Identity-Concealed Authenticated Encryption from Ring Learning With Errors (Full version)

Chao Liu[1], Zhongxiang Zheng[2], Keting Jia[2(✉)], and Limin Tao[3]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, P.R. China
liu_chao@mail.sdu.edu.cn
[2] Department of Computer Science and Technology, Tsinghua University, P.R. China
ktjia@mail.tsinghua.edu.cn
[3] Space Star Technology co., LTD, P.R. China

Abstract. *Authenticated encryption* (AE) is very suitable for a resources constrained environment for it needs less computational costs and AE has become one of the important technologies of modern communication security. *Identity concealment* is one of research focuses in design and analysis of current secure transport protocols (such as TLS1.3 and Google's QUIC). In this paper, we present a provably secure identity-concealed authenticated encryption in the public-key setting over ideal lattices, referred to as RLWE-ICAE. Our scheme can be regarded as a parallel extension of higncryption scheme proposed by Zhao (CCS 2016), but in the lattice-based setting. RLWE-ICAE can be viewed as a monolithic integration of public-key encryption, key agreement over ideal lattices, identity concealment and digital signature. The security of RLWE-ICAE is directly relied on the Ring Learning with Errors (RLWE) assumption. Two concrete choices of parameters are provided in the end.

Keywords: Authenticated encryption · RLWE · Lattice-based · Identity-concealed · Provable security

1 Introduction

Authenticated encryption (AE) is a form of encryption that guarantees the confidentiality and authenticity of data at the same time. Because AE can sign and encrypt messages in single step, the computational cost of it is lower than that of traditional signature-then-encryption methods. Some works also shows that AE is functionally equivalent to one-pass authenticated key-exchange [21,8,12]. Since Zheng proposed the first AE scheme [31] in 1997, it has become one of the important technologies of modern communication security.

By *identity concealment*, we mean that the protocol transcript shouldn't leak participants' identity information. ID concealment is relevant for several reasons. For instance, if the identity is not protected in a wireless device, an attacker can eavesdrop the communications to track the user's location, which leads to attacks directed towards selected users. Identity concealment is mandated or recommended by many standardized and deployed cryptographic protocols like TLS1.3 [24], QUIC [26], EMV [6], etc. Furthermore, we say that a player enjoys *forward ID-privacy* if his ID-privacy preserves even through his static secret-key is compromised. For some famous protocols such as Zheng's signcryption [31,3] and one-pass HMQV (HOMQV) [15,13], the issue of ID-concealment was not considered. In 2016, Zhao [30] introduced that ID-concealment can be integrated with AE to solve the problem of *0-RTT (zero-round trip time) with client authentication.* A *0-RTT option* protocol allows the establishment of a secure connection in "one-shot", which means that cryptographically protected payload data can be sent immediately along with the first single message sent from a sender to a receiver, without the need for a latency-incurring prior handshake protocol. This significant acceleration of connection establishment provides a more smoothly Web browsing experience and better performance for applications with low latency requirements. Many large projects have been developed and experimented with 0-RTT protocols, such as Google's QUIC [16], TLS1.3 and Facebook' Zero protocols [14]. But QUIC and TLS1.3 are now only supporting 0-RTT mode *without* client authentication. Zhao proposed higncryption [30] which solved the problem of 0-RTT with client authentication by integrating public-key encryption, entity authentication and ID-concealment into a single primitive.

Some other properties are considered in nowadays public-key settings. A protocol enjoys *"receiver deniability"*, which means that the session transcript, especially the authentication value, can be simulated by a receiver with public parameters and his own secret-key. A protocol enjoys *x-security* [13], which means that the leakage of ephemeral secret does not cause the exposure of sender's static secret or pre-shared secret. For some well-known protocols, Zheng's signcryption [31,3] does not enjoy x-security and is receiver undeniable. Krawczyk's one-pass HMQV (HOMQV) [13] scheme enjoys receiver deniability and x-security, but without forward ID-privacy. Zhao's higncryption [30] has a novel design, and enjoys forward ID-privacy, receiver deniability and x-security.

But above existed authenticated encryptions are mainly based on the classic hard problems, such as the computational/decisional DH problem. It is well known that DH problem is vulnerable to quantum computers [27]. Since the rapid development of quantum computers, searching other counterparts based on problems which are believed to be resistant to quantum attacks is more and more urgent. Naturally we think of such a question: can we come up with an authenticated encryption that can resist quantum attacks and enjoys above several good properties such as ID-concealment, receiver deniability and x-security? Note that lattice-based cryptographic schemes have many advantages such as asymptotic efficiency, conceptual simplicity and worst-case hardness assumption, and it is a perfect choice to build lattice-based authenticated encryption in the public-key settings.

Our Contributions. In this paper, we propose a new authenticated encryption to solve the above motivating questions. We choose Ring Learning With Errors (RLWE), which is as hard as some worst case lattice problems on ideal lattices [19,11] to construct our scheme. By utilizing some useful properties of RLWE and discrete Gaussian distributions, we present an approach to combine public/secret key in a manner similar to higncryption [30]. Our scheme not only enjoys many nice properties of higncryption such as identity concealment, 0-RTT option, forward ID-privacy, receiver deniability and x-security, but also enjoys some properties of lattice-based cryptography, such as worst-case hardness assumption, and resistance to quantum computer attacks. We manage to establish a full proof of our scheme' security in the Zhao's strong model [30] by replacing the Diffie-Hellman core of Zhao's model with the lattice-based core. Our scheme may have some other applications. For example we give a direct application of one-pass ID-concealed authenticated key exchange protocol. In the end, we choose the concrete parameters and give the security assessment.

Techniques in Our Scheme. In higncryption, the sender (the encryption party) and the receiver (the decryption party) would compute a same element, which is used in encrypting communication data. Since higncryption works on "nicely-behaved" cyclic groups, which have the property of commutativity, such a "key agreement" can be easily realized. While for lattice-based cryptographic, benefitting from the growth of lattice-based key exchange protocols [9,23,5], we can utilize the key agreement technique to construct our scheme. Ding et al. [9] firstly introduced the key reconciliation mechanism to "handling the noises" of RLWE. And Peikert [23] gave an improved version of reconciliation mechanism. We use Peikert's reconciliation mechanism to achieve the key agreement in our scheme. Furthermore, since the perfect randomization properties of cyclic groups, the static key can be "perfectly hidden" in the communication data. While for RLWE based scheme, the goal of perfectly hiding the keys can be realized by using rejection sampling [17]. In the security aspect, secret hidden is necessary, so we apply the rejection sampling technique in our scheme. To prove the security of our scheme, we introduce vPWE assumption, which is a variant of Pairing with Errors (PWE) assumption introduced by Ding et al. [10], and we show that vPWE assumption can be reduced to the RLWE problem. As long as the vPWE assumption is hard, the security of our scheme can be guaranteed.

Related Works. For authenticated protocols from ideal lattices, in 2015, Zhang et al. [29] proposed an authenticated RLWE based key exchange and a one-pass authenticated key exchange over ideal lattices. In 2017, Ding et al. [10] proposed RLWE-based password authenticated key exchange, whose security is proved by using PWE assumption. Yang et al. [28] introduced a RLWE-based two-message key exchange scheme in 2018, and they used Peikert's reconciliation mechanism to construct the scheme.

Roadmap. In Sect. 2, we introduce some backgrounds such as notations, security models, RLWE and some tools used in scheme. Our protocol RLWE-ICAE is introduced in Sect. 3. And in Sect. 4, two theorems are given to guarantee the security of the scheme. The parameters and the security assessment of our scheme are presented in Sect. 5. Finally, we conclude and discuss some further works in Sect. 6.

2 Preliminaries

2.1 Notations

Let n be an integer of the power of 2. Denote the ring of integer polynomials R as $\mathbb{Z}[x]/(x^n+1)$, and $R_q := \mathbb{Z}_q[x]/(x^n+1)$ as the ring of integer polynomials modulo x^n+1 with every coefficient is reduced modulo positive integer q. Let the norm of a polynomial be the norm of its coefficients vector. Let $x \overset{\$}{\leftarrow} \chi$ denote the coefficients of x are sampled based on the probability distribution χ. For any positive real $\beta \in \mathbb{R}$, and a vector $\mathbf{c} \in \mathbb{R}^m$, let the continuous Gaussian distribution over $\mathbb{R}^m$ with standard deviation β centered at $\mathbf{c}$ be defined by the probability function $\rho_{\beta,\mathbf{c}}(\mathbf{x}) = (\frac{1}{\sqrt{2\pi\beta^2}})^m exp(\frac{-||\mathbf{x}-\mathbf{v}||_2^2}{2\beta^2})$. Let $D_{\mathbb{Z}^n,\beta,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\beta,\mathbf{c}}(\mathbf{x})}{\rho_{\beta,\mathbf{c}}(\mathbb{Z}^m)}$ to indicate the m-dimensional discrete Gaussian distribution. The subscripts β and $\mathbf{c}$ are omitted when they are 1 and $\mathbf{0}$. Usually χ_β denotes Gaussian distribution with standard deviation β and centered at 0.

2.2 Authenticated Encryption with Associated Data

An *authenticated encryption with associated date* (AEAD) scheme transforms a message M and a public packet header, which is usually implicitly determined from the context, into a ciphertext C which provides both privacy (of M) and authenticity (of C and H) [25]. We state the security of AEAD in [30] as follows.

AEAD security. Let $\prod = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. The key space $\mathcal{K} = \{0,1\}^\kappa$ is a finite nonempty set of strings. There is a probabilistic polynomial-time algorithm takes a security parameter κ as input and samples a key K from $\mathcal{K}$. The polynomial-time encryption algorithm $\mathcal{E} : \kappa \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ and the polynomial-time decryption algorithm $\mathcal{D} : \kappa \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ satisfy:

$$\Pr[K \leftarrow \mathcal{K}; H \in \{0,1\}^*; M \in \{0,1\}^*; C \leftarrow \mathcal{E}_K(H,M) : \mathcal{D}_K(C) \neq M] \leq negl(\kappa),$$

where $negl$ is a negligible function. Generally, we assume the ciphertext C has the associate data H. Let $\mathcal{A}$ be a polynomial-time adversary. A security game for AEAD is described in Table 1. The advantage of $\mathcal{A}$ is defined to be $\mathrm{Adv}_{\prod}^{aead}(\mathcal{A}) = |2 \cdot \Pr[\mathrm{AEAD}_{\prod}^{\mathcal{A}}$ returns $\mathbf{true}] - 1|$. And we say $\prod$ scheme is AEAD-secure, if for all sufficiently large κ, $\mathrm{Adv}_{\prod}^{\mathrm{AEAD}}(\mathcal{A}) \leq negl(\kappa)$.

2.3 Security Model for ICAE

We recall the security model for identity-concealed authenticated encryption (ICAE) scheme from [30]. An ICAE scheme $\mathcal{IC}$ is specified with four polynomial-time algorithms (**Setup**, **Keygen**, **Encrypt**, **Decrypt**) as follows:

Table 1. AEAD security game

main $\mathrm{AEAD}_{\Pi}^{\mathcal{A}}$:	**procedure Enc**(H, M_0, M_1):	**procedure Dec**(C'):
$K \leftarrow \mathcal{K}$	If $\lvert M_0 \rvert \neq \lvert M_1 \rvert$, Ret $\perp$	If $\sigma = 1 \wedge C' \notin \mathcal{C}$ then
$\sigma \leftarrow \{0, 1\}$	$C_0 \leftarrow \mathcal{E}_K(H, M_0)$	Ret $\mathcal{D}_K(C')$
$\sigma' = \mathcal{A}^{\mathbf{Enc},\mathbf{Dec}}$	$C_1 \leftarrow \mathcal{E}_K(H, M_1)$	else Ret $\perp$
Ret $(\sigma' = \sigma)$	If $C_0 = \perp$ or $C_1 = \perp$, Ret $\perp$	
	$\mathcal{C} \overset{\cup}{\leftarrow} C_\sigma$; Ret C_σ	

- **Setup**: takes the security parameter κ as input and outputs the system parameter *params* used in the scheme.
- **Keygen**: takes *params* as input and outputs a key pair (pk, sk) used for encryption and decryption.
- **Encrypt**: takes the sender's private key sk_s and public identity information $pid_s = (id_s, pk_s, cert_s)$ where $cert_s$ is issued by a certificate authority, a receiver's public identity information $pid_r = (id_r, pk_r, cert_r)$, message $M \in \{0, 1\}^*$, and associated data $H \in \{0, 1\}^*$ as input. It returns a ciphertext C or $\perp$ which indicate encrypt failure. We allow $pid_s = (id_s, pk_s, cert_s)$ equal to $pid_r = (id_r, pk_r, cert_r)$, which means that a user encrypts a message to himself. We also assume some *offline-computable* intermediate randomness used in generating C is stored in a variable $\mathcal{ST}_C$.
- **Decrypt**: takes a receiver's private key sk_r, the receiver's public identity information $pid_r = (id_r, pk_r, cert_r)$, a ciphertext C as input. It outputs (pid_s, M) or an error $\perp$.

We say that an ICAE scheme is correctness if for all sufficiently large security parameter κ, key pairs (pk_s, sk_s) and (pk_r, sk_r) which are output by **Keygen**(1^κ), there is

$$\Pr[\mathbf{Decrypt}(sk_r, pid_r, \mathbf{Encrypt}(sk_s, pid_s, pid_r, H, M)) \neq (pid_s, M)] \leq negl(\kappa)$$

where $H, M \in \{0, 1\}^*$ such that $\mathbf{Encrypt}(sk_s, pid_s, pid_r, H, M) \neq \perp$, and *negl* is a negligible function.

Now we present the security model for ICAE. We assume each user possesses a single key pair for encryption and decryption, and each user can encrypt messages to himself. In this model the adversary is allowed to register users adaptively (hence has dishonest users). Let the number of users in the system be N, which is a polynomial in the security parameter κ. We assume all the honest users' key pairs are generated by the challenger according to the key generation algorithm specified in the system. Denote by **HONEST** (reps., **DISHONEST**), the set of public identity information of all the honest (resp., dishonest) users. We denote the public identity information of a user id_i as pid_i $(1 \leq i \leq n)$, the sender's (resp., the receiver's) public identity information as pid_s (resp., pid_r). The adversary's abilities are formalized by providing the adversary with the following oracles:

- **ENO**: takes (pid_s, pid_r, H, M) as inputs, where $pid_r \in \mathbf{HONEST} \bigcup \mathbf{DISH-ONEST}$. If $pid_s \in \mathbf{HONEST}$, the oracle returns $\mathbf{Encrypt}(sk_s, pid_s, pid_r,$

$H, M)$, otherwise return $\perp$. In order to allow for later **Exposure** query against a ciphertext C, some specified offline-computable intermediate randomness to generate C are allowed to be stored into $\mathcal{ST}_C$.

- **DEO**: takes (pid_r, C) as inputs. If $pid_r \in$ **HONEST**, the oracle returns **Decrypt** (sk_r, pid_r, C), otherwise, returns $\perp$.
- **Exposure**: takes $C \neq \perp$ as input. If C is output by an earlier **ENO** query, the oracle returns the offline-computable intermediate randomness (stored in $\mathcal{ST}_C$) used in generating C.
- **Corrupt**: takes $pid_i \in$ **HONEST** as input, $(1 \leq i \leq N)$, and returns user id_i's private key sk_i.

Outsider unforgeability. Consider the following experiment for $\mathcal{A}^{OU}$:
The encryption experiment Encry-forge$_{\mathcal{A}^{OU},\mathcal{IC}}(\kappa)$**:**

- $\mathcal{A}^{OU}$ is given the all the honest users' public keys and can register arbitrary public keys on its own with security parameter κ.
- $\mathcal{A}^{OU}$ is allowed to issue **ENO**, **DEO**, **Exposure** and **Corrput** queries. $\mathcal{A}^{OU}$ then outputs (pid_{r^*}, C^*) as its output.
- $\mathcal{A}^{OU}$ **succeeds** if and only if:
  1. **Decrypt**$(sk_{r^*}, pid_{r^*}, C^*) = (pid_{s^*}, M^*)$, where $pid_{s^*} \in$ **HONEST**;
  2. $\mathcal{A}^{OU}$ has not issued **Corrupt**(pid_{s^*}) or **Corrupt**(pid_{r^*}) query, but is allowed to query **Exposure**(C^*) to expose the intermediate randomness in generating C^*.
  3. C^* is not the output of **ENO**$(pid_{s^*}, pid_{r^*}, H^*, M^*)$ issued by $\mathcal{A}^{OU}$, but $\mathcal{A}^{OU}$ is still allowed to query **ENO**$(pid_{s'}, pid_{r'}, H', M')$ for $(pid_{s'}, pid_{r'}, H', M') \neq (pid_{s^*}, pid_{r^*}, H^*, M^*)$ and in particular $(pid_{s^*}, pid_{r^*}, H', M^*)$ for $H' \neq H^*$. $\mathcal{A}^{OU}$ can even query **ENO**$(pid_{s^*}, pid_{r^*}, H^*, M^*)$ as long as its outputs returned is not C^*. And parts of C^* (the H^*) may appear in previous outputs of **ENO**.
- The experiment returns 1 if $\mathcal{A}^{OU}$ **succeeds**, otherwise returns 0.

We say that an ICAE scheme $\mathcal{IC}$ has *outside unforgeability*, if for any PPT adversary $\mathcal{A}^{OU}$, there is a negligible function *negl* such that:

$$\Pr[\textbf{Encry-forge}_{\mathcal{A}^{OU},\mathcal{IC}}(\kappa) = 1] \leq negl(\kappa).$$

Next we introduce the definition of *insider confidentiality*, which is identical to outsider unforgeability, except that **Corrupt**(pid_{r^*}) is allowed to the adversary.

Insider confidentiality. We assume that all the users have equal length public identity information. Consider the following experiment for an adversary $\mathcal{A}^{IC}$:
The encryption experiment Encry-Confident$_{\mathcal{A}^{IC},\mathcal{IC}}(\kappa)$**:**

- $\mathcal{A}^{IC}$ is given the all the honest users' public keys and can register arbitrary public keys on its own with security parameter κ.

- $\mathcal{A}^{IC}$ is allowed to issue **ENO**, **DEO**, **Exposure** and **Corrput** queries. $\mathcal{A}^{IC}$ then outputs two equal length messages (M_0, M_1), an associated data H^*, and two pairs of public identity information of equal length $(pid_{s_0^*}, pid_{r^*})$ and $(pid_{s_1^*}, pid_{r^*})$ where $pid_{s_0^*}, pid_{s_1^*}, pid_{r^*} \in$ **HONEST**.
- A uniform bit $\gamma \in \{0,1\}$ is chosen, and then a ciphertext $C^* =$ **Encrypt**($sk_{s_\gamma^*}, pid_{s_\gamma^*}, pid_{r^*}, H^*, M_\gamma)$ is computed and given to $\mathcal{A}^{IC}$.
- The adversary $\mathcal{A}^{IC}$ can continue executing the second phase, except asking **DEO**(pid_{r^*}, C^*), **Exposure**(C^*) or **Corrupt**(pid_{r^*}) which will cause $\mathcal{A}^{IC}$ win the game trivially. But the adversary $\mathcal{A}^{IC}$ is allowed to issue **Corrupt**$(pid_{s_0^*})$ and **Corrupt**$(pid_{s_1^*})$, which can capture forward ID-privacy. Eventually, $\mathcal{A}^{IC}$ outputs a bit γ'.
- The output of the experiment is defined to be 1 if $\gamma' = \gamma$, and 0 otherwise. If the output of the experiment is 1, we say that $\mathcal{A}^{IC}$ **succeeds**.

We say that an ICAE scheme $\mathcal{IC}$ has insider confidentiality, if for any PPT adversary $\mathcal{A}^{IC}$ there is a negligible function *negl* such that:

$$\Pr[\textbf{Encry-Confident}_{\mathcal{A}^{IC},\mathcal{IC}}(\kappa) = 1] \le negl(\kappa).$$

Note that the definition of outsider confidentiality is identical to that of insider confidentiality, except that neither **Corrupt**$(pid^*_{s_0})$ nor **Corrupt**$(pid^*_{s_1})$.

2.4 Ring Learning with Errors

In 2010, Lyubashevsky, Peikert and Regev [19] proposed the Ring Learning with Erros problems (RLWE), which is based on the Learning with Errors (LWE) in the ring setting. Assume there are uniform random elements $a, s \overset{\$}{\leftarrow} R_q$ and an error distribution χ. Let $A_{s,\chi}$ denote the distribution of the RLWE pair $(a, as + e)$, where the error $e \overset{\$}{\leftarrow} \chi$. Given polynomial number of samples, the search version of RLWE is to find the secret s, while the decision version of the RLWE problem (DRLWE$_{q,\chi}$) is to distinguish $A_{s,\chi}$ from an uniform distribution pair (a, b) on $R_q \times R_q$. RLWE enjoys a worst case hardness guarantee, which we state here.

Theorem 1. ([19], Theorem 3.6) *Let $R = \mathbb{Z}[x]/(x^n + 1)$ where n is a power of 2, $\delta = \delta(n) < \sqrt{logn/n}$, and $q = 1 \mod 2n$ which is a* ploy(n)*-bounded prime such that $\delta q \ge \omega(\sqrt{logn})$. Then there exists a* ploy(n)*-time quantum reduction from $\tilde{O}(\sqrt{n}/\delta)$-SIVP (Short Independent Vectors Problem) on ideal lattices in the ring R to solve DRLWE$_{q,\chi}$ with $l-1$ samples, where $\chi = D_{\mathbb{Z}^n,\varsigma}$ is the discrete Gaussian distribution with parameter $\varsigma = \delta q \cdot (nl/log(nl))^{1/4}/\sqrt{2\pi}$.*

We have the following useful facts.

Lemma 1. ([17], Lemma 4.4) *For any $k > 0$, $\Pr_{x \leftarrow \chi_\beta}(|x| > k\beta) \le 2e^{-k^2/2}$.*

Note that taking $k = 13$ gives tail probability approximating 2^{-121}.

Lemma 2. ([22]) *Letting real $\beta = \omega(\sqrt{logn})$, constant $\eta > \frac{1}{\sqrt{2\pi}}$, then we have that $Pr_{\mathbf{v} \xleftarrow{\$} D_{\mathbb{Z}^n,\beta}}[||\mathbf{v}|| > \eta \cdot \beta\sqrt{n}] \leq \frac{1}{2}D^n$, where $D = \eta\sqrt{2\pi e} \cdot e^{-\pi \cdot \eta^2}$. In particular, we have $Pr_{\mathbf{v} \xleftarrow{\$} D_{\mathbb{Z}^n,\beta}}[||\mathbf{v}|| > \beta\sqrt{n}] \leq 2^{-n+1}$.*

2.5 The Rejection Sampling

Now, we recall the rejection sampling from [18].

Theorem 2. ([18], Theorem 3.4) *Let S be a subset of $\mathbb{Z}^m$, all the elements of S have norms less than T, $\beta = w(T\sqrt{logm})$ be a real, and $\phi : S \to \mathbb{R}$ be a probability distribution. Then the distribution of the following algorithm $\mathcal{F}$:*

- $\mathbf{c} \xleftarrow{\$} \phi$;
- $\mathbf{z} \xleftarrow{\$} D_{\mathbb{Z}^m,\beta,\mathbf{c}}$;
- *output* $(\mathbf{z}, \mathbf{c})$ *with probability* $min\left(\frac{D_{\mathbb{Z}^m,\beta(\mathbf{z})}}{M \cdot D_{\mathbb{Z}^m,\beta,\mathbf{c}}(\mathbf{z})}, 1\right)$.

is within statistical distance $\frac{2^{-w(logm)}}{M}$ from the distribution of the following algorithm $\mathcal{G}$:

- $\mathbf{c} \xleftarrow{\$} \phi$;
- $\mathbf{z} \xleftarrow{\$} D_{\mathbb{Z}^m,\beta}$;
- *output* $(\mathbf{z}, \mathbf{c})$ *with probability* $\frac{1}{M}$.

where $M = O(1)$ is a constant. Moreover, the probability that $\mathcal{F}$ outputs something is at leat $\frac{1-2^{-w(logm)}}{M}$. More concretely, if $\beta = \eta T$ for any positive η, then $M = e^{12/\eta + 1/(2\eta^2)}$ and the output of algorithm $\mathcal{F}$ is within statistical distance $\frac{2^{-100}}{M}$ of the output of $\mathcal{G}$, and the probability that $\mathcal{F}$ outputs something is at leat $\frac{1-2^{-100}}{M}$.

2.6 Reconciliation Mechanism

Firstly, We recall the reconciliation mechanism proposed by Peikert in [23] for transforming approximate agreement to exact agreement. For integer $q > p \geq 2$, we define the modular rounding function $\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ as $\lfloor x \rceil_p := \lfloor \frac{p}{q} \cdot x \rceil$ and downward-rounded function $\lfloor \cdot \rfloor_p : \mathbb{Z}_q \to \mathbb{Z}_p$ as $\lfloor x \rfloor_p := \lfloor \frac{p}{q} \cdot x \rfloor$.

Even modulus. Let the modulus $q \geq 2$ is even, define two disjoint intervals $I_0 := \{0, 1, \ldots, \lfloor \frac{q}{4} \rceil - 1\}$, $I_1 := \{-\lfloor \frac{q}{4} \rceil, \ldots, -1\} \bmod q$. Then when $v \in (I_0 + \frac{q}{2}) \cup (I_1 + \frac{q}{2})$, $\lfloor v \rceil_2 = 1$, and when $v \in I_0 \cup I_1$, $\lfloor v \rceil_2 = 0$. Here we define the *cross-rounding* function $\langle \cdot \rangle_2 : \mathbb{Z}_q \to \mathbb{Z}_2$ as $\langle v \rangle_2 := \lfloor \frac{4}{q} \cdot v \rfloor \bmod 2$. Obviously, $\langle v \rangle_2 = b \in \{0, 1\}$ such that $v \in I_b \cup (\frac{q}{2} + I_b)$.

Lemma 3. ([23], Claim 3.1) *For $q \geq 2$ is even, if v is uniformly random chosen from $\mathbb{Z}_q$, then given $\langle v \rangle_2$, $\lfloor v \rceil_2$ is uniformly random.*

Define the set $E := [-\frac{q}{8}, \frac{q}{8}) \cap \mathbb{Z}$. Suppose $v, w \in \mathbb{Z}_q$ are sufficiently close, and given w and $\langle v \rangle_2$, we can recover $\lfloor v \rceil_2$ using the reconciliation function rec: $\mathbb{Z}_q \times \mathbb{Z}_2 \to \mathbb{Z}_2$:

$$\mathrm{rec}(w, b) = \begin{cases} 0 & \text{if } w \in I_b + E (\mathrm{mod} q), \\ 1 & \text{otherwise.} \end{cases}$$

Lemma 4. ([23], Claim 3.2) *For $q \geq 2$ is even, if $w = v + e \bmod q$ for some $v \in \mathbb{Z}_q$ and $e \in E$, then* rec*$(w, \langle v \rangle_2)$*$= \lfloor v \rceil_2$.

Odd modulus. When q is odd, Peikert proposed a randomized function dbl: $\mathbb{Z}_q \to \mathbb{Z}_{2q}$ to avoid the bias produced in the rounding function. Let $v \in \mathbb{Z}_q$, function dbl is defined to be $\mathrm{dbl}(v) := 2v - \tilde{e} \in \mathbb{Z}_{2q}$ where $\tilde{e} \in \mathbb{Z}$ is independent of v and uniformly random modulo two. Usually we write v with an overbar to means that $\bar{v} \leftarrow \mathrm{dbl}(v)$.

Lemma 5. ([23], Claim 3.3) *For $q > 2$ is odd, if v is uniformly random chosen from $\mathbb{Z}_q$ and $\bar{v} \leftarrow \mathrm{dbl}(v) \in \mathbb{Z}_{2q}$, then $\lfloor \bar{v} \rceil_2$ is uniformly random given $\langle \bar{v} \rangle_2$.*

Define function HelpRec(X): (1) $\overline{X} \leftarrow \mathrm{dbl}(X)$; (2) $W \leftarrow \langle \overline{X} \rangle_2$, $K \leftarrow \lfloor \overline{X} \rceil_2$; (3) return (K, W).

Note that for $w, v \in \mathbb{Z}_q$, we need apply the appropriated rounding function from $\mathbb{Z}_{2q}$ to $\mathbb{Z}_2$, (which means that $\lfloor x \rceil_p = \lfloor \frac{p}{2q} \cdot x \rceil$, $\langle x \rangle_2 = \lfloor \frac{4}{2q} \cdot x \rfloor$), and similar to rec function. Then if $(K, W) \leftarrow \mathrm{HelpRec}(X)$ and $Y = X + e$ with $||e||_\infty < \frac{q}{8}$, then $\mathrm{rec}(2 \cdot Y, W) = K$. By applying coefficient-wise to the coefficients in $\mathbb{Z}_q$ of a ring elements we also can extend these definitions to R_q. That is, for a ring elements $v = (v_0, \ldots, v_{n-1}) \in R_q$, setting $\lfloor v \rceil_2 = (\lfloor v_0 \rceil_2, \ldots, \lfloor v_{n-1} \rceil_2)$; $\langle v \rangle_2 = (\langle v_0 \rangle_2, \ldots, \langle v_{n-1} \rangle_2)$, $\mathrm{HelpRec}(v) = (\mathrm{HelpRec}(v_0), \ldots, \mathrm{HelpRec}(v_{n-1}))$ and for a binary-vector $b = (b_0, \ldots,\ b_{n-1}) \in \{0, 1\}^n$, setting $\mathrm{rec}(v, b)$=$(\mathrm{rec}(v_0, b_0)$,$\ldots$, $\mathrm{rec}(v_{n-1}, b_{n-1}))$.

2.7 A Variant of Pair With Errors Problem

The vPWE *assumption.* In [10], Ding et.al. propose the Pairing with Errors (P-WE) assumption based on Ding's reconciliation mechanism [9]. Here we proposed a variant of their PWE assumption and we call it vPWE assumption. We replace the Ding's reconciliation mechanism with Peikert's reconciliation mechanism. Let χ_β be a Gaussian distribution for fixed $\beta \in \mathbb{R}^*_+$. For any $(X, s) \in R_q \times R_q$, if $(K, W) \leftarrow$HelpRec$(X \cdot s)$, then set $\tau(X, s) := K = \lfloor \overline{X \cdot s} \rceil_2$. Let $\mathcal{A}$ be probabilistic, polynomial-time algorithm. $\mathcal{A}$ takes inputs of the form (a, X, Y, W), where $(a, X, Y) \in R_q \times R_q \times R_q$ and $W \in \{0, 1\}^n$, and outputs a list of values in $\{0, 1\}^n$. Given s randomly chosen from χ_β, Y which is a "small additive perturbation" of $a \cdot s$, and $W \leftarrow \langle \overline{X \cdot s} \rangle_2$, $\mathcal{A}$'s objective will be outputting the string $\tau(X, s)$.

To states the hardness of vPWE assumption, We define the decision version of vPWE problem vDPWE as follows. If vDPWE is hard, so is vPWE.

Definition 1. (vDPWE) *Given* $(a, X, Y, W, \sigma) \in R_q \times R_q \times R_q \times \{0,1\}^n \times \{0,1\}^n$ *where* $W = \langle \overline{K} \rangle_2$ *for some* $K \in R_q$ *(*$\overline{K} \leftarrow \mathrm{dbl}(K)$*), and* $\sigma = \mathrm{rec}(2 \cdot K, W)$*. The Decision* vPWE *problem* (vDPWE) *is to decide whether* $K = Xs + e_1$*,* $Y = as + e_2$ *for some* s, e_1, e_2 *are drawn from* χ_β*, or* (K, Y) *are uniformly random in* $R_q \times R_q$*.*

In order to show the reduction of the vDPWE problem to the RLWE problem, we would like to introduce a definition to what we called the RLWE-DH problem [10] which can be reduced to RLWE problem.

Definition 2. (RLWE-DH) *Let* R_q *and* χ_β *be defined as above. Given an input ring element* (a, X, Y, K)*, where* (a, X) *is uniformly random in* R_q^2*, The* DRLWE-DH *problem is to decision if* K *is* $Xs + e_1$ *and* $Y = as + e_2$ *for some* $s, e_1, e_2 \overset{\$}{\leftarrow} \chi_\beta$ *or* (K, Y) *are uniformly random in* $R_q \times R_q$*.*

Theorem 3. ([10], Theorem 1) *Let* R_q *and* χ_β *be defined as above, then the* RLWE-DH *problem is hard to solve if* RLWE *problem is hard.*

Theorem 4. *Let* R_q *and* χ_β *be defined as above. The* vDPWE *problem is hard if the* RLWE-DH *problem is hard.*

Proof. Suppose there exists an algorithm D which can solve the vDPWE problem on input (a, X, Y, W, σ) where for some $K \in R_q$, $W = \langle \overline{K} \rangle_2$ and $\sigma = \mathrm{rec}(2 \cdot K, W)$ with non-negligible advantage. By using D as a subroutine, we can build a distinguisher D' on input (a', X', Y', K'), solve the RLWE-DH problem:

- Compute $W = \langle \overline{K'} \rangle_2$ and $\sigma = \mathrm{rec}(2 \cdot K', W)$.
- Run D using the input (a', X', Y', W, σ).
  - If D outputs 1 then K' is $X's + e_1$ for some $e_1 \overset{\$}{\leftarrow} \chi_\beta$ and $Y' = as + e_2$ for some $s, e_1 \overset{\$}{\leftarrow} \chi_\beta$.
  - Else (K', Y') is uniformly random element from $R_q \times R_q$.

Because D solves vDPWE with non-negligible advantage, D' solves RLWE-DH with non-negligible advantage as well, which contradicts RLWE-DH's hardness. □

3 Protocol Construction of Encryption

3.1 The RLWE-ICAE

In this section we present a practical and carefully designed scheme: RLWE-ICAE. The scheme consists of the following four algorithms, **Setup**, **Keygen**, **Encrypt** and **Decrypt**.

Setup: On a security parameter κ, **Setup**(1^κ) returns $params = (n, q, \alpha, \beta, a)$ specifying the underlying ring R_q, Gaussian distribution χ_α, χ_β used in the scheme and public element $a \overset{\$}{\leftarrow} R_q$, where n is a power of 2 and q is an odd prime such that $q \bmod 2n = 1$.

Keygen: On the parameters $params$, for each honest user i, $(1 \leq i \leq N)$, **Keygen** samples $s_i, e_i \xleftarrow{\$} \chi_\alpha$, sets $pk_i = a \cdot s_i + e_i$ and $sk_i = s_i$, and outputs the keypair (pk_i, sk_i). The CA issue a certificate $cert_i$ used to authenticated the binding between user identity id_i and public-key pk_i.

Encrypt: Let $\prod = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AEAD scheme. Let $h : \{0,1\}^* \to \chi_\alpha$ be a cryptographic hash function that always outputs invertible elements in R_q, $M \in \{0,1\}^*$ be the message to be encrypted with an associated data H and $KDF : G \times \{0,1\}^* \to \{0,1\}^\kappa$ be a key derivation function. We denote by Alice the sender with public identity information $pid_A = (id_A, pk_A = p_A = a \cdot s_A + e_A \in R_q, cert_A)$, where $s_A, e_A \xleftarrow{\$} \chi_\alpha$, and secret-key $sk_A = s_A$, and by Bob the receiver with possesses public identity information $pid_B = (id_B, pk_B = p_B = a \cdot s_B + e_B \in R_q, cert_B)$, where $s_B, e_B \xleftarrow{\$} \chi_\alpha$, and secret-key $sk_B = s_B$.

Encrypt$(sk_A, pid_A, pid_B, H, M)$ works as follows:

1. Sample $r, f \xleftarrow{\$} \chi_\beta$ and compute $X = a \cdot r + f \in R_q$;
2. Compute $d = h(X, pid_A, pid_B)$, $\hat{r} = r + s_A d$ and $\hat{f} = f + e_A d$;
3. Go to step 4 with probability $\min(\frac{D_{\mathbb{Z}^{2n},\beta}(\mathbf{v})}{M \cdot D_{\mathbb{Z}^{2n},\beta,\mathbf{v}_1}(\mathbf{v})}, 1)$, where $\mathbf{v} \in \mathbb{Z}^{2n}$ is the coefficient vector of element $\hat{r}$ concatenated with the coefficient vector of $\hat{f}$, and $\mathbf{v}_1 \in \mathbb{Z}^{2n}$ is the coefficient vector of $s_A d$ concatenated with the coefficient vector of $e_A d$; otherwise go back to step 1;
4. Sample $g \xleftarrow{\$} \chi_\beta$, and compute $\widetilde{X} = p_A \cdot d + X$, $PS_A = p_B \cdot (r + s_A d) + g$;
5. Compute $(PS, w) \leftarrow \text{HelpRec}(PS_A)$;
6. Derive key $K_1 = KDF(PS, \widetilde{X} || pid_B)$, where $K_1 \in \mathcal{K}$;
7. Compute $C_{AE} \leftarrow \mathcal{E}_{K_1}(H, pid_A || X || M)$;
8. Finally, send the ciphertext $C = (H, \widetilde{X}, w, C_{AE})$ to the receiver.

Decrypt$(sk_B, pid_B, C(= (H, \widetilde{X},\ w, C_{AE})))$ works as follows:

1. Compute $PS_B = \widetilde{X} \cdot s_B$ and pre-shared secrecy $PS = \text{rec}(2 \cdot PS_B, w)$, and derive the key $K_1 = KDF(PS, \widetilde{X} || pid_B)$;
2. Run $\mathcal{D}_{K_1}(H, C_{AE})$. If $\mathcal{D}_{K_1}(H, C_{AE})$ returns $\perp$, abort; otherwise get $(pid_A = (id_A, p_A,\ cert_A), X, M)$;
3. Compute $d = h(X, pid_A, pid_B)$. If $\widetilde{X}$ equals to $p_A \cdot d + X$ and pid_A is valid, accept (pid_A, M); otherwise, abort.

Our scheme is presented in Figure 1. Note that we use rejection sampling in our scheme, and this technique can protect the secret information $s_A d$ and $e_A d$ from $\widetilde{X} = a \cdot (s_A d + r) + (e_A d + f)$. In our proof of insider confidentiality, such a "secret hidden" is necessary. Reconciliation mechanism is used to compute PS from two approximate values PS_A and PS_B, and this can be regarded to be a key agreement of the sender and the receiver.

One-pass CAKE. In the RLWE-ICAE, there is $K_1 = KDF(PS, \widetilde{X} || pid_B)$. We can redefine KDF to construct an one-pass CAKE. Define $(K_1, K_2) =$

pid_A		pid_B
$pk_A : p_A \leftarrow a \cdot s_A + e_A$		$pk_B : p_B \leftarrow a \cdot s_B + e_B$
$sk_A : s_A$		$sk_B : s_B$
where $s_A, e_A \overset{\$}{\leftarrow} \chi_\alpha$		where $s_B, e_B \overset{\$}{\leftarrow} \chi_\alpha$
$X \leftarrow a \cdot r + f$ where $r, f \overset{\$}{\leftarrow} \chi_\beta$		
$d \leftarrow h(X, pid_A, pid_B)$		
$\widetilde{X} = p_A \cdot d + X$		
$PS_A \leftarrow p_B \cdot (r + s_A \cdot d) + g$		
where $g \overset{\$}{\leftarrow} \chi_\beta$		
$(PS, w) \leftarrow \text{HelpRec}(PS_A)$		
$K_1 \leftarrow KDF(PS, \widetilde{X} \| pid_B)$		
$C_{AE} \leftarrow \mathcal{E}_{K_1}(H, pid_A \| X \| M)$	$\xrightarrow{H, \widetilde{X}, w, C_{AE}}$	$PS_B \leftarrow \widetilde{X} \cdot s_B$
		$PS \leftarrow \text{rec}(2 \cdot PS_B, w)$
		$K_1 \leftarrow KDF(PS, \widetilde{X} \| pid_B)$
		$(pid_A, X, M) \leftarrow \mathcal{D}_{K_1}(H, C_{AE})$
		$d \leftarrow h(X, pid_A, pid_B)$
		Accept if pid_A valid and $\widetilde{X} = p_A \cdot d + X$

Fig. 1. Protocol structure of RLWE-ICAE.

$KDF(PS, \widetilde{X}||pid_B)$. Then to cast the RLWE-ICAE scheme into one-pass identity-concealed authenticated key-exchange (CAKE), we need set the session-key to be K_2 which is computationally independent of the key K_1. Hence the exposure of K_1 does not affect the session key security. Note that a similar scheme is Zhang's one-pass key exchange protocol from ideal lattices [29]. Compared Zhang's protocol, our scheme provides identity concealment.

3.2 Correctness

Note that in protocol, if $\lfloor \overline{PS_A} \rceil_2 = \text{rec}(2 \cdot PS_B, w)$, where $\overline{PS_A} \leftarrow \text{dbl}(PS_A)$, the protocols would be correct. By the definition of the reconciliation mechanism and Lemma 4, there needs to $||PS_A - PS_B||_\infty < \frac{q}{8}$. We have

$$\begin{aligned} PS_A &= p_B(r + s_A d) + g = (as_B + e_B)(r + s_A d) + g \\ &= ads_A s_B + e_B s_A d + ars_B + re_B + g, \\ PS_B &= \widetilde{X} s_B = (p_A d + X)s_B = (as_A d + e_A d + ar + f)s_B \\ &= ads_A s_B + e_A s_B d + ars_B + fs_B, \end{aligned}$$

therefore, we need $||PS_A - PS_B||_\infty = ||e_B s_A d + re_B + g - e_A s_B d - fs_B||_\infty < \frac{q}{8}$ with overwhelming probability.

4 Security for RLWE-ICAE

We assume KDF to be a random oracle. In this section, we show that our scheme satisfies outsider unforgeability and insider confidentiality in the random oracle model, under the AEAD security and the vPWE assumption.

4.1 Security Proof of Outsider Unforgeability

Theorem 5. *The scheme RLWE-ICAE in Figure 1 satisfies outsider unforgeability in the random oracle model, under the AEAD security and the* vPWE *assumption.*

Proof. We proof the lemma by showing that if

$$\Pr[\textbf{Encry-forge}_{\mathcal{A}^{OU},\text{RLWE-ICAE}}(\kappa) = 1] > negl(\kappa)$$

vPWE problem defined in Sect. 2.7 can be solved with non-negligible probability.

Assume that the output ciphertext of $\mathcal{A}^{OU}$ in $\textbf{Encry-forge}_{\mathcal{A}^{OU},\text{RLWE-ICAE}}(\kappa)$ is (pid_{r^*}, C^*). For presentation simplicity, we assume the adversary can correctly guess the honest users (pid_{s^*}, pid_{r^*}) of successful forgery (which occurs with probability at least $\frac{1}{N^2}$). We assume that $\mathcal{A}^{OU}$'s first successful forgery output is (pid_{r^*}, C^*), and it does not query $\textbf{DEO}(pid'_r, C')$ such that (pid'_r, C') is also a valid forgery before $\mathcal{A}^{OU}$ outputting (pid_{r^*}, C^*).

Given $(pk_{s^*} = a \cdot sk_{s^*} + e_{s^*}, pk_{r^*} = a \cdot sk_{r^*} + e_{r^*})$, where $sk_{s^*}, e_{s^*}, sk_{r^*}, e_{r^*} \xleftarrow{\$} \chi_\alpha$. We present the proof for the commonly viewed harder case: when $pid_{s^*} = pid_{r^*}$. The proof can be extended to the case when $pid_{s^*} \neq pid_{r^*}$ straightforwardly. Construct a simulator S which takes $(params, pid_A)$ as input, where the $p_A = as_A + e_A$ is included in pid_A for (s_A, e_A) unknown to S, also given (a, p_A, p_A, W) where $W \leftarrow \langle \overline{p_A s_A} \rangle_2$ ($\overline{p_A s_A}$ $\leftarrow$dbl$(p_A s_A)$, defined in Sect.2.6), and its goal is to compute $\tau(p_A, s_A) = \text{rec}(\overline{p_A s_A}, W)$. Note computing $\tau(p_A, s_A)$ is as hard as breaking vPWE assumption.

To compute $\tau(p_A, s_A)$, S sets the public-key of sender and receiver to be p_A, and sets the other honest users's public and secret keys on its own. Then the simulator can act on behalf of all the honest users except user pid_A. Hence we focus on the simulation of pid_A to deal with oracle queries made by $\mathcal{A}^{OU}$ against pid_A.

To keep the consistency of the random oracle KDF, we use a proof strategy used in [20,10] that when an adversary causes some events to occur, the simulator can set the associated values of that event. Define two events, corresponding to the adversary who makes a KDF inputs guess in **ENO** and **DEO** query. We also define an associated values for those event.

- **Encorrectsk**: for some $\widetilde{X}$, pid_r, $\mathcal{A}$ makes a $KDF(PS, \widetilde{X}||pid_r)$ query, an **ENO** query with input (pid_s, pid_r, H, M), where the latest query is either the KDF query or the **ENO** query, $\langle \overline{PS_{\mathcal{A}}} \rangle_2 = w$, $\lfloor \overline{PS_{\mathcal{A}}} \rceil_2 = PS = \text{rec}(2 \cdot \widetilde{X} \cdot sk_r, w)$, $(\overline{PS_{\mathcal{A}}} \leftarrow \text{dbl}(PS_{\mathcal{A}}))$ where $PS_{\mathcal{A}}$ is generated by $\mathcal{A}$ and sk_r is the secret key of pid_r. The associated value of this event is $(w, KDF(PS, \widetilde{X}||pid_r)(= K_1))$.

- **Decorrectsk**: for some $\widetilde{X}$, pid_r, $\mathcal{A}$ makes a $KDF(PS, \widetilde{X}||pid_r)$ query, an **DEO** query with input $(pid_r, C = (H, \widetilde{X}, w, C_{AE}))$, where the latest query is either the KDF query or the **DEO** query, $\text{rec}(2 \cdot PS_{\mathcal{A}}, w) = PS = \text{rec}(2 \cdot \widetilde{X} \cdot sk_r, w)$ where $PS_{\mathcal{A}}$ is generated by $\mathcal{A}$ and sk_r is the secret key of pid_r. The associated value of this event is $KDF(PS, \widetilde{X}||pid_r)(= K_1)$.

Note that in outsider unforgeability, since **Corrupt**(pid_{s^*}) or **Corrupt**(pid_{r^*}) is disallowed, S can perfectly handle all **Corrput** queries in the security game.

Now we consider a query **ENO**(pid_s, pid_r, H, M) with the user who with identity information pid_r and public-key pk_r. Consider the different case of pid_s in the following.

I. When $pid_s \in$ **DISHONEST**: the output of **ENO**(pid_s, pid_r, H, M) is simply defined to be $\perp$; **II**. When $pid_s \in$ **HONEST** : (1) In the case of $pid_s \neq pid_A$: such an oracle query can be perfectly handled by the simulator S as the scheme does; (2) In the case of $pid_s = pid_A$: If $pid_r \in$ **HONEST** but $pid_r \neq pid_A$, let $pk_r = p_B = as_B + e_B$, where s_B, e_B is sampled by S himself. Then S works as the honest pid_A does, except $PS_A = \widetilde{X} \cdot s_B$. Otherwise ($pid_r \notin$ **HONEST** or $pid_r = pid_A$), S computes $X = ar + f$, $d = h(X, pid_A, pid_B)$ and $\widetilde{X} = p_A d + X$ as the honest sender does. Then if the adversary makes this **ENO** query such that causes an **Encorrectsk** event to occur, sets w to be the first element of the associated values of the event, and K_1 to be the second element of the associated values of the event, otherwise S sets $w \xleftarrow{\$} \{0,1\}^n$, and K_1 to be a string which is taken uniformly at random from $\mathcal{K}$ of AEAD. Finally S computes $C_{AE} = \mathcal{E}_{K_1}(H, pid_A||X||M)$, and returns $C = (H, \widetilde{X}, w, C_{AE})$ as the output of **ENO**(pid_s, pid_r, H, M).

Also the intermediate randomness generated by S is stored in $\mathcal{ST}_C$. Then S can perfectly handle all the **ENO**, **Corrupt** and **Exposure** queries.

Note that for a query **DEO**$(pid_r, C = (H, \widetilde{X}, w, C_{AE}))$ made by the adversary, if $pid_r \in$ **DISHONEST**, the oracle can output $\perp$ simply, and other case except that $pid_r \in$ **HONEST**, $pid_r = pid_A$, S can works as the scheme does. Hence we only consider the case of $pid_r \in$ **HONEST** and $pid_r = pid_A$.

Firstly, if C was ever output by **ENO**(pid_s, pid_A, H, M) query for some $M \in \{0,1\}^*$ and $pid_s \in$ **HONEST**, S outputs (pid_s, M) simply. Otherwise, if this **ENO** query causes a **Decorrecksk** event to occur, sets K_1 to the associated value of the event and uses K_1 to decrypt C_{AE}, finally returns the results to the adversary. Otherwise, the simulator returns $\perp$ which indicates C is an invalid ciphertext for pid_r. Denote by **failure** event that the simulator S outputs $\perp$ for (pid_A, C), while **DEO**(pid_A, C) does not. If **failure** event does not occur, then the simulation S for **DEO** is perfect. So we next to show that the probability of **failure** event occur is negligible.

If C was ever output by the **ENO** oracle, there are two cases: (1) When $C \leftarrow$ **ENO**(pid_i, pid_A, H, M) for arbitrary (pid_i, H, M), where $pid_i \in$ **HONEST**, S outputs (pid_s, M) simply, and **failure** event will never occur. (2) When $C \leftarrow$ **ENO**$(pid_i,\ pid_j, H, M)$ for arbitrary (pid_i, H, M) where $pid_j \neq pid_A$, since in **ENO** oracle the input for KDF is $\widetilde{X}||pid_j$ and in **DEO** oracle is $\widetilde{X}||pid_A$, by the security of AEAD defined in Sect.2.2, $\mathbf{DEO}(pid_A, C)$ outputs $\perp$ with overwhelming probability, and **failure** event will never occur.

So when **failure** occurs, with overwhelming probability it holds that C was not ever output by the **ENO** oracle and **Decorrecksk** event does not occur. But still $\mathbf{DEO}(pid_A, C = (H, \widetilde{X}, w, C_{AE}))$ query outputs a valid decryption value, then there must have that (H, C_{AE}) make up a valid ciphertext with respect to $K_1 = KDF(PS, \widetilde{X}||pid_A)$ for AEAD, where $PS = \mathrm{rec}(2 \cdot \widetilde{X} \cdot sk_A, w)$. If we can show that the last case occur with negligible probability, then **failure** occurs with negligible probability.

Consider two cases. (1) If $K_1 = KDF(PS, \widetilde{X}||pid_A)$ was set by S when dealing with a $\mathbf{ENO}(pid_A, pid_A,\ H', M')$ query. Recall that in this case K_1 is set without querying KDF query (but with checking whether **Encorrectsk** event occurs or not). This implies that by the security of KDF, with overwhelming probability, $(\widetilde{X}, w)$ is part of the output of $\mathbf{ENO}(pid_A, pid_A, H', M')$ query set by S. We denote by $(H', \widetilde{X}, w, C'_{AE})$ the output of S when dealing with $\mathbf{ENO}(pid_A, pid_A, H', M')$ query. Note that $C = (H, \widetilde{X}, w, C_{AE})$ was not ever output by the **ENO** oracle, so $(H', C'_{AE}) \neq (H, C_{AE})$. This means that (H', C'_{AE}) is a new valid AEAD ciphertext w.r.t. K_1, and by the AEAD security this occurs with negligible probability. (2) Otherwise, K_1 was neither defined for the KDF oracle nor ever set by S. By the AEAD security, this case occurs with negligible probability. Hence **failure** event occurs with negligible probability and S can handle all **DEO** queries in the security game.

For keeping the consistency, from now on whenever $\mathcal{A}^{OU}$ makes an KDF oracle query, if this query causes an **Encorrectsk** event or a **Decorrecksk** event to occur, outputs the associated value of that event (for **Encorrectsk** event, outputs the second one of the associated values), else outputs K_1 to be a string which is taken uniformly at random from $\mathcal{K}$ of AEAD.

Then from the view of $\mathcal{A}^{OU}$, the simulation S is computationally indistinguishable from that in the real attack game.

Recall that $(pid_{r^*}, C^* = (H^*, \widetilde{X}^*, w^*, C^*_{AE}))$ is a successful forgery by $\mathcal{A}^{OU}$. We say that $\mathcal{A}^{OU}$ must have made the RO query $d^* = h(X^*, pid_{s^*}, pid_{r^*})$ such that $\widetilde{X}^* = pk_{s^*} \cdot d^* + X^* = p_A \cdot d^* + X^*$, where X^* may be generated by $\mathcal{A}^{OU}$ himself. Otherwise, with overwhelming probability, $\mathbf{Decrypt}(sk_{r^*}, pid_{r^*}, C^*)$ returns $\perp$ in the random oracle model. Then similar with the analysis of **failure** event occurs in simulating the **DEO** query, $\mathcal{A}^{OU}$ must have cause an **Encorrectsk** event or a **Decorrectsk** event to occur, which means that $\mathcal{A}^{OU}$ must have made the RO query to get $K_1^* = KDF(PS, \widetilde{X}^*||pid_A)$, where $PS = \mathrm{rec}(2 \cdot PS_{\mathcal{A}}, w) = \mathrm{rec}(2 \cdot \widetilde{X}^* \cdot s_A, w^*)$ and $PS_{\mathcal{A}}$ is generated by $\mathcal{A}$ himself.

Then rewind the adversary to the point that it just made the RO query $h(X^*, pid_{s^*},\ pid_{r^*}) = h(X^*, pid_A, pid_A)$, and let this RO query return back a $d^{*'}$ which was different from d^* and was taken uniformly at random from h's range. By the general forking lemma [4], $\mathcal{A}^{OU}$ will also output a successful forgery $(pid_{r^*}, C^{*'} = (H^{*'}, \widetilde{X}^{*'}, w^{*'}, C^{*'}_{AE}))$ with non-negligible probability in the second run after rewinding. In this case $\widetilde{X}^{*'} = p_A \cdot d^{*'} + X^*$, and $\mathcal{A}^{OU}$ will make the query $KDF(PS', \widetilde{X}^{*'}||pid_A)$ with $PS' = \mathrm{rec}(2 \cdot PS'_{\mathcal{A}}, w^{*'}) = \mathrm{rec}(2 \cdot \widetilde{X}^{*'} \cdot s_A, w^{*'})$, where

$PS'_{\mathcal{A}}$ is generated by the adversary himself. This means that $\widetilde{X}^* \cdot s_A \approx PS_{\mathcal{A}}$; $\widetilde{X}^{*'} \cdot s_A \approx PS'_{\mathcal{A}}$. The simulator can set $\xi_1 = PS_{\mathcal{A}} \approx (p_A \cdot d^* + X^*) \cdot s_A$, $\xi_2 = PS'_{\mathcal{A}} \approx (p_A \cdot d^{*'} + X^*) \cdot s_A$, then compute $\xi_3 = (\xi_1 - \xi_2) \cdot (d^* - d^{*'})^{-1} \approx p_A s_A$. Finally, for a quadruple (a, p_A, p_A, W) where $p_A = as_A + e_A$ with $s_A, e_A \xleftarrow{\$} \chi_\alpha$ and $W \leftarrow \langle \overline{p_A s_A} \rangle_2$, S adds the value of $\mathrm{rec}(2 \cdot \xi_3, W)$ to the list of possible values for $\tau(p_A, s_A)$, which violates the vPWE assumption.

But to apply the forking lemma, it needs to ensure that the RO query $d^* = h(X^*, pid_A, pid_A)$ must be prior to the RO query $KDF(PS, \widetilde{X}^* || pid_A)$ where $PS = \mathrm{rec}(2 \cdot \widetilde{X}^* \cdot s_A, w)$ for some w in the successful forge. In the following we show that the probability that $\mathcal{A}^{OU}$ makes $KDF(PS, \widetilde{X}^* || pid_A)$ prior to $h(X^*, pid_A, pid_A)$ is negligible.

Suppose $(H, \widetilde{X}^*, w, C_{AE}) = \mathbf{Encrypt}(pid_i, pid_A, H, M)$ with $pid_i \in$ **DISHO−NEST** or $pid_i \in$ **HONEST** but corrupted. We denote by $pk_i = p_C = as_C + e_C$, $\widetilde{X}^* = p_C \cdot d_i + X_i$ with $d_i = h(X_i, pid_i, pid_A)$. This means that the target $\widetilde{X}^*$, which is appeared in the successful forgery, has already appeared in a former output of $\mathbf{Encrypt}(pid_i, pid_A, H, M)$ with some $pid_i \neq pid_A$ (that is $\widetilde{X}^* = p_C \cdot d_i + X_i = p_A \cdot d^* + X^*$). We refer to such an event as **collision**. Then when **collision** event occurs, prior to the query $d^* = h(X^*, pid_{s^*}, pid_{r^*})$, the $KDF(PS, \widetilde{X}^* || pid_A)$ oracle query was either made by $\mathcal{A}^{OU}$ itself or by the corrupted user pid_i. Because for any pair of $(pid_i, pid_j, X) \neq (pid_{i'}, pid_{j'}, X')$, there is

$$\Pr[pk_i \cdot h(X, pid_i, pid_j) + X = pk_{i'} \cdot h(X', pid_{i'}, pid_{j'}) + X'] \leq 2^{-l},$$

where l is the output length of random oracle h. Hence for a polynomial-time adversary $\mathcal{A}$, the probability of $\mathcal{A}$ causes a **collision** event to occur is negligible. Hence finishes the proof of outsider unforgeability. □

4.2 Security Proof of Insider Confidentiality

Theorem 6. *The scheme RLWE-ICAE in Figure 1 satisfies insider confidentiality in the random oracle model, under the AEAD security and the* vPWE *assumption.*

Proof. We proof the lemma by showing that if

$$\Pr[\mathbf{Encry\text{-}Confident}_{\mathcal{A}^{IC},\text{RLWE-ICAE}}(\kappa) = 1] > negl(\kappa),$$

vPWE problem defined in Sect. 2.7 can be solved with non-negligible probability.

We first assume that the challenger $\mathcal{C}$ has correctly guessed the target receiver pid_{r^*} (for this happens with probability $\frac{1}{N}$). Assume $p_B = as_B + e_B$, $X^* = ar^* + f^*$ and $W = \langle \overline{X^* s_B} \rangle_2$ ($\overline{X^* s_B} \leftarrow$dbl$(X^* s_B)$, defined in Sect.2.6), where $s_B, e_B \xleftarrow{\$} \chi_\alpha$ and $r^*, f^* \xleftarrow{\$} \chi_\beta$. The goal of $\mathcal{C}$ is to compute $\tau(X^*, s_B)$ given (a, X^*, p_B, W).

Firstly, $\mathcal{C}$ sets the target receiver's public-key to be $pk_{r^*} = p_B$, and sets all the rest users' public/secret key pairs by itself. Then $\mathcal{C}$ can perfectly handle the oracle queries made by the adversary $\mathcal{A}^{IC}$ except against user $pid_{r^*} = pid_B$. In the following, we make $\mathcal{C}$ simulates the target receiver pid_B similar to the proof of outsider unforgeability.

Recall the encrypt experiment $\textbf{Encry-Confident}_{\mathcal{A}^{IC},\text{RLWE-ICAE}}(\kappa)$, the adversary outputs the associated data H, two equal length messages (M_0, M_1), two pairs of public identity information $(pid_{s_0^*}, pid_{r^*})$ and $(pid_{s_1^*}, pid_{r^*})$ with $pid_{s_0^*}, pid_{s_1^*}, pid_{r^*} \in \textbf{HONEST}$ and we assume $pid_{r^*} = pid_B$. $\mathcal{C}$ chooses a random bit $\gamma \overset{\$}{\leftarrow} \{0,1\}$ and sets the target ciphertext C^* as follows.

For presentation simplicity, we denote by $pk_{s_\gamma^*} = p_A$ the public key of the user s_γ^*, (there is possible that $pid_{s_\gamma^*} = pid_{r^*}$ and thus $p_A = p_B$). $\mathcal{C}$ computes $d^* = h(X^*, pid_{s_\gamma^*}, pid_{r^*}) = h(X^*, pid_A, pid_B)$ with X^*, and then computes $\widetilde{X}^* = p_A \cdot d^* + X^*$. Then if this **ENO** query causes an **Encorrectsk** (defined in the proof of outsider unforgeability) event to occur, set w^* to be the first element of the associated values of the event, and K_1 to be the second element of the associated values of the event, otherwise S sets $w^* \overset{\$}{\leftarrow} \{0,1\}^n$, and K_1 to be a string which is taken uniformly at random from $\mathcal{K}$ of AEAD. Finally $\mathcal{C}$ computes $C_{AE}^* = \mathcal{E}_{K_1}(H, pid_{s_\gamma^*}||X^*||M_\gamma)$ to return. Then whenever the adversary makes a KDF query, if this query cause an **Encorrectsk** event or a **Decorrecksk** event to occur, outputs the associated value of that event (for **Encorrectsk** event, outputs the second one of the associated values), else outputs K_1 to be a string which is taken uniformly at random from $\mathcal{K}$ of AEAD.

Note that $\widetilde{X}^* = p_A \cdot d^* + X^* = a(s_A d^* + r^*) + (e_A d^* + f^*)$, and d^* is returned by RO oracle. By Theorem 1 of rejection sampling, the finally distribution of $s_A d^* + r^*$ and $e_A d^* + f^*$ in the outputs will be indistinguishable with r^* and f^* for the adversary. As long as the adversary can't solve the decision RLWE problem, $\widetilde{X}^*$ is indistinguishable with random chosen elements from R_q. In this way, $\widetilde{X}^*$ can perfectly hides the sender's identity information even if the adversary corrupt $pid_{s_0^*}$ or $pid_{s_1^*}$. Then to win such an insider confidentiality game in the RO model, by the AEAD security, the adversary must have caused **Encorrectsk** event to occur, (which is similar to the proof of outsider unforgeability) which means that $\mathcal{A}^{IC}$ must have made the RO query to get $K_1^* = KDF(PS, \widetilde{X}^*||pid_B)$, where $PS = \text{rec}(2 \cdot PS_{\mathcal{A}}, w^*) = \text{rec}(2 \cdot \widetilde{X}^* \cdot s_B, w^*)$ and $PS_{\mathcal{A}}$ is generated by $\mathcal{A}$ himself.

Similar to the proof of outsider unforgeability, $\mathcal{C}$ rewinds the adversary to the point that it just makes the oracle query $h(X^*, pid_A, pid_B)$, and redefines a new output $d^{*'} = h(X^*, pid_A, pid_B)$ with $d^{*'}$ is different from d^* and is taken uniformly at random from h's range. Then re-runs the adversary from this rewinding point. By the forking lemma, the adversary will make the query make the query $KDF(PS', \widetilde{X}^{*'}||pid_B)$ with $PS' = \text{rec}(2 \cdot PS'_{\mathcal{A}}, w^{*'}) = \text{rec}(2 \cdot \widetilde{X}^{*'} \cdot s_B, w^{*'})$, where $PS'_{\mathcal{A}}$ is chosen by the adversary himself.

Simulator can get $\xi_1 = PS_{\mathcal{A}} \approx (p_A \cdot d^* + X^*) \cdot s_B$, $\xi_2 = PS'_{\mathcal{A}} \approx (p_A \cdot d^{*'} + X^*) \cdot s_B$, then the simulator can compute $\xi_3 = \xi_1 - (\xi_1 - \xi_2) \cdot (d^* - d^{*'})^{-1} \cdot d^* \approx X^* s_B$. Finally, S adds the value of $\text{rec}(2 \cdot \xi_3, W)$ to the list of possible values for $\tau(X^*, s_B)$, which violates the vPWE assumption. □

5 Concrete Parameters

In this section, we present the choices of parameters and give the complexity assessment of RLWE-ICAE.

We use the property for product of two Gaussian distributed random values which are stated in [29]. Let $x, y \in R$ be two polynomials with degree of n. Assume that the coefficients of x and y are distributed according to a discrete Gaussian distribution with parameter β_x, β_y, respectively. Then we have that the individual coefficients of the polynomial xy are approximately normally distributed around zero with parameter $\beta_x \beta_y \sqrt{n}$. Hence for $||PS_A - PS_B||_\infty = ||e_B s_A d + r e_B + g - f s_B - e_A s_B d||_\infty < \frac{q}{8}$, applying Lemma 1 we have that $||k_A - k_B||_\infty > 13 \cdot \sqrt{2n\alpha^2\beta^2 + \beta^2 + 2n^2\alpha^6}$ with probability approximating 2^{-121}. We set $13 \cdot \sqrt{2n\alpha^2\beta^2 + \beta^2 + 2n^2\alpha^6} < \frac{q}{8}$ to make sure the correctness of the scheme. Note that since the Theorem 1 of rejection sampling, the distributions of $r + s_A d$ is according to χ_β. We follow a way of parameter choosing in [29]. To choose an appropriate β, we set $\eta = 1/2$ in Lemma 2 such that $||s_A d|| \leq 1/2n\alpha^2$ with probability at most $2 \cdot 0.943^{-n}$. In order to make the rejection sampling work, we need to set $\beta \geq \zeta \cdot 1/2n\alpha^2$ for some constant ζ. When we set $\zeta = 12$, by Theorem 1, there is an expect number of rejection sampling about $M = 2.72$ and a statistical distance about $\frac{2^{-100}}{M}$.

For the security of our parameters, Alkim et al. [2] analysised RLWE and LWE using two BKZ types attacks: prime attack and dual attack [7]. The thoughts of their approach is to replace the enumeration core-SVP algorithm in BKZ by sieve algorithm, and only evaluate the cost of one call to an SVP oracle in dimension b. For more detail, we refer to [2]. We use their techniques to assess the core-SVP security. But to estimate the security of our scheme more accurately, we follow Albrecht's estimation [1] about the number for the calls to core-SVP oracle. Albrecht estimated it to be $8d$, where d is the dimension of the embedding lattice. We will first compute the core-SVP security, then multiple it with $8d$ to obtain the final security.

Two recommend parameters choices is given in Table 2. Remark that q must be a prime and satisfies $q = 1 \bmod 2n$. In the table, we denote classical security as the best-known classical attack time complexity, and quantum security as the best-known quantum attack time complexity [2].

Table 2. Recommend Parameters for RLWE-ICAE

		I	**II**
n	power of 2	1024	2048
α		2.828	2.828
β	$> \frac{1}{2}n\alpha^2\zeta = \frac{1}{2}n\alpha^2 \cdot 12$	49152	98304
$log_2\beta$		$\approx$15.6	$\approx$16.6
q	$> 104 \cdot \sqrt{2n\alpha^2\beta^2 + \beta^2 + 2n^2\alpha^6}$	231362561	654340097
$log_2 q$		$\approx$27.8	$\approx$29.3
classical security		120 bits	256 bits
quantum security		110 bits	234 bits

6 Conclusion

We proposed a lattice based identity-concealed authenticated encryption scheme: RLWE-ICAE. The scheme enjoys many nice properties of higncryption such as 0-RTT option, forward ID-privacy, receiver deniability and x-security. Meanwhile since our scheme is based on RLWE, it also enjoys the properties of lattice-based cryptography, such as conceptual simplicity, worst-case hardness assumption, and resistance to quantum computer attacks. Our scheme benefits from Peikert's reconciliation mechanism [23] technique which can help two parties compute a same element from two approximate values. We use the rejection sampling technique to hide the static secret information. To prove the security of our scheme, we introduce vPWE assumption, which is a variant of Pairing with Errors assumption [10] by replacing the reconciliation mechanism in [10] with Peikert's version [23]. For further works, we will consider to construct an identity concealed key exchange from ideal lattices.

Acknowledgments. This article is supported by The National Key Research and Development Program of China (Grant No. 2017YFA0303903), National Cryptography Development Fund (No. MMJJ20170121), and Zhejiang Province Key R&D Project(No. 2017C01062). Authors thank Aijun Ge for discussions and the anonymous ProvSec'19 reviewers for helpful comments.

References

1. Albrecht, M.R.: On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In: Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II. pp. 103–129 (2017)
2. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - A new hope. In: 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016. pp. 327–343 (2016), `https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim`

3. Baek, J., Steinfeld, R., Zheng, Y.: Formal proofs for the security of signcryption. J. Cryptology **20**(2), 203–235 (2007)
4. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006. pp. 390–399 (2006)
5. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015. pp. 553–570 (2015)
6. Brzuska, C., Smart, N.P., Warinschi, B., Watson, G.J.: An analysis of the EMV channel establishment protocol. In: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013. pp. 373–386 (2013)
7. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. pp. 1–20 (2011)
8. Dent, A.W.: Hybrid cryptography. IACR Cryptology ePrint Archive **2004**, 210 (2004), `http://eprint.iacr.org/2004/210`
9. Ding, J.: A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptology ePrint Archive **2012**, 688 (2012), `http://eprint.iacr.org/2012/688`
10. Ding, J., Alsayigh, S., Lancrenon, J., RV, S., Snook, M.: Provably secure password authenticated key exchange based on RLWE for the post-quantum world. In: Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings. pp. 183–204 (2017)
11. Ducas, L., Durmus, A.: Ring-lwe in polynomial rings. In: Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings. pp. 34–51 (2012)
12. Gorantla, M.C., Boyd, C., Nieto, J.M.G.: On the connection between signcryption and one-pass key establishment. In: Cryptography and Coding, 11th IMA International Conference, Cirencester, UK, December 18-20, 2007, Proceedings. pp. 277–301 (2007)
13. Halevi, S., Krawczyk, H.: One-pass HMQV and asymmetric key-wrapping. In: Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings. pp. 317–334 (2011)
14. Iyengar, S., Nekritz, K.: Building zero protocol for fast, secure mobile connections (2017). `https://code.fb.com/android/building-zero-protocol-for-fast-secure-mobile-connections/`
15. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is ssl?). In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. pp. 310–331 (2001)
16. Langley, A., Chang, W.T.: Quic crypto (2014). `https://docs.google.com/document/d/1g5nIXAIkN_Y-7XJW5K45IblHd_L2f5LTaDUDwvZ5L6g`
17. Lyubashevsky, V.: Lattice signatures without trapdoors. IACR Cryptology ePrint Archive **2011**, 537 (2011), `http://eprint.iacr.org/2011/537`

18. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. pp. 738–755 (2012)
19. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. pp. 1–23 (2010)
20. MacKenzie, P.: The pak suite: Protocols for password-authenticated key exchange. In: DIMACS Technical Report 2002-46 (2002). p. 7 (2002)
21. Menezes, A., Qu, M., Vanstone, S.A.: Some new key agreement protocols providing mutual implicit authentication (1995)
22. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007)
23. Peikert, C.: Lattice cryptography for the internet. In: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. pp. 197–219 (2014)
24. Rescorla, E.: The transport layer security (TLS) protocol version 1.3. RFC **8446**, 1–160 (2018)
25. Rogaway, P.: Authenticated-encryption with associated-data. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002. pp. 98–107 (2002)
26. Roskind, J.: Quick udp internet connections: Multiplexed stream transport over udp. 2012
27. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
28. Yang, Z., Chen, Y., Luo, S.: Two-message key exchange with strong security from ideal lattices. In: Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings. pp. 98–115 (2018)
29. Zhang, J., Zhang, Z., Ding, J., Snook, M., Dagdelen, Ö.: Authenticated key exchange from ideal lattices. In: Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II. pp. 719–751 (2015)
30. Zhao, Y.: Identity-concealed authenticated encryption and key exchange. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 1464–1479 (2016)
31. Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In: Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings. pp. 165–179 (1997)