the steps is described as follows.

- Step 1: Involves $(n^2 \log^2 q) \times_{\mathbb{Z}}$ and $+_{\mathbb{Z}}$.
- Step 2: Includes $(n \log q) \times_{\mathbb{Z}}$ and $+_{\mathbb{Z}}$, as well as $(n \log q) \times_{\mathbb{Z}_q}$ and $+_{\mathbb{Z}_q}$.
- Step 3: Requires $(2n^2 \log q + n^2) \times_{\mathbb{Z}_q}$ and $+_{\mathbb{Z}}$.

Therefore, the total computational overhead of Algorithm *Inver* is as follows.

Proposition 8. *The computational cost of Algorithm* ***Inver*** *of [18] is about* $n^2 \log^2 q + 2n^2 \log q + n^2 + n \log q$ *multiplications and* $n^2 \log^2 q + 2n^2 \log q + n^2 - 2n$ *additions over* $\mathbb{Z}_q$*,* $2n \log q$ *multiplications and additions over* $\mathbb{Z}$*. In the process, the cost of the inversion oracle is* $2n \log q$ *multiplications and additions over* $\mathbb{Z}$.

In addition to solving system of linear equations, Algorithm SamplePre also involves executing the randomized nearest-plane algorithm. The computational cost of its steps is as follows.

- step a: Involves $(2m) \times_{\mathbb{R}}$ and $(2m - 2) +_{\mathbb{R}}$.
- step b: Requires 1 DGS.
- step c: Needs $m \times_{\mathbb{Z}}$ and $(2m) +_{\mathbb{Z}}$.

The procedure is carried out m times. Consequently, the total computational cost of SamplePre without solving equations is calculated as follows.

Proposition 9. *Except for solving equations, the computational cost of Algorithm SamplePre of [41] is about* $2\ m^2$ *multiplications and additions over* $\mathbb{R}$*,* m^2 *multiplications and* $2\ m^2$ *additions over* $\mathbb{Z}$ *and* m *DGS.*

The operations involved in Algorithm Gaussian_Sampler of [38] are almost identical to those in Algorithm SamplePre except for the solving equations part. However, the dimension of the basis used in Gaussian_Sampler is $2n \times 2n$. Consequently, its computational overhead includes $(8n^2)$ $\times_{\mathbb{R}}$ and $+_{\mathbb{R}}$, $(4n^2)$ $\times_{\mathbb{Z}}$ and $(8n^2)$ $+_{\mathbb{Z}}$, and $2n$ DGS. However, the circulant property of the basis matrix allows for the use of fast Fourier orthogonalization, which can speed up the procedure significantly [45]. According to [45], when adopting fast Fourier orthogonalization, the complexity of the Gaussian_Sampler is given as follows.

Proposition 10. *The computational cost of Algorithm Gaussian_Sampler of [38] is about* $2n$ *DGS,* $4\Theta(n \log n)$ *multiplications and additions over* $\mathbb{R}$, $4\Theta(n \log n)$ *multiplications and additions over* $\mathbb{Z}$.

In Table 3, the sizes of public parameters, public key, private key, ciphertext, and security are compared. The sizes of the public parameters and the public key of SC-NTRU are approximately in the order of magnitude of $1/(n \log q)$ of those of YWW+13 [17], SS18 [19], and YCL+19 [20]. The private key size of SC-NTRU is roughly in the order of magnitude of $2/(n \log^2 q)$ of those of YWW+13 [17] and SS18 [19], and $4/(n \log^2 q)$ of that of YCL+19 [20]. Except for the plaintext length, the ciphertext size of SC-NTRU is about at the order of magnitude of $1/\log q$ of those of YWW+13 [17], SS18 [19], and YCL+19 [20]. Regarding security, the proposed scheme achieves IND-CCA2 security against adaptively chosen ciphertext attacks, similar to the other schemes in the table. However, when facing adaptively chosen message attacks, the proposed scheme is EUF-CMA secure instead of SUF-CMA security, as in YWW+13 [17] and SS18 [19]. Our intention is to demonstrate that a EUF-CMA secure signature component is sufficient to guarantee the IND-CCA2 security of the signcryption ciphertext.

Table 3. Comparison of the key size, public parameter size, and security of the related schemes.

Scheme	**YWW+13** [17]	**SS18** [19]	**YCL+19** [20]	**Ours**
public parameter	$(\lambda+3)n^2\log^2 q$ $+n\ell\log q$	$(\lambda+3)n^2\log^2 q$ $+n\ell\log q$	$(2k'+\ell+5)$ $n^2\log^2 q$	$(\aleph+1)n\log q$ [1]
public key	$3n^2\log^2 q$ $+n\ell\log q$	$3n^2\log^2 q$ $+n\ell\log q$	$n^2\log^2 q$	$3n\log q$
private key	$24n^2\log^2 q\log\sigma$ [2]	$24n^2\log^2 q\log\sigma$	$12n^2\log^2 q\log\sigma$	$48n\log\sigma$
ciphertext	$2n\log^2 q+\lvert\mu\rvert$ [3] $+48n\log q\log\sigma_1$	$2n\log^2 q+\lvert\mu\rvert$ $+12(7n+\ell)\log q$ $\cdot\log\sigma_2$	$4n\log^2 q+\lvert\mu\rvert$ $+36n\log q\log\sigma_3$ $+n+\ell$	$3n\log q+\lvert\mu\rvert$ $+24n\log\sigma_4$
security	IND-CCA2, SUF-CMA	IND-CCA2, SUF-CMA	IND-CCA2, EUF-CMA	IND-CCA2, EUF-CMA
based on NIST	No	No	No	Yes

[1] $\lambda, \ell, \aleph$ are all with the security parameter, so they are roughly the same. [2] Every element of the private key is stored in $12\log\sigma$, when its Gaussian parameter is σ. [3] $|\mu|$ denotes the length of the message.

In Table 4, the computational overhead of YWW+13 [17], SS18 [19], YCL+19 [20], and SC-NTRU is compared. First, the cost of the key generation is compared. In the key generation phase of YWW+13 [17] and SS18 [19], the computation of $(\frac{2}{3}n^3+2n^2)$ $\times_{\mathbb{Z}_q}$ is required to invert the matrix $\mathbf{H}$. As it is the repeating computation in signcryption, it is reasonable to count its overhead into the key generation instead of signcryption. The numbers of $\times_{\mathbb{Z}_q}$ and $+_{\mathbb{Z}_q}$ of SC-NTRU are in the order of magnitude of $1/n$ of those of YWW+13 [17] and SS18 [19], and at a lower order of magnitude of YCL+19 [20]. The number of DGS of SC-NTRU is about $2/(n\log^2 q)$ of those of YWW+13 [17] and SS18 [19]. In SC-NTRU, the samples from $U(\mathbb{Z}_q)$ are not needed; however, the numbers of samples reach $n^2\log q$ in YWW+13 [17] and SS18 [19].

Table 4. Comparison of the computation overhead of the related signcryption schemes.

	Computation Type	**YWW+13** [17]	**SS18** [19]	**YCL+19** [20]	**Ours**
Setup	$\leftarrow U(\mathbb{Z}_q)$	$(\lambda+3)n^2\log q$ $+n\ell$	$(\lambda+3)n^2\log q$ $+n\ell+n$	$(\ell+5)n\log q$	$(\aleph+5)n$
KeyGen	$\leftarrow U(\mathbb{Z}_q)$	$n^2\log q$	$n^2\log q$	$\sigma n((m-2\sigma-r)$ $/\lceil\log q\rceil+r+2)$	0
	DGS	$n^2\log^2 q$	$n^2\log^2 q$	0	$2n$
	$\times_{\mathbb{Z}_q}$	$n^3\log^2 q$ $+\frac{2}{3}n^3+2n^2$	$n^3\log^2 q$ $+\frac{2}{3}n^3+2n^2$	$(m-2\sigma-r)$ $(r+\sigma)$	$7O(n^2\log n)$ $+8O(n\log n)$
	$+_{\mathbb{Z}_q}$	$n^3\log^2 q$ $+\frac{2}{3}n^3+2n^2$	$n^3\log^2 q$ $+\frac{2}{3}n^3+2n^2$	$(m-2\sigma-r)$ $(r+\sigma)$	$7O(n^2\log n)$ $+8O(n\log n)$
	$\times_{\mathbb{Z}}$	0	0	0	$4n$
	$+_{\mathbb{Z}}$	0	0	0	$4n$
Signcrypt	DGS	$8n\log q$	$9n\log q+\ell$	$n^2\log^2 q$ $+6n\log q+n$	$2n$
	$\times_{\mathbb{Z}_q}$	$10n^2\log q$ $+2n\log q$ $+n^2+n\ell$	$10n^2\log q$ $+n\log q$ $+n^2+n\ell$	$n\log q(3\log q+8)$ $\cdot O(n\log n)+$ $(2\log q-1)n^2\log q$	$5O(n\log n)$
	$+_{\mathbb{Z}_q}$	$(\lambda+10)n^2\log q$ $+n^2+n\ell-2n$	$(\lambda+11)n^2\log q$ $+n^2+n\ell-5n$ $+4n\log q+\ell$	$n\log q(3\log q+8)$ $\cdot O(n\log n)+n\ell+$ $(2\log q-1)n^2\log q$	$5O(n\log n)$ $+(\aleph+3)n$
	$\times_{\mathbb{Z}}$	$2n^2\log^2 q$ $+n^2\log q$ $+3n\log q+n^2$	$2n^2\log^2 q$ $+n^2\log q$ $+3n\log q+\ell$	$(n\log^3 q+2\log^2 q)$ $\cdot O(n\log n)$	$4\theta(n\log n)$
	$+_{\mathbb{Z}}$	$2n^2\log^2 q$ $+n^2\log q$ $+2n\log q+n^2$	$2n^2\log^2 q$ $+n^2\log q$ $+n\log q$	$(n\log^3 q+2\log^2 q)$ $\cdot O(n\log n)+n\log q\ell$	$4\theta(n\log n)$
	$\times_{\mathbb{R}}$	0	0	$(n\log^3 q+2\log^2 q)$ $\cdot O(n\log n)$	$4\theta(n\log n)$
	$+_{\mathbb{R}}$	0	0	$(n\log^3 q+2\log^2 q)$ $\cdot O(n\log n)$	$4\theta(n\log n)$
UnSigncrypt	DGS	0	$3nk\ell$	$2n\log q$	$3n$
	$\times_{\mathbb{Z}_q}$	$8n^2\log q$ $+n\ell$	$n^2\log^2 q$ $+n^2\log q(\ell+11)$ $+n^2(\ell+1)$ $+2n\log q\ell$	$(8\log q+1)O(n\log n)$ $+O(n\log q\log(n\log q))$	$4O(n\log n)$ $+O(2n\log 2n)$
	$+_{\mathbb{Z}_q}$	$8n^2\log q$ $+\lambda n\log q$ $+n\ell-n$	$n^2\log^2 q+n^2$ $\log q(\lambda+\ell+11)$ $+n^2(\ell+1)$ $+2n\log q(2\ell+1)$	$(8\log q+1)O(n\log n)$ $+O(n\log q\log(n\log q))$ $+6n\log q$	$4O(n\log n)$ $+O(2n\log 2n)$ $+\aleph n$
	$\times_{\mathbb{Z}}$	$2n^2\log^2 q$ $+8n\log q$ $+2\log q$	$2n^2\log^2 q\ell$ $+n^2\log q\ell$ $+2n\log q\ell$ $+5n\log q+2\ell$	$\log qO(n\log n)$ $+2n\log q$	$4\theta(n\log n)$ $+5n$
	$+_{\mathbb{Z}}$	$2n^2\log^2 q$ $+6n\log q$ $+2\log q$	$2n^2\log^2 q\ell$ $+n^2\log q\ell$ $+2n\log q\ell$ $+5n\log q+2\ell$	$\log qO(n\log n)$ $+2n\log q$	$4\theta(n\log n)$ $+3n$
	$\times_{\mathbb{R}}$	0	0	$\log qO(n\log n)$ $+2n\log q$	$4\theta(n\log n)$
	$+_{\mathbb{R}}$	0	0	$\log qO(n\log n)$ $+2n\log q$	$4\theta(n\log n)$

Next, the overhead of the signcryption is compared. The $\times_{\mathbb{Z}_q}$ and $+_{\mathbb{Z}_q}$ operations in SC-NTRU are in the order of magnitude of $1/n$ of YWW+13 [17] and SS18 [19], respectively. The

numbers of $\times_{\mathbb{Z}}$ and $+_{\mathbb{Z}}$ of SC-NTRU are in the order of magnitude of $1/(n \log q)$ of those of YWW+13 [17] and SS18 [19], and at a lower order of magnitude of YCL+19 [20]. The number of DGS of SC-NTRU is no more than $1/(4 \log q)$ of those of YWW+13 [17] and SS18 [19], and at a lower order of magnitude of YCL+19 [20]. Compared with YWW+13 [17] and SS18 [19], SC-NTRU requires $(4\theta(n \log n))$ additional $\times_{\mathbb{R}}$ and $+_{\mathbb{R}}$, and they are about $4/(n \log^3 q)$ of YCL+19 [20]. In summary, although the schemes of YWW+13 [17] and SS18 [19] are built on the ordinary lattice, their signcryption performance significantly outperforms that of YCL+19 [20], due to not requiring the expensive basis extension. However, their signcryption cost is several orders of magnitude higher than that of SC-NTRU.

Finally, the cost of unsigncryption is compared. The numbers of $\times_{\mathbb{Z}_q}$ and $+_{\mathbb{Z}_q}$ operations of SC-NTRU are in the order of magnitude of $1/n$ and $1/(n \log q + \ell)$ of those of YWW+13 [17] and SS18 [19], respectively. The numbers of $\times_{\mathbb{Z}}$ and $+_{\mathbb{Z}}$ operations of SC-NTRU are in the order of magnitude of $1/(n \log q)$ and $1/(n \log q \ell)$ of those of YWW+13 [17] and SS18 [19], respectively. Compared with YWW+13 [17], SC-NTRU needs $3n$ additional DGS; however, it is only $1/(k\ell)$ of that of SS18 [19]. The numbers of DGS, $\times_{\mathbb{Z}_q}$ and $+_{\mathbb{Z}_q}$, $\times_{\mathbb{Z}}$ and $+_{\mathbb{Z}}$, $\times_{\mathbb{R}}$ and $+_{\mathbb{R}}$ of SC-NTRU are in the order of magnitude of $3/(2 \log q)$, $1/\log q$, $4/\log q$, and $4/\log q$ of those of YCL+19 [20]. In summary, since YCL+19 [20] adopts the ideal lattice, the performance of the unsigncryption greatly outperforms YWW+13 [17] and SS18 [19]; however, its overheads are $\log q/4$ to $2 \log q/3$ of those of SC-NTRU.

To sum up, due to the efficiency of the NTRU trapdoor and reasonable construction, the proposed scheme achieves orders of magnitude of improvement in computation cost over the existing signcryption schemes.

5.3. Experiment

To assess the actual performance of SC-NTRU, we conducted experiments using the C programming language. The experimental environment is set up on a Ubuntu platform with 4 GB of memory. The dimension of the NTRU lattice has a significant impact on security. Therefore, we choose three sets of typical parameters: n = 256, 512, and 1024, corresponding to low, medium, and high levels of security, respectively. We run the experiment 1000 times and calculate the average time. The experimental data are presented in Table 5. According to the data in the table, the running time of key generation, signcrypt, and unsigncrypt exhibit running times on the order of milliseconds. Notably, the running time of signcrypt (and unsigncrypt) ranges between 1.3 and 6.2 (1.1 and 5.5), demonstrating the efficiency of SC-NTRU.

Table 5. Actual performance of SC-NTRU under different parameters.

n	q	Keygen (ms)	SC (μs)	USC (μs)
256	52,145,447,681	16.45	1342.48	1198.03
512	425,478,982,619	36.17	2904.77	2585.58
1024	3,470,791,299,527	95.81	6134.78	5436.07

6. Conclusions

In this paper, a signcryption scheme following the StE paradigm is proposed based on the intractability of the NTRU lattice and RLWE, which serves as the security foundation of Falcon in NIST PQC. First, it is shown how to embed some sensitive information into a general lattice-based public key encryption (PKE) and bind it with the message being encrypted by PKE. The malleability to the ciphertext ultimately leads to the modification in the message–signature pair. Consequently, the signature for the message can also be utilized to verify and guarantee the IND-CCA2 security of the ciphertext. Thus, the need for the MAC to transfer from the public key to the signature is eliminated.

Secondly, a new abort-resistant hash is proposed to match the "partiality" of the pre-image in relation to the checkout polynomial, so that an NTRU signature secure in the

standard model can be built with it. The computational overhead analysis demonstrates a significant improvement in the efficiency of SC-NTRU, surpassing existing lattice-based signcryption methods by orders of magnitude. The experiment shows that SC-NTRU is very efficient.

Author Contributions: Methodology, J.Y., X.L., M.L. and L.W.; Validation, J.Y., X.L., J.Z. and W.Y.; Formal analysis, J.Y., L.W. and W.Y.; Investigation, M.L., J.Z. and W.Y.; Writing—original draft, J.Y. and X.L.; Writing—review & editing, M.L.; Supervision, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Defense Basic Scientific Research program of China (Grant No. JCKY2020602B008), the Study Abroad Foundation Visiting Program (No. 201908370041), the Natural Science Foundation of Shandong Province (No. ZR2017MF035), and the National Natural Science Foundation of China (NSFC) (No. 61972050).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, Y. Digital Signcryption or How to Achieve Cost(Signature & Encryption) «Cost(Signature) + Cost(Encryption)». In *Proceedings of the Advances in Cryptology—CRYPTO'97, 17th Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 1997*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1997; Volume 1294, pp. 165–179. [CrossRef]
2. Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
3. Li, F.; Bin Muhaya, F.T.; Khan, M.K.; Takagi, T. *Lattice-Based Signcryption*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2012. [CrossRef]
4. Rao Sreenivasa, Y. A secure and efficient Ciphertext-Policy Attribute-Based Signcryption for Personal Health Records sharing in cloud computing. *Future Gener. Comput. Syst. FGCS* **2017**, *52*, 95–108.
5. Deng, F.; Wang, Y.; Li, P.; Hu, X.; Geng, J.; Qin, Z. Ciphertext-Policy Attribute-Based Signcryption with Verifiable Outsourced Designcryption for Sharing Personal Health Records. *IEEE Access* **2018**, *6*, 39473–39486. [CrossRef]
6. Yan, J.; Wang, L.; Dong, M.; Yang, Y.; Yao, W. Identity-based signcryption from lattices. *Secur. Commun. Netw.* **2015**, *8*, 3751–3770. [CrossRef]
7. Yan, J.; Wang, L.; Li, M.; Ahmad, H.; Yue, J.; Yao, W. Attribute-Based Signcryption from Lattices in the Standard Model. *IEEE Access* **2019**, *7*, 56039–56050. [CrossRef]
8. Zhang, X.; Xu, C.; Xue, J. Efficient multi-receiver identity-based signcryption from lattice assumption. *Int. J. Electron. Secur. Digit. Forensics* **2018**, *10*, 20. [CrossRef]
9. Wang, F.; Hu, Y.; Wang, C. Post-quantum secure hybrid signcryption from lattice assumption. *Appl. Math. Inf. Sci.* **2012**, *6*, 23–28.
10. Lu, X.; Wen, Q.; Wang, L.; Du, J. A Lattice-based Signcryption Scheme without Trapdoors. *J. Electron. Inf.* **2016**, *38*, 2287. [CrossRef]
11. Gérard, F.; Merckx, K. *SETLA: Signature and Encryption from Lattices*; Springer: Cham, Switzerland, 2018.
12. Liu, Z.; Han, Y.L.; Yang, X.Y. A Signcryption Scheme Based Learning with Errors over Rings without Trapdoor. In Proceedings of the National Conference of Theoretical Computer Science, Lanzhou, China, 2–4 August 2019.
13. Liu, Z.; Han, Y.; Yang, X.; Yang, S. Provable security signcryption scheme based on RLWE without trapdoor. *J. Commun.* **2020**, *41*, 12.
14. Savu, L. Signcryption scheme based on schnorr digital signature. *arXiv* **2012**, arXiv:1202.1663.
15. Bai, S.; Galbraith, S.D. An improved compression technique for signatures based on learning with errors. *IACR Cryptol. EPrint Arch.* **2013**, *2013*, 838.
16. Güneysu, T.; Lyubashevsky, V.; Pöppelmann, T. Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems. In *Proceedings of the CHES*; Lecture Notes in Computer Science; Prouff, E., Schaumont, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7428, pp. 530–547.
17. Yan, J.; Wang, L.; Wang, L.; Yang, Y.; Yao, W. Efficient Lattice-Based Signcryption in Standard Model. *Math. Probl. Eng.* **2013**, *2013*, 702539. [CrossRef]
18. Micciancio, D.; Peikert, C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *Proceedings of the Advances in Cryptology—EUROCRYPT 2012*; Lecture Notes in Computer Science; Pointcheval, D., Johansson, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7237, pp. 700–718. [CrossRef]
19. Sato, S.; Shikata, J. *Lattice-Based Signcryption without Random Oracles*; Springer: Cham, Switzerland, 2018.
20. Yang, X.; Cao, H.; Li, W.; Xuan, H. Improved Lattice-Based Signcryption in the Standard Model. *IEEE Access* **2019**, *7*, 155552–155562. [CrossRef]

21. Lyubashevsky, V.; Peikert, C.; Regev, O. On Ideal Lattices and Learning with Errors over Rings. In *Proceedings of the EUROCRYPT*; Lecture Notes in Computer Science; Gilbert, H., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6110, pp. 1–23.
22. Peikert, C. Lattice Cryptography for the Internet. In *Proceedings of the PQCrypto*; Mosca, M., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8772, pp. 197–219.
23. Zhang, J.; Zhang, Z.; Ding, J.; Snook, M. Authenticated Key Exchange from Ideal Lattices. *IACR Cryptol. EPrint Arch.* **2014**, *2014*, 589.
24. Liu, Z.Y.; Tso, R.; Tseng, Y.F.; Mambo, M. Signcryption from NTRU Lattices without Random Oracles. In Proceedings of the 2019 14th Asia Joint Conference on Information Security (AsiaJCIS), Kobe, Japan, 1–2 August 2019.
25. del Pino, R.; Lyubashevsky, V.; Pointcheval, D. The Whole is Less Than the Sum of Its Parts: Constructing More Efficient Lattice-Based AKEs. In Proceedings of the Security and Cryptography for Networks—10th International Conference, SCN 2016, Amalfi, Italy, 31 August–2 September 2016; Lecture Notes in Computer Science; Zikas, V., Prisco, R.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9841, pp. 273–291.
26. Zhang, Y.; Hu, Y.; Xie, J.; Jiang, M. Efficient ring signature schemes over NTRU Lattices. *Secur. Commun. Netw.* **2016**, *9*, 5252–5261. [CrossRef]
27. An, J.H.; Dodis, Y.; Rabin, T. On the Security of Joint Signature and Encryption. In *Proceedings of the Advances in Cryptology—EUROCRYPT 2002*; Knudsen, L.R., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 83–107.
28. Matsuda, T.; Matsuura, K.; Schuldt, J.C.N. Efficient Constructions of Signcryption Schemes and Signcryption Composability. In Proceedings of the International Conference on Cryptology in India, New Delhi, India, 13–16 December 2009.
29. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Stehlé, D. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2018**, *2018*, 238–268. [CrossRef]
30. Fouque, P.A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Zhang, Z. Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU. *Submiss. NIST's Post-Quantum Cryptogr. Stand. Process* **2018**, *36*, 1–75.
31. Joseph, D.; Misoczki, R.; Manzano, M.; Tricot, J.; Pinuaga, F.D.; Lacombe, O.; Leichenauer, S.; Hidary, J.; Venables, P.; Hansen, R. Transitioning organizations to post-quantum cryptography. *Nature* **2022**, *605*, 237–243. [CrossRef]
32. Waters, B. Efficient Identity-Based Encryption without Random Oracles. In *Proceedings of the Advances in Cryptology—EUROCRYPT 2005*; Lecture Notes in Computer Science; Cramer, R., Ed.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3494, pp. 114–127. [CrossRef]
33. Agrawal, S.; Boneh, D.; Boyen, X. Efficient Lattice (H)IBE in the Standard Model. In *Advances in Cryptology—EUROCRYPT 2010*; Lecture Notes in Computer Science; Gilbert, H., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6110, pp. 553–572. [CrossRef]
34. Boyen, X. Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More. In *Public Key Cryptography—PKC 2010*; Lecture Notes in Computer Science; Nguyen, P., Pointcheval, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6056, pp. 499–517. [CrossRef]
35. Chen, Y.; Genise, N.; Mukherjee, P. Approximate Trapdoors for Lattices and Smaller Hash-and-Sign Signatures. In *Proceedings of the Advances in Cryptology—ASIACRYPT 2019—25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, 8–12 December 2019*; Proceedings, Part III; Lecture Notes in Computer Science; Galbraith, S.D., Moriai, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11923, pp. 3–32.
36. López-Alt, A.; Tromer, E.; Vaikuntanathan, V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, 19–22 May 2012*; Karloff, H.J., Pitassi, T., Eds.; ACM: New York, NY, USA, 2012; pp. 1219–1234.
37. Hoffstein, J.; Pipher, J.; Silverman, J.H. NTRU: A Ring-Based Public Key Cryptosystem. In Proceedings of the ANTS: 3rd International Algorithmic Number Theory Symposium (ANTS), Portland, OR, USA, 21–25 June 1998.
38. Ducas, L.; Lyubashevsky, V.; Prest, T. Efficient Identity-Based Encryption over NTRU Lattices. *IACR Cryptol. EPrint Arch.* **2014**, *2014*, 794.
39. Lyubashevsky, V. Lattice Signatures without Trapdoors. In *Advances in Cryptology—EUROCRYPT 2012*; Lecture Notes in Computer Science; Pointcheval, D., Johansson, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7237, pp. 738–755. [CrossRef]
40. Micciancio, D.; Regev, O. Worst-case to average-case reductions based on Gaussian measure. *SIAM J. Comput.* **2007**, *37*, 267–302. [CrossRef]
41. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC '08, New York, NY, USA, 17–20 May 2008; pp. 197–206. [CrossRef]
42. Peikert, C. An Efficient and Parallel Gaussian Sampler for Lattices. In *Advances in Cryptology—CRYPTO 2010*; Lecture Notes in Computer Science; Rabin, T., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6223, pp. 80–97. [CrossRef]
43. Zhang, J.; Yu, Y.; Fan, S.; Zhang, Z. Improved lattice-based CCA2-secure PKE in the standard model. *Sci. China Inf. Sci.* **2020**, *63*, 182101. [CrossRef]

44. Micciancio, D.; Goldwasser, S. *Complexity of Lattice Problems: A Cryptographic Perspective*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 671.
45. Ducas, L.; Prest, T. Fast Fourier Orthogonalization. In Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC '16, New York, NY, USA, 20–22 July 2016; pp. 191–198. [CrossRef]