Key Exchange and Authenticated Key Exchange with Reusable Keys Based on RLWE Assumption

Jintai Ding[1], Pedro Branco*[2], and Kevin Schmitt[1]

[1]University of Cincinnati
[2]SQIG-IT / IST-Universidade de Lisboa

Abstract

Key Exchange (KE) is, undoubtedly, one of the most used cryptographic primitives in practice. Its authenticated version, Authenticated Key Exchange (AKE), avoids man-in-the-middle-based attacks by providing authentication for both parties involved. It is widely used on the Internet, in protocols such as TLS or SSH. In this work, we provide new constructions for KE and AKE based on ideal lattices in the Random Oracle Model (ROM). The contributions of this work can be summarized as follows:

- It is well-known that RLWE-based KE protocols are not robust for key reuses since the signal function leaks information about the secret key. We modify the design of previous RLWE-based KE schemes to allow key reuse in the ROM. Our construction makes use of a new technique called *pasteurization* which enforces a supposedly RLWE sample sent by the other party to be indeed indistinguishable from a uniform sample and, therefore, ensures no information leakage in the whole KE process.
- We build a new AKE scheme based on the construction above. The scheme provides implicit authentication (that is, it does not require the use of any other authentication mechanism, like a signature scheme) and it is proven secure in the Bellare-Rogaway model with weak Perfect Forward Secrecy in the ROM. It improves previous designs for AKE schemes based on lattices in several aspects. Our construction just requires sampling from only one discrete Gaussian distribution and avoids rejection sampling and noise flooding techniques, unlike previous proposals (Zhang *et al.*, EUROCRYPT 2015). Thus, the scheme is much more efficient than previous constructions in terms of computational and communication complexity.

Since our constructions are provably secure assuming the hardness of the RLWE problem, they are considered to be robust against quantum adversaries and, thus, suitable for post-quantum applications.

*Email: pmbranco@math.tecnico.ulisboa.pt

1 Introduction

Key Exchange (KE) is a cryptographic primitive that allows two parties to agree on a shared key while a third party eavesdropping the communication gets no information about the shared key. Subsequently, the shared key can be used to securely communicate or provide authentication. This cryptographic primitive was presented in the seminal paper by Diffie and Hellman [22], which marked the birth of modern cryptography, and is, undoubtedly, one of the most used primitives in both theoretical and real applications.

However, the standard KE primitive is not robust to man-in-the-middle (or impersonation) attacks: an adversary controlling the network can easily modify the communication between two parties, making them believe that they are privately communicating with each other while, in fact, the conversation is being controlled by the adversary. Authenticated Key Exchange (AKE) is a flavor of KE which provides authentication to the parties involved and, thus, it avoids man-in-the-middle attacks. Due to its robustness to this type of attacks, AKE is used in a wide range of applications such as SSL [32] and TLS [21].

Authentication for KE can be achieved explicitly (that is, by explicitly using other primitives that provide authentication, like signature schemes) or implicitly (that is, without requiring the explicit use of other primitives). The idea of using implicit authentication for KE was presented in [48] and has been intensively studied since then. In these protocols, a key is shared by means of static and ephemeral keys belonging to the parties involved in the protocol; authentication is guaranteed by the static key while the ephemeral key usually provides Perfect Forward Secrecy (PFS).[1] The efficiency in terms of both communication and computational complexity achieved by this kind of protocols, as well as its simplicity and elegance, has led to their massive standardization by institutions all over the world. Examples of such schemes are the MQV-like protocols (that is, MQV [49, 43], HMQV [41] and OAKE [55]), which have been extensively used as standard [38], and NAXOS [42].

All the above schemes are based on the Diffie-Hellman protocol, hence, their security is based on the discrete logarithm assumption. With the possible advent of a large-scale quantum computer, these protocols will become obsolete [54]. This fact has led the National Institute of Standards and Technology (NIST) to announce a call to define the next post-quantum standard protocols for KE and digital signatures to be used by national institutions in the USA [1]. Hence, the development of post-quantum KE schemes and its variants (such as AKE schemes) is of high priority and we should expect these types of protocols to be used in the near future.

1.1 Lattice-based Key Exchange

One of the first post-quantum KE was presented in [28], where a KE protocol based on the Ring Learning with Errors (RLWE) assumption [46] was pre-

[1] Recall that PFS guarantees that in case a static secret key is revealed to an adversary, previously established session keys are not compromised.

sented.[2] The scheme of [28] is also our starting point. hence, to ease the presentation of our results, we recall the scheme of [28] which we refer to as Ding's KE. Let $R_q = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$, for some prime q, and χ_α be a discrete Gaussian distribution. Let P_i and P_j be two parties that want to exchange a key. User P_i (resp. P_j) has a secret key s_i (resp. s_j) sampled from χ_α and the public key is a RLWE sample of the form $as_i + 2e_i$ (resp. $as_j + 2e_j$) where e_i (resp. e_j) is an error vector sampled from χ_α. Party P_i starts by sending $x_i = as_i + 2e_i$. Party P_j computes $y_j = as_j + 2e_j$ and $k_j = x_i s_j + 2g_j$ where g_j is sampled from χ_α. Now, party P_j computes the signal w_j of k_j, using a function Sig that just tells if each coefficient of k_j is within an interval or not. Party P_j sends y_j and the signal w_j of k_j to P_i. Both parties can now agree on a shared key using an extractor function Mod_2.

Recent results have noticed that information is leaked by the signal function, in RLWE-based KE protocols [31, 23, 25]. In particular, if the value x_i sent by party P_i is not computed honestly (that is, if it is not an RLWE sample), then party P_i can recover information about party P_j's secret key. This can be done by noticing the behavior of the signal sent by P_j after several executions of the scheme. Hence, one cannot reuse the same keys in several executions of the protocol, risking itself of having its secret key exposed to someone else.

After the introduction of Ding's KE [28], several other lattice-based KE [51, 56, 14, 5] were proposed. However, most of these schemes do not provide authentication by themselves. So, authentication is guaranteed by means of an explicit mechanism (such as signature schemes). As far as we are aware, the only lattice-based AKE that provides implicit authentication is the scheme of [56]. Unfortunately, the use of techniques such as rejection sampling and noise flooding may raise implementation issues [30]. They also turn the parameters of the scheme large and it is required the use of more than one discrete Gaussian distribution.

Another different approach to exchange keys is to use Key Encapsulation Mechanism (KEM), for which several lattice-based proposals have been made in recent years [15, 33, 34, 13]. However, KEMs require the use of a decryption algorithm, which is usually more computationally expensive than using a KE protocol. Also, KEMs usually do not provide PFS, that is, all the previously established session secret keys are compromised in case the secret key of a user is exposed. Recall that, in a nutshell, a KEM is a Public-Key Encryption (PKE) used as a KE; hence, once the secret key of a party is revealed, every message that was encrypted using the corresponding public key is also revealed (in the case of a KEM, this corresponds to established session keys).

1.2 Contributions and techniques

In this work, we present post-quantum solutions for KE and AKE that allow for key reuse. We base the security of our schemes in the RLWE assumption [46],

[2] Another prominent line of research in post-quantum Key Exchange adopts the supersingular isogeny-based approach [39].

a well-established assumption in cryptography that enjoys an average-case reduction from worst-case lattice problems. Schemes based on this assumption usually provide post-quantum security and are asymptotically more efficient than their discrete log-based counterpart.

1.2.1 Key Exchange with reusable keys

First, we remark that if P_i's message is an RLWE sample, then the value k_j computed by P_j is indistinguishable from a uniformly chosen value and, thus, the signal of k_j is also indistinguishable from a uniformly chosen value. Hence, we just have to force each party to behave honestly in the protocol.

We use a technique, which we call *pasteurization*, to force the parties involved in the KE scheme to behave honestly. The technique was previously introduced in [26] in the context of zero-knowledge proofs. The idea of this technique is the following: after receiving x_i from P_i, the party P_j *pasteurizes* x_i, i.e., it computes

$$\bar{x}_i = x_i + a\mathsf{H}(x_i) + 2f_j,$$

where H is a random oracle whose outputs are sampled from χ_α and f_j is sampled from χ_α. If x_i is indeed an RLWE sample, then the pasteurization $\bar{x}_i$ is also an RLWE sample, for which P_i knows the secret. However, when x_i is not an RLWE sample, then $\bar{x}_i$ looks pseudorandom to P_i. Thus, the signal of k_j is also pseudorandom and P_i cannot extract information about P_j's secret key from it. We conclude that party P_i gains nothing by not following the protocol. To guarantee that party P_i can also reuse its key in several executions of the protocol, we make it pasteurize y_j sent from P_j. A scheme of the protocol is presented in Figure 1.

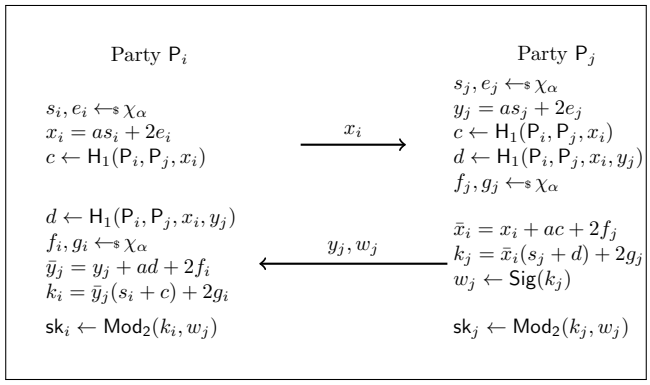

Figure 1: Ding's Ke with reusable keys: χ_α is a discrete Gaussian distribution over R_q with standard deviation α, $\mathsf{H}_1 : \{0,1\}^* \to \chi_\alpha$ is a hash function whose outputs are sampled from χ_α and Sig and Mod_2 are the signal and the extraction functions (respectively), as defined in [28].

In the Diffie-Hellman KE [22], exchanged messages from both parties are supposed to be in a group $\mathbb{G}$. We avoid possible attacks by making the parties verify if all the exchanged values are in $\mathbb{G}$, which can be done in polynomial time. However, in the RLWE-based KE, since the exchange messages are RLWE samples, it is impossible to straightforwardly check if they are honestly computed. The pasteurization technique can be seen as the analog of checking if the exchanged messages are in $\mathbb{G}$, in the Diffie-Hellman KE, since the technique also enforces good behavior by the parties involved.

1.2.2 New Authenticated Key Exchange scheme

Our major contribution is the design of a new AKE scheme based on the RLWE assumption. At the heart of our construction is the RLWE-based KE described above. The scheme can be found in Figure 2.

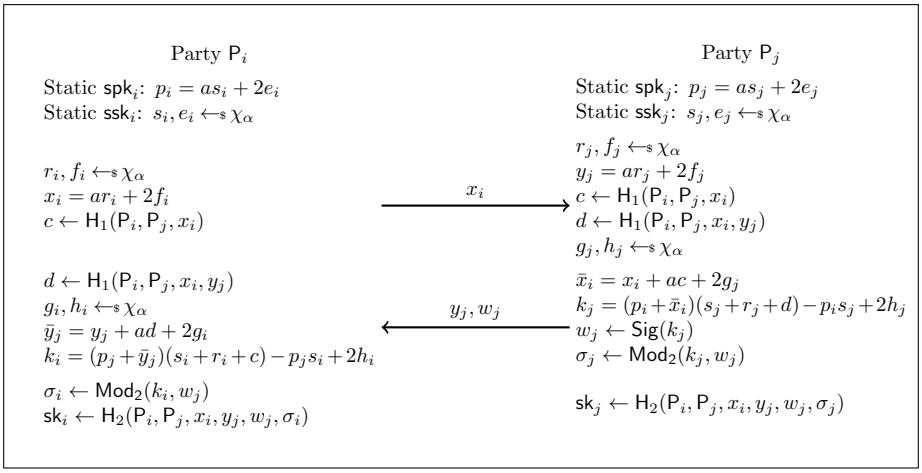

Figure 2: The new AKE protocol: χ_α is a discrete Gaussian distribution over R_q with standard deviation α, $\mathsf{H}_1 : \{0,1\}^* \to \chi_\alpha$ is a hash function whose outputs are sampled from χ_α, $\mathsf{H}_2 : \{0,1\}^* \to \{0,1\}^\kappa$ is a κ-bit key derivation function and Sig and Mod_2 are the signal and the extraction functions (respectively), as defined in [28].

Protocol idea. Let $(\mathsf{spk}_i, \mathsf{ssk}_i)$ (resp. $(\mathsf{spk}_j, \mathsf{ssk}_j)$) and $(\mathsf{epk}_i, \mathsf{esk}_i)$ (resp. $(\mathsf{epk}_j, \mathsf{esk}_j)$) be pairs of static and ephemeral public and secret keys of party P_i (resp. P_j). Symbolically, the key k_i that party P_i computes can be viewed as the sum of the shared keys between $(\mathsf{ssk}_i, \mathsf{epk}_j)$ (static secret key of P_i with ephemeral public key of P_j), $(\mathsf{esk}_i, \mathsf{spk}_j)$ (ephemeral secret key of P_i with static public key of P_j) and $(\mathsf{esk}_i, \mathsf{epk}_j)$ (ephemeral secret key of P_i with ephemeral public key of P_j). Similarly, P_j computes the sum of the shared keys between $(\mathsf{ssk}_j, \mathsf{epk}_i)$, $(\mathsf{esk}_j, \mathsf{spk}_i)$ and $(\mathsf{esk}_j, \mathsf{epk}_i)$.

More precisely, let $\mathsf{spk}_i = p_i = as_i + 2e_i$ and $\mathsf{epk} = x_i = ar_i + 2f_i$ (resp. $\mathsf{spk}_j = p_j = as_j + 2e_j$ and $\mathsf{epk} = y_j = ar_j + 2f_j$) be the static and ephemeral public keys of P_i (resp. P_j). The key k_i computed by P_i is equal to $(p_j + \bar{y}_j)(s_i + r_i + c) - p_j s_i$, which is the sum of all possible combinations between static and ephemeral keys of both parties, minus the key resulting of the exchange between static keys. As in the previous construction, we pasteurize y_j to avoid any leakage of information. Similarly, the key k_j computed by P_j is equal to $(p_i + \bar{x}_i)(s_j + r_j + d) - p_i s_j$. Of course k_i and k_j are just approximately equal, hence the functions Sig and Mod_2 are used to agree on a shared value.

Similarities with NAXOS protocol. We recall the NAXOS AKE protocol of [42] which is based on the Diffie-Hellman protocol. Let $\mathbb{G}$ be a group and let $(\mathsf{spk}_i, \mathsf{ssk}_i)$ (resp. $(\mathsf{spk}_j, \mathsf{ssk}_j)$) and $(\mathsf{epk}_i, \mathsf{esk}_i)$ (resp. $(\mathsf{epk}_j, \mathsf{esk}_j)$) be pairs of static and ephemeral public and secret keys of party P_i (resp. P_j). Party P_i computes the shared key as $\mathsf{H}(\mathsf{epk}_j^{\mathsf{ssk}_i}, \mathsf{spk}_j^{\mathsf{esk}_i}, \mathsf{epk}_j^{\mathsf{esk}_i})$ where H is a random oracle.

Our AKE protocol shares similarities with the NAXOS protocol. However, we compute the shared key in one operation while NAXOS computes three keys individually. This allows saving a couple of multiplications in the ring R_q, improving the efficiency.

However, we were not able to prove security in the (extended) Canetti-Krawczyk (eCK) model [42], as in NAXOS. This is due to the fact that, given two session transcripts $(x_i, (y_j, w_j))$ and $(x_i, (y_j, w'_j))$, these two sessions have the same state (that is, k_j) and the eCK model allows the adversary to get the state of parties in a session.[3]

Comparison with scheme of Zhang *et al.* As far as we are aware, the only RLWE-based AKE scheme with implicit authentication was presented in [56] (we refer to it as ZZD+ scheme, for convenience). We compare our scheme with this one.

In terms of computational complexity, the ZZD+ scheme requires ten multiplications in the ring R_q, five for each party. This is due to the use of the rejection sampling technique [44, 45], in which each party has to check if the ephemeral key leaks information.[4] Although for a proper choice of parameters, the rejection happens rarely, the test has to be done in every execution. The scheme also requires to sample six times from a discrete Gaussian distribution (three for each user), half the number of samples compared with our protocol. However, the ZZD+ scheme requires three distributions χ_α and χ_β with $\beta >> \alpha$ and χ_τ , since the *noise flooding* technique is used. [5] This turns the

[3]The scheme of [56] was not analyzed in the Canetti-Krawczyk (CK) model for the same reason.

[4]Rejection sampling is needed in the ZZD+ scheme since the scheme (implicitly) resorts to signatures to provide authentication, just like HMQV which (implicitly) relies on Schnorr signatures [53]. Our design does not require signatures as authentication is provided by the shared keys between ephemeral secret keys and static public keys.

[5]The noise flooding technique is used in ZZD+ in order for the ephemeral secret key to

implementation of the scheme way more complicated than the implementation of our scheme, which only needs the distribution χ_α.

	Rounds	Multiplications in R_q	Samples	Required distributions	Rejection Sampling
[56]	2	10	6	χ_α, χ_β ($\beta >> \alpha$), χ_τ	✓
Ours	2	6(+2 offline)	8	χ_α	✗

Table 1: Comparison with other AKE schemes with implicit authentication.

The ZZD+ scheme requires two elements of R_q and the signal to be sent during the execution of the protocol. Hence, it achieves the same communication complexity as our scheme. However, the use of several discrete Gaussian distributions (in particular, the use of χ_β with $\beta >> \alpha$) implies the use of a larger q than in our scheme for the same security level. This fact leads to much larger parameters to be used, comparing to our proposal, for the same security parameter We elaborate more on this in Section 6.

1.3 Other previous work

Prominent work on security models for AKE schemes was presented in [11, 19, 42]. Here, we work on the BR-model, which is the most common model used and it is believed to be enough for practical purposes since it also provides composability [18].

The idea of sanitizing the other's party message in the context of KE with lattices was already employed in [35]. However, the strategy used in [35] consists in multiplying the key obtained by P_j by a small value and then revealing it to P_i, so that P_i can also compute the key. Although the authors of [35] give arguments on why their construction is robust to the key reuse attacks of [31, 23], no proofs of security are presented. Contrarily to the construction of [35], we can prove robustness for key reuse for our scheme.

Identity-based Encryption (IBE) schemes can also be used as KEM that provide authentication and several constructions for IBE based on lattices have been proposed before [36, 2, 20, 29]. However, the efficiency of these schemes is too cumbersome to be used in practice.

Password-authenticated key-exchange is yet another flavor of AKE. Previous work on RLWE-based PAKE were presented in [40] (via public-key encryption scheme) and [24] (via KE).

2 Preliminaries

Let $\mathcal{D}$ be an algorithm. By $y \leftarrow \mathcal{D}(x)$ we denote the output y after running $\mathcal{D}$ on input x. If S is a set and ρ a distribution over S we denote by $x \leftarrow_{\$} S$ the

obliterate every information about the static secret key and, thus, to allow for key reuse. Our approach for the key reuse problem is to pasteurize RLWE samples, as explained previously.

element x sampled uniformly at random from S (if S is finite) and by $x \leftarrow_{\$} \rho$ the element x sampled from S according to ρ.

Let n be a power of 2. For a prime q, let $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Notice that R_q can be embedded into $\mathbb{R}^n$. In this work, we consider the coefficient embedding where each polynomial $a(X) = a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$ is mapped to the vector $(a_0, \ldots, a_{n-1}) \in \mathbb{R}^n$. For $a \in R_q$, $\|a\|$ denotes the usual ℓ_2 norm of the embedded vector $(a_0, \ldots, a_{n-1}) \in \mathbb{R}^n$ and $\|a\|_\infty$ denotes its ℓ_∞ norm.

We define the statistical distance between two random variables X and Y by

$$d(X, Y) = \frac{1}{2} \sum_r \left| \Pr[X = r] - \Pr[Y = r] \right|.$$

Let $\mathcal{X} = \{X_\kappa\}$ and $\mathcal{Y} = \{Y_\kappa\}$ two probability distributions. We say that $\mathcal{X}$ and $\mathcal{Y}$ are statistically close to uniform if $d(X_\kappa, Y_\kappa) \leq \mathsf{negl}(\kappa)$.

2.1 BR security model

We describe the Bellare-Rogaway (BR) model [11] adapted to two-pass AKE schemes, which was also used in [56]. In this model, the adversary has full control of the network, which means that it can read, modify, intercept, and inject messages in the network. It is also allowed to reveal session keys that have been established, to modeled possible leaks of information in the real-world use of a protocol, and to reveal the static secret keys of users, in order to capture PFS. We briefly survey the security model.

In this work, an execution of an AKE scheme Π is performed by two parties, the initiator I and the responder R. Let N be the number of users using the AKE protocol. Each user has a pair of static public and secret keys. As usual, we assume that static public keys of the users are validated either by a Certificate Authority (CA) or using some other mechanism.

Session. A session $\mathsf{sid} = (\Pi, I, \mathsf{P}_i, \mathsf{P}_j, X_i, Y_j)$ (or $\mathsf{sid} = (\Pi, R, \mathsf{P}_j, \mathsf{P}_i, X_i, Y_j)$) is a single execution of Π, where I (or R) denotes the role of the session owner, P_i and P_j are the parties involved in the session (the first one being the owner of the session), X_i is the message sent from P_i to P_j and Y_j is the message sent from P_j to P_i. A session has a owner which is the party that activates it. A session is said to be completed when a party computes a session key. A session $\mathsf{sid} = (\Pi, I, \mathsf{P}_i, \mathsf{P}_j, X_i, Y_j)$ has a matching session if $\widetilde{\mathsf{sid}} = (\Pi, R, \mathsf{P}_j, \mathsf{P}_i, X_i, Y_j)$ exists (and vice-versa).

A session can be activated by a message of the form $(\Pi, I, \mathsf{P}_i, \mathsf{P}_j)$ (when the session belongs to the initiator) or of the form $(\Pi, R, \mathsf{P}_j, \mathsf{P}_i, X_i)$ (when the session belongs to the responder). In the first case, we say that P_i is the initiator and it should output a message X_i. After receiving a message of the form $(\Pi, R, \mathsf{P}_j, \mathsf{P}_i, X_i)$, P_j takes the role of the responder and should output a message Y_j and computes the shared session key. Finally, upon receiving a message of the form $(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, X_i, Y_j)$, P_i computes the shared session key, which will be the same as the one computed by P_j.

Oracles. The adversary $\mathcal{A}$ has access to the following oracles:

- $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j)$: party P_i is activated as the initiatior. Initiate returns X_i, a message intended for party P_j.
- $\mathsf{Respond}(\Pi, R, \mathsf{P}_j, \mathsf{P}_i, X_i)$: party P_j is activated as the responder. Respond returns Y_j, a message intended for P_i.
- $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, X_i, Y_j)$: the message Y_j is sent to P_i to complete a session previously activated by an Initiate query, which outputted X_i.
- $\mathsf{skReveal}(\mathsf{sid})$: it returns the session key of session sid, if it exists.
- $\mathsf{Corrupt}(\mathsf{P}_i)$: it returns the static secret key of party P_i.
- $\mathsf{Test}(\mathsf{sid})$: it chooses $b \leftarrow_{\$} \{0,1\}$. If $b = 0$, it returns a uniformly chosen key. Else, it returns the session key of session sid.

A party that has its key revealed (by querying Corrupt) is called dishonest.

We just allow Test to be called once and on a fresh session to avoid trivial attacks. The definition of fresh session is presented below.

Security of AKE. First, we define the concept of a fresh session.

Definition 1 (Fresh session)**.** *Let* sid *be a completed session and let* $\widetilde{\mathsf{sid}}$ *be the matching session (if it exists). We say that* sid *is fresh if:*

1. skReveal *was not queried on* sid *nor on* $\widetilde{\mathsf{sid}}$*;*
2. Corrupt *was not queried on* P_i *nor on* P_j*, if* $\widetilde{\mathsf{sid}}$ *does not exist.*

Weak Perfect Forward Secrecy (wPFS) means that it is infeasible for an adversary to recover a session key that was established without its intervention [41]. This should hold even when the attacker knows the static secret keys of both parties involved in the key exchange. Restricting the adversary to query the Test oracle on a fresh session captures the notion of wPFS. Recall that wPFS is the strongest type of PFS that a two-pass KE protocol can achieve [41].

Let Π be an AKE scheme and κ a security parameter. Consider the following security game: $\mathcal{A}$ can query a polynomial number of times the oracles described above, except for Test oracle, which is queried only once on a fresh session. Let b be the bit chosen by the oracle Test. The game ends with $\mathcal{A}$ outputting b', a guess of b. We define the advantage of $\mathcal{A}$ as $\mathsf{Adv}_{\Pi,\mathcal{A}}(\kappa) = \Pr[b' = b] - 1/2$.

Definition 2. *Let* Π *be a AKE scheme and* κ *be the security parameter. We say that* Π *is secure if* $\mathsf{Adv}_{\Pi,\mathcal{A}}(\kappa) \leq \mathsf{negl}(\kappa)$*, for any adversary* $\mathcal{A}$*.*

2.2 Ring-Learning with Errors

In this section, we present the Ring Learning with Errors (RLWE) problem [46], a famous variant of the Learning with Errors (LWE) problem, firstly presented by Regev [52].

Let $\rho^n_{v,\alpha}(a) = \frac{1}{\alpha\sqrt{2\pi}} exp\left(\frac{-\|a-v\|^2}{2\alpha^2}\right)$ be the probability distribution of the Gaussian distribution over $\mathbb{R}^n$ centered at $v \in \mathbb{R}^n$ and with standard deviation α. We define the *discrete Gaussian distribution* over R_q centered at $v \in R_q$ and with standard deviation α by the probability distribution

$$\chi_{v,\alpha}(a) = \frac{\rho^n_{v,\alpha}(a)}{\rho^n_{v,\alpha}(R_q)}$$

for all $a \in R_q$. The subscript v is omitted when it is equal to zero.

We recall some basic facts about the ℓ_2 and ℓ_∞ norms: for all $a, b \in R_q$, $\|a.b\| \leq \sqrt{n}\,\|a\|\,.\,\|b\|$ and $\|a\|_\infty \leq \|a\|$.

Lemma 3 ([50])**.** *For any $\alpha = \omega(\sqrt{\log n})$, we have*

$$\mathsf{Pr}[\|x\| \geq \alpha\sqrt{n} : x \leftarrow \chi_\alpha] \leq 2^{-n+1}.$$

Let $s \leftarrow\!\!\$\, R_q$. The *RLWE distribution* $\mathcal{D}^{\mathrm{RLWE}}_{s,\chi_\alpha}$ samples $a \leftarrow\!\!\$\, R_q$ and $e \leftarrow\!\!\$\, \chi_\alpha$ and outputs $(a, as + e)$.

Definition 4 (Ring Learning with Errors)**.** The decision version of the *RLWE problem*, denoted by $\mathrm{RLWE}_{q,\chi_\alpha}$, asks to distinguish samples $(a, as+e) \leftarrow\!\!\$\, \mathcal{D}^{\mathrm{RLWE}}_{s,\chi_\alpha}$ from samples $(a, u) \leftarrow\!\!\$\ R_q \times R_q$.

It was shown that solving the RLWE assumption on average is at least as hard as solving worst-case lattice problems (namely the Approximate Shortest Independent Vector Problem), which is assumed to be hard for classical and quantum computers [46].

It is well-known that the RLWE problem is still hard when the secret s is sampled from the error distribution χ_α, instead of being chosen uniformly from R_q [8, 46, 47]. This is usually called the Hermite Normal Form-RLWE (we will denote it by HNF-$\mathrm{RLWE}_{q,\chi_\alpha}$) and it is proven to be as hard as $\mathrm{RLWE}_{q,\chi_\alpha}$.

It is also known that the RLWE problem is still hard when we scale the error of the sample by a constant t (which is co-prime with q), that is, $as + te$ [16]. Moreover, it is straightforward to prove that the RLWE problem is still hard when s is sampled from $\chi_{k\alpha}$ for $k = 2$ or $k = 3$, instead of being sampled from χ_α. We will use this fact to guarantee the security of our scheme.

2.3 Signal Function

We define the signal function which was firstly presented in [28] and has found numerous applications such as in key exchange (and its variants) [28, 56, 24], zero-knowledge proof [26] or oblivious transfer [17].

Let $E = \left\{ \left\lfloor \frac{q}{4} \right\rfloor, ..., \left\lfloor \frac{q}{4} \right\rceil \right\} \subset \mathbb{Z}_q$ and $E + 1 = \left\{ -\left\lfloor \frac{q}{4} \right\rfloor + 1, \left\lfloor \frac{q}{4} \right\rceil + 1 \right\}$. We define the functions $\mathsf{Sig}_0, \mathsf{Sig}_1 : \mathbb{Z}_q \to \{0, 1\}$ as

$$\mathsf{Sig}_0(a) = \begin{cases} 0, \text{if } a \in E \\ 1, \text{ if } a \notin E \end{cases} \quad \text{and} \quad \mathsf{Sig}_1(a) = \begin{cases} 0, \text{ if } a \in E + 1 \\ 1, \text{ if } a \notin E + 1 \end{cases}$$

The randomized function $\mathsf{Sig}_*(a)$ is given by choosing $b \leftarrow_\$ \{0, 1\}$ and outputting $\mathsf{Sig}_b(a)$. We extend Sig_* to a function $\mathsf{Sig} : R_q \to R_2$ on R_q given by $\mathsf{Sig}(a) = (\mathsf{Sig}_*(a_0), ..., \mathsf{Sig}_*(a_{n-1}))$, where $a = a_0 + a_1 X + \ldots a_{n-1} X^{n-1} \in R_q$.

Lemma 5. *Let $k \leftarrow_\$ R_q$ and $w \leftarrow \mathsf{Sig}(k)$. Then w follows a uniform distribution over R_2.*

Proof. If k is sampled uniformly from R_q, then each of its coefficients is sampled uniformly from $\mathbb{Z}_q$. Then, this means that each coefficient of k has a $1/2$ probability of being in the set E. Thus, the signal w of k is a uniformly chosen polynomial from R_2. □

We also define the extractor function $\mathsf{Mod}_2 : R_q \times R_2 \to R_2$ as

$$\mathsf{Mod}_2(a, w) = \left(a + w \frac{q-1}{2} \mod q \right) \mod 2.$$

Lemma 6 ([28])**.** *Let $q > 8$, $u, v \in R_q$ such that $\|u - v\|_\infty < q/4$ and $w \leftarrow \mathsf{Sig}(v)$. Then*

$$\mathsf{Mod}_2(u, w) = \mathsf{Mod}_2(v, w).$$

Recall that the min-entropy of a random variable V is defined by

$$-\log \left(\max_{s \in \mathcal{S}} \mathsf{Pr}[V = s] \right)$$

and says that it is infeasible for an adversary (even with unlimited computational power) to guess v chosen uniformly at random from V with probability greater than $2^{-\log(\max_{s \in \mathcal{S}} \mathsf{Pr}[V=s])}$.

Lemma 7 ([56])**.** *Let q be an odd prime and R_q be as above. For any $b \in R_2$ and any $v' \in R_q$, the output distribution of $\mathsf{Mod}_2(v + v', b)$ given $\mathsf{Sig}(v)$ has min-entropy of at least*

$$-n \log \left(\frac{1}{2} + \frac{1}{|E| - 1} \right)$$

where $v \leftarrow_\$ R_q$.

By the Lemma above, we have that, when $q > 203$, then $-n \log \left(\frac{1}{2} + \frac{1}{|E|-1} \right) > 0.97n$ [56].

3 Key exchange with reusable keys

Several attacks have been found on RLWE-based key exchanges [31, 23, 25]. All of these attacks rely on the fact that the signal function leaks information about the secret key. Hence, we want to show that there is no leakage of information in our protocol. In particular, we want to show that there is no leakage of information about the static secret key of each party. This can be proven due to the fact that, by pasteurizing the message sent by the other party, the signal function will look completely random.

In this section, we present a variant of the KE of [28] which allows for key reuse.

3.1 The protocol

Let $a \leftarrow_{\$} R_q$. Let $\mathsf{H}_1 : \{0,1\}^* \to \chi_\alpha$ be a random oracle whose outputs are sampled from χ_α.[6]

1. P_i does the following:
   - It samples $s_i, e_i \leftarrow_{\$} \chi_\alpha$ and computes $x_i = as_i + 2e_i$.
   - It sends x_i to P_j.
2. Upon receiving x_i from P_i, P_j does the following:
   - It samples $s_j, e_j \leftarrow_{\$} \chi$ and computes $y_j = as_j + 2e_j$.
   - It computes $c \leftarrow \mathsf{H}_1(\mathsf{P}_i, \mathsf{P}_j, x_i)$ and $d \leftarrow \mathsf{H}_1(\mathsf{P}_i, \mathsf{P}_j, x_i, y_j)$.
   - It samples $f_j \leftarrow_{\$} \chi_\alpha$ and computes $\bar{x}_i = x_i + ac + 2f_j$.
   - It samples $g_j \leftarrow_{\$} \chi_\alpha$ and computes $k_j = \bar{x}_i(s_j + d) + 2g_j$.
   - It computes $w_j \leftarrow \mathsf{Sig}(k_j)$ and sets the session key as
     $$\mathsf{sk}_j \leftarrow \mathsf{Mod}_2(k_j, w_j).$$
   - It sends (y_j, w_j) to P_i.
3. Upon receiving (y_j, w_j) from P_j, P_i does the following:
   - It computes $c \leftarrow \mathsf{H}_1(\mathsf{P}_i, \mathsf{P}_j, x_i)$ and $d \leftarrow \mathsf{H}_1(\mathsf{P}_i, \mathsf{P}_j, x_i, y_j)$.
   - It samples $f_i \leftarrow_{\$} \chi_\alpha$ and computes $\bar{y}_j = y_j + ad + 2f_i$.
   - It samples $g_i \leftarrow_{\$} \chi_\alpha$ and computes $k_i = \bar{y}_j(s_i + c) + 2g_i$;
   - It sets the the session key as $\mathsf{sk}_i \leftarrow \mathsf{Mod}_2(k_i, w_j)$.

[6] Note that such a hash function H_1 can be trivially implemented by means of a usual hash function $\widetilde{\mathsf{H}}_1$ and an algorithm S that samples according to χ_α, and by using the value $\widetilde{\mathsf{H}}(x)$ as the seed in S.

Discussion. The main idea of this new protocol is that an adversary gains nothing by sending something that it is not an RLWE sample. To see this, assume that party P_i is dishonest, and controlled by an adversary $\mathcal{A}$, (as in the setup of the attacks of [31, 23]) and sends x_i to P_j. If x_i is an HNF-RLWE sample, then $\bar{x}_i$ is also a HNF-RLWE sample for which P_i has the corresponding secret. However, when x_i is not an HNF-RLWE sample, but rather a value that follows some other arbitrary distribution over R_q, then we prove that $\bar{x}_i$ follows a distribution that is statistically close to the uniform distribution from the point of view of $\mathcal{A}$. This happens because H is modeled as a random oracle and, thus, $\mathcal{A}$ has no control over it. In particular, it has no control over the value $a\mathsf{H}(x_i) + e$ since $\mathsf{H}(x_i)$ is independent of x_i (since H is a random oracle) and e is sampled by the other party.

Also, we remark that the probability of $\mathcal{A}$ finding x_i such that $\bar{x}_i$ is some particular value is equal to the probability of honestly sampling HNF-RLWE samples and getting the same particular value (which should be negligible).

Correctness. We prove that the scheme is correct with overwhelming probability.

Lemma 8. *Suppose that* $q > 16(4\alpha^2 n^{3/2} + \alpha\sqrt{n})$. *Then* $\mathsf{sk}_i = \mathsf{sk}_j$, *except with negligible probability.*

Proof. By Lemma 6, we have to show that $\|k_i - k_j\|_\infty < q/4$. First, note that

$$k_i = a\tilde{s} + 2\tilde{e}_i \quad \text{and} \quad k_j = a\tilde{s} + 2\tilde{e}_j$$

where

$$\begin{aligned}\tilde{s} =& (s_i + c)(s_j + d)\\ \tilde{e}_i =& e_j s_i + e_j c + f_j s_i + f_i c + g_i\\ \tilde{e}_j =& e_i s_j + e_i d + f_i s_j + f_j d + g_j.\end{aligned}$$

Recall that $\|a.b\| \leq \sqrt{n}\,\|a\|\,.\,\|b\|$ for any $a, b \in R_q$. Plugging this fact together with the triangular inequality and Lemma 3, we have that

$$\|k_i - k_j\|_\infty < 2(8\alpha^2 n^{3/2} + 2\alpha\sqrt{n}).$$

Since, by assumption, we have that

$$q > 8(8\alpha^2 n^{3/2} + 2\alpha\sqrt{n}),$$

then $\|k_i - k_j\|_\infty < q/4$ and correctness of the protocol follows. □

3.2 Security against passive adversaries

Let $\mathcal{A}$ be an adversary. Consider the following security game for KE protocols. $\mathcal{A}$ is given a honestly generated transcript of the KE. Then, a random bit b is chosen uniformly at random. If $b = 0$, then $\mathcal{A}$ is given a uniformly chosen key

and, if $b = 1$, then $\mathcal{A}$ is given the actual session key k. Finally, $\mathcal{A}$ must output a bit b', which a guess of b. We define the advantage of $\mathcal{A}$ to be $\mathsf{Adv}^{\mathsf{pas}}_{\mathsf{KE},\mathcal{A}}(\kappa) = \Pr[b = b'] - 1/2$ and say that the KE scheme is *secure against passive adversaries* if $\mathsf{Adv}^{\mathsf{pas}}_{\mathsf{KE},\mathcal{A}}(\kappa) \leq \mathsf{negl}(\kappa)$, for any adversary $\mathcal{A}$.

Theorem 9. *The scheme is secure against passive adversary, given that q is a prime as in Lemma 8 and* HNF-RLWE$_{q,\chi_\alpha}$ *is hard*

Proof. The security proof of the scheme for passive adversaries follows the same line as the proof in [28]. We omit it here. □

3.3 Robustness of the scheme to key reuse

We prove that the scheme is robust to key reuse. That is, it is infeasible for an adversary to get information about the other party's secret key s, even when the same keys are reused in several executions of the protocol.

We say that a KE scheme is *robust to key reuse* if it is robust to key reuse for both parties involved in the protocol (we formally define robustness for each party below).

Lemma 10 ([26])**.** *Let ϕ be an arbitrary distribution over R_q and let ψ be a distribution over R_q which is statistically close to the uniform distribution over R_q. Let $x, y \in R_q$ such that $x \hookleftarrow_{\$} \phi$ and $y \hookleftarrow_{\$} \psi$. Then, the distribution of $\bar{x} = x + y$ is statistically close to uniform.*

Proof. The proof is presented in [26], however we present it here for completeness. For any $r \in R_q$, we have that

$$\begin{aligned}
\Pr[x + y = r] &= \sum_{i \in R_q} \Pr[x = r - i] \Pr[y = i] \\
&= \sum_{i \in R_q} \left(\frac{1}{q^n} + \mathsf{negl}_i(\kappa) \right) \Pr[x = r - i] \\
&= \frac{1}{q^n} \sum_{i \in R_q} \Pr[x = r - i] + \sum_{i \in R_q} \mathsf{negl}_i(\kappa) \Pr[x = r - i] \\
&\leq \frac{1}{q^n} + \sum_{i \in R_q} \mathsf{negl}_i(\kappa)
\end{aligned}$$

and the sum $\sum_{i \in R_q} \mathsf{negl}_i(\kappa)$ is a negligible value. Hence, the distribution of $\bar{x} = x + y$ is statistically close to the uniform distribution. □

Lemma 11. *Let $s \hookleftarrow_{\$} \chi_\alpha$. Given $(a, y = as + e) \hookleftarrow_{\$} \mathcal{D}^{\mathrm{RLWE}}_{s,\chi_\alpha}$, the probability $\Pr[y = r]$ for any $r \in R_q$ is less or equal to $1/q^n$.*

Proof. First, note that, if $a \in R_q$ is uniformly chosen, then the probability that $\Pr[as = r]$ for some $r \in R_q$ is $1/q^n$. To see this, note that the product $a_0 s_0$ is uniform in $\mathbb{Z}_q$, for $a_0, s_0 \in Z_q$ (since q is prime, then $\mathbb{Z}_q$ forms a field). Hence,

each coefficient of the product of two polynomials $a = a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$ and $s = s_0 + s_1 X + \cdots + s_{n-1} X^{n-1}$ in R_q is nothing but a sum of (some of) the coefficients of a and s. Since the coefficients of a are uniformly chosen from $\mathbb{Z}_q$, then the product $a_k s_\ell$ is also uniform in Z_q, for any $k, \ell = 1, \ldots, n-1$. Therefore as is uniform in R_q since its coefficients are sums of uniformly chosen values from $\mathbb{Z}_q$.

Now, using Lemma 10, we conclude that the value $as + e$ follows a distribution statistically close to the uniform distribution over R_q. We conclude that $\Pr[as + e = r] = 1/q^n + \mathsf{negl}(\kappa)$. □

From the lemma presented above, we immediately get the following corollary.

Corollary 12. *The distribution $\mathcal{D}^{\mathrm{RLWE}}_{s,\chi_\alpha}$ is statistically close to the uniform distribution over R_q.*

Proof. This is a direct consequence of the previous lemma. □

Corollary 13. *Let H be a random oracle whose outputs are sampled from χ_α and let $x \leftarrow\!\!\$\, \phi$, where ϕ is a distribution over R_q, different from $\mathcal{D}^{\mathrm{RLWE}}_{s,\chi_\alpha}$, where $s \leftarrow\!\!\$\, \chi_\alpha$. Then, the distribution of $\bar{x} = x + a\mathsf{H}(x) + e$, where $e \leftarrow\!\!\$\, \chi_\alpha$, is statistically close to the uniform distribution over R_q, in the ROM.*

Proof. The proof follows from Lemma 10, Corollary 12 and by noting that the distribution of $\mathsf{H}(x)$ is independent of the distribution of x (because H is a random oracle). □

Key reuse for party P_j. Let $\mathcal{A}$ be an adversary. Consider the following security game: $\mathcal{A}$ is allowed to open as many sessions as it wants with party P_j (always playing the role of P_i). At some point, a key is exchanged between P_i and P_j with $\mathcal{A}$ passively observing. Observe that $\mathcal{A}$ is not given access to the secret key of P_i. Then, a random bit b is chosen uniformly at random. If $b = 0$, then $\mathcal{A}$ is given a uniformly chosen key and, if $b = 1$, then $\mathcal{A}$ is given the actual session key k (computed by P_j) between P_i and P_j. Finally, $\mathcal{A}$ must output a bit b', which a guess of b. We define the advantage of $\mathcal{A}$ to be $\mathsf{Adv}^{\mathsf{krj}}_{\mathsf{KE},\mathcal{A}}(\kappa) = \Pr[b = b'] - 1/2$ and say that the KE scheme is *robust to key reuse for* P_j if $\mathsf{Adv}^{\mathsf{krj}}_{\mathsf{KE},\mathcal{A}}(\kappa) \leq \mathsf{negl}(\kappa)$, for any adversary $\mathcal{A}$.

Theorem 14. *Let q be as in Lemma 8. The proposed KE scheme is robust to key reuse for party P_j in the ROM, given that the* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. First, we show that, whatever the strategy used by the adversary$\mathcal{A}$, it cannot get any information on the secret key s_j of party P_j. When interacting with P_j, $\mathcal{A}$ sends a value $x \in R_q$. There are two cases to consider:

1. $(a, x) \leftarrow\!\!\$\, \mathcal{D}^{\mathrm{RLWE}}_{s,\chi_\alpha}$, where $s \leftarrow\!\!\$\, \chi_\alpha$.
2. The value x follows some other distribution ϕ over R_q, different from $\mathcal{D}^{\mathrm{RLWE}}_{s,\chi_\alpha}$, where $s \leftarrow\!\!\$\, \chi_\alpha$.

The case 1 reduces to the case of passive security. Hence, in this case, the adversary $\mathcal{A}$ cannot get information about s_j (the secret of P_j).

In the second case, suppose that $\mathcal{A}$ sends x sampled from an arbitrary distribution ϕ. Then, by Lemma 13, the distribution of $\bar{x} = x + a\mathsf{H}(x) + e$, where e is sampled from χ_α by P_j, is statistically close to the uniform distribution. Hence, by the HNF-RLWE assumption, the key k_j is indistinguishable from a uniformly chosen value of R_q from the point-of-view of $\mathcal{A}$. Then, by Lemma 5, the signal w_j is uniform in R_2. By a simple hybrid argument, we can replace the key k_j and the signal w_j of each of these sessions by random elements of R_q and R_2, by Lemma 5. We conclude that, in this case, it is infeasible for $\mathcal{A}$ to get s_j, except with negligible probability.

If it is infeasible for $\mathcal{A}$ to get information about s_j when interacting with P_j, then the case where $\mathcal{A}$ passively observes the execution the protocol between P_i and P_j falls in the case of Theorem 9. Thus, it follows that $\mathcal{A}$ has a negligible advantage in the game and, therefore, the scheme is robust to key reuse for party P_j. □

Key reuse for party P_i. Similarly, we define the concept of robustness to key reuse for P_i.

Consider the following security game for any adversary $\mathcal{A}$: $\mathcal{A}$ is allowed to open as many sessions as it wants with party P_i (always playing the role of P_j). At some point, a key is exchanged between P_i and P_j with $\mathcal{A}$ watching passively. Observe that $\mathcal{A}$ is not given access to the secret key of P_j. Similarly to the previous case, a random bit b is chosen uniformly at random. If $b = 0$, then $\mathcal{A}$ is given a uniformly chosen key and, if $b = 1$, then $\mathcal{A}$ is given the actual session key k (computed by party P_i) between P_i and P_j. Finally, $\mathcal{A}$ must output a bit b', which a guess of b. We define the advantage of $\mathcal{A}$ to be $\mathsf{Adv}^{\mathsf{kri}}_{\mathsf{KE},\mathcal{A}}(\kappa) = \Pr[b = b'] - 1/2$ and say that the KE scheme is *robust to key reuse for* P_i if $\mathsf{Adv}^{\mathsf{kri}}_{\mathsf{KE},\mathcal{A}}(\kappa) \leq \mathsf{negl}(\kappa)$, for any adversary $\mathcal{A}$.

Theorem 15. *Let q be as in Lemma 8. The proposed KE scheme is robust to key reuse for party P_i in the ROM, given that the* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. The analysis is similar to the proof of Theorem 14. We omit it for briefness. □

3.4 Efficiency and comparison

We compare our scheme with Ding's KE [28] (based on the RLWE assumption) in terms of computational complexity. The comparison is presented in Table 2. Our proposal maintains the same communication complexity: The same amount of information is exchanged in the same number of rounds. As we can see in Table 2, we obtain a small computational overhead in order to guarantee robustness to key reuse.

	Rounds	Communication complexity	Multiplications in R_q	Samples	Key reuse
Ding's KE [28]	2	$2n \log q + n$	4	4	✗
Ours	2	$2n \log q + n$	6	8	✓

Table 2: Comparison with other KE schemes.

4 The AKE Protocol

Let $a \leftarrow\!\!\$\, R_q$ be a uniformly chosen public element of R_q. The static public key of party P_i is $p_i = as_i + 2e_i$ where $s_i, e_i \leftarrow\!\!\$\, \chi_\alpha$. Its static secret key is s_i. Similarly for party P_j, its static public key is $p = as_j + 2e_j$ and its static secret key is s_j. Let $\mathsf{H}_1 : \{0,1\}^* \to \chi_\alpha$ be a hash function whose outputs are sampled from χ_α and let $\mathsf{H}_2 : \{0,1\}^* \to \{0,1\}^\kappa$ be a key derivation function. Both of these functions are modeled as random oracles.

The protocol is composed by three algorithms: Initiate, Respond and Complete. We specify it in full detail:

1. Initiate: P_i does the following:
   - It samples $r_i, f_i \leftarrow\!\!\$\, \chi_\alpha$ and computes $x_i = ar_i + 2f_i \mod q$.
   - It sends x_i to P_j.
2. Respond: Upon receiving x_i, P_j does the following:
   - It samples $r_j, f_j \leftarrow\!\!\$\, \chi_\alpha$ and computes $y_j = ar_j + 2f_j \mod q$.
   - It computes $c \leftarrow \mathsf{H}_1(\mathsf{P}_i, \mathsf{P}_j, x_i)$ and $d = \mathsf{H}_1(\mathsf{P}_i, \mathsf{P}_j, x_i, y_j)$.
   - It samples $g_j \leftarrow\!\!\$\, \chi_\alpha$ and computes $\bar{x}_i = x_i + ac + 2g_j$.
   - It samples $h_j \leftarrow\!\!\$\, \chi_\alpha$ and computes
   $$k_j = (p_i + \bar{x}_i)(s_j + r_j + d) - p_i s_j + 2h_j.$$
   - It computes $w_j \leftarrow \mathsf{Sig}(k_j)$ and $\sigma_j \leftarrow \mathsf{Mod}_2(k_j, w_j)$.
   - It sets $\mathsf{sk}_j \leftarrow \mathsf{H}_2(\mathsf{P}_i, \mathsf{P}_j, x_i, y_j, w_j, \sigma_j)$ as the shared key.
   - It sends (y_j, w_j) to P_i.
3. Complete: Upon receiving (y_j, w_j), P_i does the following:
   - It sets $c \leftarrow \mathsf{H}_1(\mathsf{P}_i, \mathsf{P}_j, x_i)$ and $d = \mathsf{H}_1(\mathsf{P}_i, \mathsf{P}_j, x_i, y_j)$
   - It samples $g_i \leftarrow\!\!\$\, \chi_\alpha$ and computes $\bar{y}_j = y_j + ad + 2g_i$.
   - It samples $h_i \leftarrow\!\!\$\, \chi_\alpha$ and computes
   $$k_i = (p_j + \bar{x}_j)(s_i + r_i + c) - p_j s_i + 2h_i.$$
   - It computes $\sigma_i \leftarrow \mathsf{Mod}_2(k_i, w_j)$.
   - It sets $\mathsf{sk}_i \leftarrow \mathsf{H}_2(\mathsf{P}_i, \mathsf{P}_j, x_i, y_j, w_j, \sigma_i)$ as the shared key.

The following lemma proves the correctness of the scheme, that is, parties P_i and P_j end up with the same key after executing the protocol.

Lemma 16 (Correctness)**.** *If* $q > 8\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right)$ *then* $\mathsf{sk}_i = \mathsf{sk}_j$*, except with negligible probability.*

Proof. To prove the correctness of the scheme is it enough to show that $\sigma_i = \sigma_j$. By Lemma 6, we have to show that $\|k_i - k_j\|_\infty < q/4$. First, note that

$$k_i = a\tilde{s} + 2\tilde{e}_i \quad \text{and} \quad k_j = a\tilde{s} + 2\tilde{e}_j$$

where

$$\begin{aligned}
\tilde{s} =& r_j s_i + d s_i + s_j r_i + r_j r_i + d r_i + s_j c + r_j c + dc \\
\tilde{e}_i =& f_j s_i + g_i s_i + e_j r_i + f_j r_i + g_i r_i + e_j c + f_j c + g_i c + h_i \\
\tilde{e}_j =& f_i s_j + g_j s_j + e_i r_j + f_i r_j + g_j r_j + e_i d + f_i d + g_j d + h_j.
\end{aligned}$$

Recall that $\|a.b\| \leq \sqrt{n}\,\|a\|\,.\,\|b\|$ for any $a, b \in R_q$. Plugging this fact together with the triangular inequality and Lemma 3, we have that

$$\|k_i - k_j\|_\infty \leq 2\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right).$$

Since, by assumption, we have that

$$q > 8\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right),$$

then $\|k_i - k_j\|_\infty < q/4$ and the correctness of the protocol follows. □

5 Security proof for the AKE scheme

Before proving security of the scheme in the BR-model, remark that no information about the secret static keys of each party is leaked during the execution of each session. This is guaranteed by Theorem 14 and Theorem 15.

We present the result that guarantees the security of the proposed scheme in the BR-model.

Theorem 17. *Let* κ *be the security parameter. Suppose that* n *is a power of* 2 *such that* $0.97n \geq 2\kappa$*,* q *is a prime such that* $q > 203$ *and* $q > 8\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right)$*, and the* HNF-RLWE$_{q,\chi_\alpha}$ *is hard. Then, the proposed AKE scheme is secure in the BR-model in the ROM.*

The rest of this section is dedicated to the proof of this Theorem, which follows from Lemmas 18, 25, 32, 39 and 46.

First, note that the test session may have a matching session or not. When it has a matching session, then the adversary can corrupt parties and recover their static secret key, by the definition of fresh session (Definition 1). We enumerate the several types of adversaries:

- Adversary chooses a test session that has a matching session:
  - **Type** $\mathcal{A}_1$: Let $\mathsf{sid} = \big(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*, x_i^*, (y_j^*, w_j^*)\big)$ be the test session where y_j^* was outputted by $\mathsf{Respond}\,\big(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*\big)$.
  - **Type** $\mathcal{A}_2$: Let $\mathsf{sid} = \big(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*, (y_j^*, w_j^*)\big)$ be the test session where x_i^* was outputted by $\mathsf{Initiate}\,\big(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*\big)$ and P_i^* either completes the session with y_j^* or it never completes it.
- Adversary chooses a test session that does not have a matching session:
  - **Type** $\mathcal{A}_3$: Let $\mathsf{sid} = \big(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*, (y_j^*, w_j^*)\big)$ be the test session where x_i^* was not outputted by $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*)$.
  - **Type** $\mathcal{A}_4$: Let $\mathsf{sid} = \big(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*, x_i^*, (y_j^*, w_j^*)\big)$ be the test session where (y_j^*, w_j^*) was not outputted by $\mathsf{Respond}(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*)$.
  - **Type** $\mathcal{A}_5$: Let $\mathsf{sid} = \big(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*, (y_j^*, w_j^*)\big)$ be the test session where x_i^* was outputted by $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*)$ but P_i^*'s session is completed with $y_j' \neq y_j^*$.

Weak perfect forward secrecy (wPFS) is obtained from **Type** $\mathcal{A}_1$ and **Type** $\mathcal{A}_2$ adversaries, since these types of adversaries can corrupt parties involved in the test session. Observe that **Type** $\mathcal{A}_3$, **Type** $\mathcal{A}_4$ and **Type** $\mathcal{A}_5$ adversaries cannot query $\mathsf{Corrupt}$ on either P_i^* or P_j^*, since sid^* has no matching session.

The idea of the proof is very simple. Consider, for example, a session belonging to party P_i^*, when interacting with P_j^*. Either P_j^*'s static public key p_j^* or ephemeral public key y_j^* are indistinguishable from a uniformly chosen value to the adversary: When the test session has a matching session, the adversary is allowed to get the static secret keys of both parties but it is not allowed to modify the messages exchanged between them (by the BR-model).Therefore, in this case the ephemeral public key of P_j^* can be replaced by a uniformly chosen value. Thus, the key obtained by P_i^* is indistinguishable from a uniformly chosen value by the HNF-RLWE assumption. When the test session does not have a matching session, then the adversary is not allowed to get static secret keys. Thus, the static public key of P_j^* is indistinguishable from a uniformly random value and, by the same reasoning, the shared key obtained by P_i^* is also indistinguishable from a uniformly chosen value. In this case, we just have to show that the simulator is able to simulate the execution of sessions involving P_j^* which it can since it knows all the static secret keys of all other parties.

5.1 Test session has a matching session

By the definition of freshness, when the test session has a matching session, then the adversary is allowed to corrupt both parties, get their static secret key and eavesdrop the communication.

Here, we have two possible types of adversaries: one that uses a test session belonging to the initiator and other that uses a test session belonging to the responder.

5.1.1 Adversary Type $\mathcal{A}_1$

Lemma 18. *For any adversary $\mathcal{A}$ of **Type** $\mathcal{A}_1$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ is negligible in the ROM, given that $0.97n \geq 2\kappa$, q is a prime such that $q > 203$ and $q > 8\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right)$, and the* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

The proof of this lemma follows from the sequence of games $G_{1,0}, \ldots, G_{1,4}$.

Game $G_{1,0}$. The simulator chooses $a \leftarrow\!\!\$\, R_q$ and creates static public keys for each user, following the protocol. The simulator $\mathcal{S}$ chooses $(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*, x_i^*, (y_j^*, w_j))$ as the test session where $\mathsf{P}_i^*, \mathsf{P}_j^* \leftarrow\!\!\$\, \{\mathsf{P}_1, \ldots, \mathsf{P}_N\}$, $s_i^*, s_j^* \leftarrow\!\!\$\, \{1, \ldots, m\}$, x_i^* is outputted by a query $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*)$ on the s_i^*-th session of P_i^* and y_i^* is outputted by $\mathsf{Respond}(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*)$ on the s_j^*-th session of P_j^*. $\mathcal{S}$ runs internally $\mathcal{A}$ and simulates the oracles in the following way:

- H_1 and H_2: let L_1 and L_2 be two lists of pairs (q, h) (i.e., query made to the random oracles and their respective response). As usual, if the query q is made to H_1, $\mathcal{S}$ checks if there is a pair (q, h) in L_1. If there is, it returns h, else $\mathcal{S}$ samples $h \leftarrow\!\!\$\, \chi_\alpha$, returns h and keeps $(q, h) \in L_1$. If the query q is made to H_2, $\mathcal{S}$ checks if there is a pair (q, h) in L_2. If there is, it returns h, else $\mathcal{S}$ chooses $h \leftarrow\!\!\$\, R_q$, returns h and keeps $(q, h) \in L_2$.
- $\mathsf{Initiate}$, $\mathsf{Respond}$ and $\mathsf{Complete}$ are simulated following the AKE protocol.
- $\mathsf{skReveal}$ and $\mathsf{Corrupt}$ as described in Section 2.1.
- $\mathsf{Test}(\mathsf{sid})$: Let $\mathsf{sid} = (\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$ be the test session queried by $\mathcal{A}$. If $(\mathsf{P}_i, \mathsf{P}_j) \neq (\mathsf{P}_i^*, \mathsf{P}_j^*)$ or x_i and is not outputted in the s_i^*-th session of P_i^* or y_j is not outputted in the s_j^* session of P_j^*, $\mathcal{S}$ aborts the execution. Else, it chooses $b \leftarrow\!\!\$\, \{0, 1\}$ and returns either a random key $\mathsf{sk}_i' \leftarrow\!\!\$\, \{0, 1\}^\kappa$, if $b = 0$, or the session key sk_i of session sid, if $b = 1$.

Claim 19. *The probability that $\mathcal{S}$ aborts in $G_{1,0}$ is $\frac{1}{m^2 N^2}$.*

Game $G_{1,1}$**.** $\mathcal{S}$ simulates the oracles as in $G_{1,0}$ except for $\mathsf{Complete}$:

- When it is queried on $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$, $\mathcal{S}$ does the following: If $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i and (y_j, w_j) was outputted on the s_j^*-th session of P_j, then S sets $\mathsf{sk}_i = \mathsf{sk}_j$.

  Else, it simulates the oracles as in game $G_{1,0}$.

Claim 20. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes between games $G_{1,0}$ and $G_{1,1}$ is negligible.*

Proof. Note that completeness of the protocol still holds in game $G_{1,1}$. Hence, there is no difference between the games $G_{1,0}$ and $G_{1,1}$. □

Game $G_{1,2}$. $\mathcal{S}$ simulates the oracles as in $G_{1,1}$, except for Initiate:

- When it is queried $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j)$, $\mathcal{S}$ does the following: If $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i, then $\mathcal{S}$ samples $x_i \leftarrow\!\!\$\, R_q$ (instead of computing $x_i = ar_i + 2f_i$).

Claim 21. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes between games $G_{1,1}$ and $G_{1,2}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. It is straightforward to construct a algorithm that decides the HNF-RLWE problem, if there is an algorithm that can distinguish both games. □

Game $G_{1,3}$. $\mathcal{S}$ simulates the oracles as in $G_{1,1}$, except for Complete:

- When it is queried $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$, $\mathcal{S}$ does the following: If $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i and (y_j, w_j) was not outputted on the s_j^*-th session of B, then $\mathcal{S}$ samples $k_i \leftarrow\!\!\$\, R_q$.

Claim 22. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes between games $G_{1,2}$ and $G_{1,3}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. In this case, we do not know which is the distribution of the value y_j since it was not outputted by the oracle Respond. However, by Corollary 13, we have that the pasteurization $\bar{y}_j$ of y_j is statistically close to a uniformly chosen value, independently of the distribution of y_j. Hence, we consider the key computed by P_i which is

$$k_i = (p_j + \bar{y}_j)(s_i + r_i + c) - p_j s_i + 2h_i.$$

Rewriting the expression, we have that

$$k_i = \bar{y}_j(s_i + r_i + c) + 2h_i + p_j(r_i + c).$$

By the HNF-RLWE assumption, we have that $\bar{y}_j(s_i + r_i + c) + 2h_i$ is indistinguishable from a uniformly chosen value in R_q, since $\bar{y}_j$ is uniform in R_q, and $(s_i + r_i + c)$ and h_i are discrete Gaussian samples. Observe that the HNF-RLWE assumption still holds when the secret is chosen from $\chi_{\sqrt{3}\alpha}$ and the error from the distribution $\chi_{\sqrt{2}\alpha}$.[7]

Hence, consider the following hybrid game $G'_{1,2}$, where $\mathcal{S}$ chooses $r_i \leftarrow\!\!\$\, R_q$ and computes $k_i = r_i + p_j(r_i + c)$. From the reasoning above, it is infeasible for any adversary $\mathcal{A}$ to distinguish $G_{1,2}$ from the hybrid game $G'_{1,2}$, given that the HNF-RLWE assumption holds.

Since r_i is uniform in R_q then, by Lemma 10 we have that k_i is also uniform in R_q. Hence $G'_{1,2}$ and $G_{1,3}$ are indistinguishable from the point of view of the adversary $\mathcal{A}$. □

[7] It is trivial to build the reduction: Given a HNF-RLWE sample $(a, y = as + e)$ where $s \leftarrow\!\!\$\, \chi_\alpha$ and $e \leftarrow\!\!\$\, \chi_\alpha$, just choose $s', s'' \leftarrow\!\!\$\, \chi_\alpha$ and $e \leftarrow\!\!\$\, \chi_\alpha$, and compute $y' = y + a(s' + s'') + e'$.

Game $G_{1,4}$. $\mathcal{S}$ simulates the oracles as in $G_{1,3}$, except for $\mathsf{Respond}$:

- When it is queried on $\mathsf{Respond}(\Pi, R, \mathsf{P}_i, \mathsf{P}_j, x_i)$, $\mathcal{S}$ does the following: if $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and x_i was outputted in the s_i^*-th session of P_i^* and it is the s_j^*-th session of P_j^*, $\mathcal{S}$ chooses $y_j \leftarrow_\$ R_q$ and the key $k_j \leftarrow_\$ R_q$. It sends (y_j, w_j) where $w_j \leftarrow \mathsf{Sig}(k_j)$. Else, it simulates $\mathsf{Respond}$ as in $G_{1,3}$.

Claim 23. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes between games $G_{1,3}$ and $G_{1,4}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. The key computed by P_j is equal to

$$k_j = (p_i + \bar{x}_i)(s_j + r_j + d) - p_i s_j + 2h_j.$$

Using the same argument as in the proof of Claim 22, we conclude that games $G_{1,3}$ and $G_{1,4}$ are indistinguishable from the point of view of the adversary. □

Finally, we prove that the advantage of any adversary in the game $G_{1,4}$ is negligible.

Claim 24. *For any adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ in game $G_{1,4}$ is negligible, given that $0.97n > 2\kappa$.*

Proof. Since k_i is chosen uniformly from R_q, we have that σ_i has high min-entropy, even when w_j is given, by Lemma 7. In particular, when $0.97n > 2\kappa$ then the probability that $\mathcal{A}$ queries H_2 on input $(\mathsf{P}_i, \mathsf{P}_j, x_i, y_j, w_j, \sigma_j)$ is at most $2^{-0.97n} + \mathsf{negl}(\kappa)$. □

5.1.2 Adversary Type $\mathcal{A}_2$

Lemma 25. *For any adversary $\mathcal{A}$ of **Type** $\mathcal{A}_2$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ is negligible in the ROM, given that $0.97n \geq 2\kappa$, q is a prime such that $q > 203$ and $q > 8\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right)$, and the* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

The proof of this lemma follows from the sequence of games $G_{2,0}, \ldots, G_{2,4}$.

Game $G_{2,0}$. Similar to $G_{1,0}$ but now $\mathcal{S}$ chooses $\mathsf{sid} = \left(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*, (y_j^*, w_j^*)\right)$ as the test session where $\mathsf{P}_i^*, \mathsf{P}_j^* \leftarrow_\$ \{\mathsf{P}_1, \ldots, \mathsf{P}_N\}$, $s_i^*, s_j^* \leftarrow_\$ \{1, \ldots, m\}$, x_i^* is outputted by a query $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^j)$ on the s_i^*-th session of P_i^* and y_i^* is outputted by $\mathsf{Respond}(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*)$ on the s_j^*-th session of P_j^*. $\mathcal{S}$ runs internally $\mathcal{A}$ and simulates the oracles as in $G_{1,0}$ except for Test:

- $\mathsf{Test}(\mathsf{sid})$: Let $\mathsf{sid} = (\Pi, R, \mathsf{P}_j, \mathsf{P}_i, x_i, (y_j, w_j))$ be the test session queried by $\mathcal{A}$. If $(\mathsf{P}_i, \mathsf{P}_j) \neq (\mathsf{P}_i^*, \mathsf{P}_j^*)$ or x_i and is not outputted in the s_i^*-th session of P_i^* or y_j is not outputted in the s_j^* session of P_j^*, $\mathcal{S}$ aborts the execution. Else, it chooses $b \leftarrow_\$ \{0,1\}$ and returns either a random key $\mathsf{sk}_i' \leftarrow_\$ \{0,1\}^\kappa$, if $b = 0$, or the session key sk_i of session sid, if $b = 1$.

Claim 26. *The probability that $\mathcal{S}$ aborts in $G_{2,0}$ is $\frac{1}{m^2 N^2}$.*

Game $G_{2,1}$. $\mathcal{S}$ simulates the oracles as in $G_{2,0}$, except for $\mathsf{Complete}$:

- When it is queried $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$, $\mathcal{S}$ does the following: if $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i^* and (y_j, w_j) was outputted on the s_j^* session of P_j, $\mathcal{S}$ sets $\mathsf{sk}_i = \mathsf{sk}_j$. Else, it simulates the oracles as in $G_{2,0}$.

Claim 27. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{2.0}$ and $G_{2,1}$ is negligible.*

Proof. Similar to the proof of Claim 20. □

Game $G_{2,2}$. $\mathcal{S}$ simulates the oracles as in $G_{2,1}$, except for $\mathsf{Initiate}$:

- When it is queried $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j)$, $\mathcal{S}$ does the following: if $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i^*, $\mathcal{S}$ samples $x_i \leftarrow\!\!\$\, R_q$, instead of computing $x_i = ar_i + 2f_i$.

Claim 28. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{2,1}$ and $G_{2,2}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. Similar to the proof of Claim 21. □

Game $G_{2,3}$. $\mathcal{S}$ simulates the oracles as in $G_{2,2}$, except for $\mathsf{Complete}$:

- When it is queried $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$, $\mathcal{S}$ does the following: If $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i and (y_j, w_j) was not outputted on the s_j^*-th session of B, then $\mathcal{S}$ samples $k_i \leftarrow\!\!\$\, R_q$.

Claim 29. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes between games $G_{1,2}$ and $G_{1,3}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. Similar to the proof of Claim 22. □

Game $G_{2,4}$. $\mathcal{S}$ simulates the oracles as in $G_{2,3}$, except for $\mathsf{Respond}$:

- When it is queried on $\mathsf{Respond}(\Pi, R, \mathsf{P}_i, \mathsf{P}_j, x_i)$, $\mathcal{S}$ does the following: if $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and x_i was outputted in the s_i^*-th session of P_i^* and it is the s_j^*-th session of P_j^*, $\mathcal{S}$ chooses $y_j \leftarrow\!\!\$\, R_q$ and the key $k_j \leftarrow\!\!\$\, R_q$. It sends (y_j, w_j) where $w_j \leftarrow \mathsf{Sig}(k_j)$. Else, it simulates $\mathsf{Respond}$ as in $G_{1,3}$.

Claim 30. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes between games $G_{1,3}$ and $G_{1,4}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. Similar to the proof of Claim 23. □

Claim 31. *For any adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ in game $G_{2,4}$ is negligible, given that the HNF-RLWE assumption holds and $0.97n \geq 2\kappa$.*

Proof. Similar to the proof of Claim 24. □

5.2 Test session does not have a matching session

When the test session does not have a matching session, then the adversary is not allowed to corrupt parties involved in the test session.

However, here we cannot replace the messages sent by the parties by random values since these messages do not have to be created according to the protocol. Since the adversary cannot ask $\mathsf{Corrupt}$ for none of these parties, then we can replace their static public key by random values. It should be infeasible for an adversary to notice that the public key of a party was replaced a random value by the HNF-RLWE assumption. When the static public key of a party is replaced by a random value, then the key computed by another party interacting with the first is also indistinguishable from uniformly random value, by the HNF-RLWE. So, we may conclude that the advantage of an adversary is negligible.

5.2.1 Adversary Type $\mathcal{A}_3$

Lemma 32. *For any adversary $\mathcal{A}$ of **Type** $\mathcal{A}_3$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ is negligible in the ROM, given that $0.97n \geq 2\kappa$, q is a prime such that $q > 203$ and $q > 8\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right)$, and the* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

The proof of this lemma follows from Claims 33, ..., 38.

Game $G_{3,0}$. Similar to $G_{1,0}$ but now $\mathcal{S}$ chooses $\mathsf{sid} = \left(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*, (y_j^*, w_j^*)\right)$ as the test session where $\mathsf{P}_i^*, \mathsf{P}_j^* \leftarrow_\$ \{\mathsf{P}_1, \ldots, \mathsf{P}_N\}$, $s_j^* \leftarrow_\$ \{1, \ldots, m\}$, y_i^* is outputted by $\mathsf{Respond}(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*)$ on the s_j^*-th session of P_j^*. $\mathcal{S}$ runs internally $\mathcal{A}$ and simulates the oracles as in $G_{1,0}$ except for Test:

- $\mathsf{Test}(\mathsf{sid})$: Let $\mathsf{sid} = (\Pi, R, \mathsf{P}_j, \mathsf{P}_i, x_i, (y_j, w_j))$ be the test session queried by $\mathcal{A}$. If $(\mathsf{P}_i, \mathsf{P}_j) \neq (\mathsf{P}_i^*, \mathsf{P}_j^*)$ or y_j is not outputted in the s_j^* session of P_j^*, $\mathcal{S}$ aborts the execution. Else, it chooses $b \leftarrow_\$ \{0,1\}$ and returns either a random key $\mathsf{sk}_i' \leftarrow_\$ \{0,1\}^\kappa$, if $b = 0$, or the session key sk_i of session sid, if $b = 1$.

Claim 33. *The probability that $\mathcal{S}$ aborts in $G_{3,0}$ is $\frac{1}{mN^2}$.*

Proof. The probability of choosing the right session, out of m possible values, and the right parties, out of N possibilities is $1/(mN^2)$. □

Game $G_{3,1}$**.** $\mathcal{S}$ simulates the oracles as in $G_{3,0}$, except for $\mathsf{Initiate}$ and $\mathsf{Complete}$:

- When it is queried $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$, $\mathcal{S}$ does the following: If $\mathsf{P}_i = \mathsf{P}_i^*$ and it is the s_i^*-th session of P_i^*, $\mathcal{S}$ computes the key

$$k_i = p_j(r_i + c) + \bar{y}_j(s_j + r_i + c) + 2h_i.$$

  Else, it simulates as in $G_{3,0}$.

Claim 34. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{3,0}$ and $G_{3,1}$ is negligible.*

Proof. Note that $\mathcal{S}$ knows all the static secret keys. So, by setting $k_i = p_j(r_i + c) + \bar{y}_j(s_j + r_i + c) + 2h_i$, we guarantee the correctness of the scheme in the simulation. □

Game $G_{3,2}$. $\mathcal{S}$ simulates the oracles as in $G_{3,1}$, except for $\mathsf{Respond}$:

- When it is queried $\mathsf{Respond}(\Pi, R, \mathsf{P}_j, \mathsf{P}_i, x_i)$, $\mathcal{S}$ does the following: if $\mathsf{P}_j = \mathsf{P}_i^*$ and it is the s_i^*-th session of P_i^*, $\mathcal{S}$ computes

$$k_j = p_i(r_j + d) + \bar{x}_i(s_i + r_j + d) + 2h_j.$$

  Else it simulates $\mathsf{Respond}$ as in $G_{3,1}$.

Claim 35. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{3,1}$ and $G_{3,2}$ is negligible.*

Proof. Again, note that $\mathcal{S}$ knows all the static secret keys. Hence, by setting $k_j = p_i(r_j + d) + \bar{x}_i(s_i + r_j + d) + 2h_j$, we guarantee the correctness of the scheme in the simulation. □

Game $G_{3,3}$. $\mathcal{S}$ simulates the oracles as in $G_{3,2}$, except for:

- It replaces p_i^* (the static public key of P_i^*) by $u_i \leftarrow\!\!\$\, R_q$

Claim 36. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{3,2}$ and $G_{3,3}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. Since the value p_i^* is a HNF-RLWE sample, then it is indistinguishable from a uniformly random value given that the HNF-RLWE assumption holds. □

Game $G_{3,4}$. $\mathcal{S}$ simulates the oracles as in $G_{3,4}$, except for:

- When it is queried $\mathsf{Respond}(\Pi, R, \mathsf{P}_j, \mathsf{P}_i, x_i)$, $\mathcal{S}$ does the following: If $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_j^*-th session of P_j^* and x_i not was outputted by $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*)$, $\mathcal{S}$ chooses $k_j \leftarrow\!\!\$\, R_q$. Else, it simulates $\mathsf{Respond}$ as in $G_{3,3}$.

Claim 37. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{3,3}$ and $G_{3,4}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. Remark that the key k_j is computed as

$$k_j = p_i^*(r_j + d) + \bar{x}_i(s_i + r_j + d) + 2h_j.$$

Remark that the term $p_i^*(r_j + d) + 2h_j$ is a HNF-RLWE sample since $p_i^* \leftarrow\!\!\$\, R_q$, $r_j + d \leftarrow\!\!\$\, \chi_{\sqrt{2}\alpha}$ and h_j is an error term sampled from χ_α.

Therefore, we define a hybrid game $G'_{3,3}$, where the simulator chooses $t_j \leftarrow\!\!\$\, R_q$ and computes $k_j = t_j + \bar{x}_i(s_i + r_j + d)$. Games $G_{3,3}$ and $G'_{3,3}$ are indistinguishable by the HNF-RLWE assumption.

Now, since t_j is uniform, k_j is also uniform in R_q by Lemma 10. Hence, games $G'_{3,3}$ and $G_{3,4}$ are indistinguishable. □

Claim 38. *For any adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ in game $G_{3,4}$ is negligible, given that the HNF-RLWE assumption holds and $0.97n \geq 2\kappa$.*

Proof. Similar to the proof of Claim 24. □

5.2.2 Adversary Type $\mathcal{A}_4$

Lemma 39. *For any adversary $\mathcal{A}$ of **Type** $\mathcal{A}_4$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ is negligible in the ROM, given that $0.97n \geq 2\kappa$, q is a prime such that $q > 203$ and $q > 8\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right)$, and the $\text{HNF-RLWE}_{q,\chi_\alpha}$ is hard.*

The proof of this lemma follows from Claims 40, ..., 45.

Game $G_{4,0}$. Similar to $G_{1,0}$ but now $\mathcal{S}$ chooses $\mathsf{sid} = \left(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*, x_i^*, (y_j^*, w_j^*)\right)$ as the test session where $\mathsf{P}_i^*, \mathsf{P}_j^* \leftarrow_\$ \{\mathsf{P}_1, \ldots, \mathsf{P}_N\}$, $s_i^* \leftarrow_\$ \{1, \ldots, m\}$, x_i^* is outputted by $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*)$ on the s_i^*-th session of P_i^*. $\mathcal{S}$ runs internally $\mathcal{A}$ and simulates the oracles as in $G_{1,0}$ except for Test:

- $\mathsf{Test}(\mathsf{sid})$: Let $\mathsf{sid} = (\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$ be the test session queried by $\mathcal{A}$. If $(\mathsf{P}_i, \mathsf{P}_j) \neq (\mathsf{P}_i^*, \mathsf{P}_j^*)$ or x_i is not outputted in the s_i^* session of P_i^*, $\mathcal{S}$ aborts the execution. Else, it chooses $b \leftarrow_\$ \{0,1\}$ and returns either a random key $\mathsf{sk}_i' \leftarrow_\$ \{0,1\}^\kappa$, if $b = 0$, or the session key sk_i of session sid, if $b = 1$.

Claim 40. *The probability that $\mathcal{S}$ aborts in $G_{4,0}$ is $\frac{1}{mN^2}$.*

Game $G_{4,1}$. $\mathcal{S}$ simulates the oracles as in $G_{4,0}$, except for $\mathsf{Respond}$:

- When it is queried $\mathsf{Respond}(\Pi, R, \mathsf{P}_j, \mathsf{P}_i, x_i)$, $\mathcal{S}$ does the following: if $\mathsf{P}_j = \mathsf{P}_j^*$ and it is the s_j^*-th session of P_j^*, $\mathcal{S}$ computes
$$k_j = p_i(r_j + d) + \bar{x}_i(s_i + r_j + d) + 2h_j.$$
Else it simulates $\mathsf{Respond}$ as in $G_{3,1}$.

Claim 41. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{4,0}$ and $G_{4,1}$ is negligible.*

Proof. The proof is similar to the proof of Claim 35. □

Game $G_{4,2}$. $\mathcal{S}$ simulates the oracles as in $G_{4,1}$, except for $\mathsf{Initiate}$ and $\mathsf{Complete}$:

- When it is queried $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$, $\mathcal{S}$ does the following: If $\mathsf{P}_i = \mathsf{P}_j^*$ and it is the s_j^*-th session of P_j^*, $\mathcal{S}$ computes the key
$$k_i = p_j(r_i + c) + \bar{y}_j(s_j + r_i + c) + 2h_i.$$
Else, it simulates as in $G_{4,1}$.

Claim 42. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{4,1}$ and $G_{4,2}$ is negligible.*

Proof. The proof is similar to the proof of Claim 34. □

Game $G_{4,3}$. $\mathcal{S}$ simulates the oracles as in $G_{4,2}$, except for:

- It replaces p_j^* (the static public key of P_j^*) by $v_j \leftarrow\!\!\$\, R_q$

Claim 43. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{4,2}$ and $G_{4,3}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. The proof is similar to the proof of Claim 36. □

Game $G_{4,4}$. $\mathcal{S}$ simulates the oracles as in $G_{4,4}$, except for:

- When it is queried $\mathsf{Complete}(\Pi, R, \mathsf{P}_j, \mathsf{P}_i, x_i)$, $\mathcal{S}$ does the following: If $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i^* and y_j not was outputted by $\mathsf{Respond}(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i)$, $\mathcal{S}$ chooses $k_i \leftarrow\!\!\$\, R_q$. Else, it simulates $\mathsf{Complete}$ as in $G_{4,3}$.

Claim 44. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{4,3}$ and $G_{4,4}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. The proof is similar to the proof of Claim 37 □

Claim 45. *For any adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ in game $G_{4,4}$ is negligible, given that $0.97n \geq 2\kappa$.*

Proof. Similar to the proof of Claim 24. □

5.2.3 Adversary Type $\mathcal{A}_5$

Lemma 46. *For any adversary $\mathcal{A}$ of **Type** $\mathcal{A}_5$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ is negligible in the ROM, given that $0.97n \geq 2\kappa$, q is a prime such that $q > 203$ and $q > 8\left(16\alpha^2 n^{3/2} + 2\alpha\sqrt{n}\right)$, and the* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

The proof of the lemma follows the sequence of games $G_{5,0}, \ldots, G_{5,3}$.

Game $G_{5,0}$. Similar to $G_{1,0}$ but now $\mathcal{S}$ chooses $\mathsf{sid} = \left(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*, (y_j^*, w_j^*)\right)$ as the test session where $\mathsf{P}_i^*, \mathsf{P}_j^* \leftarrow\!\!\$\, \{\mathsf{P}_1, \ldots, \mathsf{P}_N\}$, $s_i^*, s_j^* \leftarrow\!\!\$\, \{1, \ldots, m\}$, x_i^* is outputted by a query $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i^*, \mathsf{P}_j^*)$ on the s_i^*-th session of P_i^* and y_i^* is outputted by $\mathsf{Respond}(\Pi, R, \mathsf{P}_j^*, \mathsf{P}_i^*, x_i^*)$ on the s_j^*-th session of P_j^*.

Claim 47. *The probability that $\mathcal{S}$ aborts in $G_{5,0}$ is $\frac{1}{m^2N^2}$.*

Game $G_{5,1}$. $\mathcal{S}$ simulates the oracles as in $G_{5,0}$ except for $\mathsf{Complete}$:

- When it is queried on $\mathsf{Initiate}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j)$, $\mathcal{S}$ does the following: If $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i, then S samples $x_i \leftarrow\!\!\$\, R_q$.

  Else, it simulates the oracles as in game $G_{5,0}$.

Claim 48. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{5,0}$ and $G_{5,1}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. The proof is the same as the proof of Claim 21. □

Game $G_{5,2}$. $\mathcal{S}$ simulates the oracles as in $G_{5,1}$, except for Respond:

- When it is queried $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$, $\mathcal{S}$ does the following: if $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_i^*-th session of P_i^* and (y_j, w_j) was not outputted on the s_j^*-th session of P_j^*, $\mathcal{S}$ samples $k_i \leftarrow\!\!\$\, R_q$.

Claim 49. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{5,1}$ and $G_{5,2}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. The proof is the same as the proof of Claim 22. □

Game $G_{5,3}$. $\mathcal{S}$ simulates the oracles as in $G_{5,2}$, except for Respond:

- When it is queried $\mathsf{Complete}(\Pi, I, \mathsf{P}_i, \mathsf{P}_j, x_i, (y_j, w_j))$, $\mathcal{S}$ does the following: if $(\mathsf{P}_i, \mathsf{P}_j) = (\mathsf{P}_i^*, \mathsf{P}_j^*)$ and it is the s_j^*-th session of P_j^* , $\mathcal{S}$ samples $k_j \leftarrow\!\!\$\, R_q$.

Claim 50. *For every adversary $\mathcal{A}$, the probability that $\mathcal{A}$ distinguishes games $G_{5,2}$ and $G_{5,3}$ is negligible, given that* HNF-RLWE$_{q,\chi_\alpha}$ *is hard.*

Proof. The proof is the same as the proof of Claim 23. □

Claim 51. *For any adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}$ in game $G_{5,3}$ is negligible, given that $0.97n \geq 2\kappa$.*

Proof. The proof is the same as the proof of Claim 24. □

6 Efficiency of the AKE scheme and comparison

Communication complexity. The messages exchanged are x_i, carrying $n \log q$ bits of information, and (y_j, w_j) carrying $n + n \log q$ bits of information. Hence, the total number of bits exchange during one execution of the protocol is $n + 2n \log q$.

Computational complexity. First, note that party P_i can perform the multiplication $p_j s_i$ offline and save this value for later use. Hence the scheme requires 6 multiplication in the ring R_q, 3 for each party. One execution of the protocol requires to sample 8 times from discrete Gaussian distributions, 4 for each party.

Proposed parameters. For a security of (at least) 128 bits, the proposed parameters of [27] are $n = 512$, $\alpha = 4.19$ and $q = 120\,833$. These parameters of [27] were estimated based the attacks of [3, 37, 7, 9]. However, we cannot use these parameters in our scheme, since correctness is not guaranteed, by Lemma 16. Hence, we consider $q = 26\,038\,273$, which is the minimum prime that satisfies $q \equiv 1 \mod 2n$ and Lemma 16. We require that $q \equiv 1 \mod 2n$ because of efficiency purposes [4, 27]. Note that, by increasing the value of q, the security can only increase. For a security level of (at least) 256 bits, we propose the parameters $n = 1024$, $\alpha = 2.6$ and $q = 28\,434\,433$. These parameters

were chosen in a similar way as the previous ones. With these parameters, the probability of failure is negligible, according to Lemma 16.

Table 3 presents a comparison with the ZZD+ scheme of the proposed parameters. Table 4 presents a comparison with the ZZD+ scheme of the keys and messages size (expressed in kyloBytes). As we can see, our scheme achieves smaller keys and smaller exchanged messages. As discuss in the introduction, this is due to the fact that our scheme avoids the use of the rejection sampling technique. Since the scheme of ZZD+ requires rejection sampling and several discrete Gaussian distributions (in particular, it is required to use a discrete Gaussian distribution χ_β with $\beta >> \alpha$), the value of q needs to be larger than usual. This results in larger keys and exchanged messages.

	n	α	τ	$\log \beta$	q	Sec. level (bits)
ZZD+ [56]	1024	3.397	12	16.1	2^{45}	80
Ours (I)	512	4.19	–	–	26 038 273	128
Ours (II)	1024	2.6	–	–	28 434 433	256

Table 3: Comparison of parameters with other AKE schemes with implicit authentication.

	Size		
	(Static) pk	init. msg	resp. msg
ZZD+ [56]	5.625 KB	5.625 KB	5.75 KB
Ours (I)	1.577 KB	1.577 KB	1.641 KB
Ours (II)	3.153 KB	3.153 KB	3.281 KB

Table 4: Comparison of size with other AKE schemes with implicit authentication.

7 Conclusion and open problems

We propose an efficient technique that allows for key reuse in lattice-based KE and which we have called pasteurization. We believe that this technique may be of independent interest, as it can be used to sanitize RLWE samples.

We also present a new AKE scheme using the above technique and that surpasses state-of-the-art AKE with implicit authentication in both communication and computational complexity.

Note that our proofs are in the ROM [10]. We leave as future work to present a proof of security for the AKE in the Quantum Random Oracle Model (QROM) [12] as, by now, the lack of proofs techniques makes this problem quite hard. However, we remark that very few constructions are known to be secure in the ROM, but insecure in the QROM [6].[8] Thus, we strongly believe that the

[8] We also remark that the attack of [6] is quite impractical in real-life.

scheme is secure in the QROM since we are not aware of any quantum attack that breaks the scheme, although we present no proof for this.

Acknowledgments

Pedro Branco thanks the support of DP-PMI and FCT (Portugal) through the grant PD/BD/135181/2017.

References

[1] NIST: post-quantum cryptography - call for proposals. `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/call-for-proposals`. Accessed: 2018-12-12.

[2] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 553–572, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[3] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 297–322, Cham, 2017. Springer International Publishing.

[4] Erdem Alkim, Roberto Avanzi, Joppe Bos, Léo Ducas, Antonio de la Piedra, Thomas Pöppelmann, Peter Schwabe, and Douglas Stebila. New hope: Algorithm specifications and supporting documentation. National Institute of Standards and Technology's call for Post-Quantum Standardization (2017), available at https://csrc.nist. gov/projects/post-quantum-cryptography/round-1-submissions, 2017. `https://newhopecrypto.org/resources.shtml`.

[5] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange—a new hope. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 327–343, Austin, TX, 2016. USENIX Association.

[6] A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 474–483, Oct 2014.

[7] Yoshinori Aono, Xavier Boyen, Le Trieu Phong, and Lihua Wang. Key-private proxy re-encryption under lwe. In Goutam Paul and Serge Vau-

denay, editors, *Progress in Cryptology – INDOCRYPT 2013*, pages 1–18, Cham, 2013. Springer International Publishing.

[8] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, pages 595–618, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[9] Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary lwe. In Willy Susilo and Yi Mu, editors, *Information Security and Privacy*, pages 322–337, Cham, 2014. Springer International Publishing.

[10] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, pages 62–73, New York, NY, USA, 1993. ACM.

[11] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, pages 232–249, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[12] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 41–69, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[13] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. Crystals - kyber: A cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 353–367, April 2018.

[14] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy*, pages 553–570, May 2015.

[15] Colin Boyd, Yvonne Cliff, Juan Gonzalez Nieto, and Kenneth G. Paterson. Efficient one-round key exchange in the standard model. In Yi Mu, Willy Susilo, and Jennifer Seberry, editors, *Information Security and Privacy*, pages 69–83, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[16] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 505–524, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[17] Pedro Branco, Jintai Ding, Manuel Goulão, and Paulo Mateus. Universally composable oblivious transfer protocol based on the RLWE assumption.

Cryptology ePrint Archive, Report 2018/1155, 2018. https://eprint.iacr.org/2018/1155.

[18] Christina Brzuska, Marc Fischlin, Bogdan Warinschi, and Stephen C. Williams. Composability of Bellare-Rogaway key exchange protocols. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pages 51–62, New York, NY, USA, 2011. ACM.

[19] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 453–474, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[20] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 523–552, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[21] Tim Dierks and Eric Rescorla. The transport layer security (TLS) protocol version 1.2. Technical report, 2008.

[22] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, Nov 1976.

[23] J. Ding, S. Alsayigh, R. V. Saraswathy, S. Fluhrer, and X. Lin. Leakage of signal function with reused keys in RLWE key exchange. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[24] Jintai Ding, Saed Alsayigh, Jean Lancrenon, Saraswathy RV, and Michael Snook. Provably secure password authenticated key exchange based on RLWE for the post-quantum world. In Helena Handschuh, editor, *Topics in Cryptology – CT-RSA 2017*, pages 183–204, Cham, 2017. Springer International Publishing.

[25] Jintai Ding, Scott Fluhrer, and Saraswathy R.V. Complete attack on RLWE key exchange with reused keys, without signal leakage. In Willy Susilo and Guomin Yang, editors, *Information Security and Privacy*, pages 467–486, Cham, 2018. Springer International Publishing.

[26] Jintai Ding, Saraswathy RV, Saed Alsayigh, and Crystal Clough. How to validate the secret of a ring learning with errors (RLWE) key. Cryptology ePrint Archive, Report 2018/081, 2018. https://eprint.iacr.org/2018/081.

[27] Jintai Ding, Tsuyoshi Takagi, Xinwei Gao, and Yuntao Wang. Ding key exchange. National Institute of Standards and Technology's call for Post-Quantum Standardization (2017), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions, 2017.

[28] Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive, Report 2012/688, 2012. https://eprint.iacr.org/2012/688.

[29] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, and Daniel Masny. New constructions of identity-based and key-dependent message secure encryption schemes. In Michel Abdalla and Ricardo Dahab, editors, *Public-Key Cryptography – PKC 2018*, pages 3–31, Cham, 2018. Springer International Publishing.

[30] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 1857–1874, New York, NY, USA, 2017. ACM.

[31] Scott Fluhrer. Cryptanalysis of ring-LWE based key exchange with key share reuse. Cryptology ePrint Archive, Report 2016/085, 2016. https://eprint.iacr.org/2016/085.

[32] Alan Freier, Philip Karlton, and Paul Kocher. The secure sockets layer (SSL) protocol version 3.0. Technical report, 2011.

[33] Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. Strongly secure authenticated key exchange from factoring, codes, and lattices. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography – PKC 2012*, pages 467–484, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[34] Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 83–94, New York, NY, USA, 2013. ACM.

[35] X. Gao, J. Ding, L. Li, and J. Liu. Practical randomized rlwe-based key exchange against signal leakage attack. *IEEE Transactions on Computers*, 67(11):1584–1593, Nov 2018.

[36] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 197–206. ACM, 2008.

[37] Gottfried Herold, Elena Kirshanova, and Alexander May. On the asymptotic complexity of solving lwe. *Designs, Codes and Cryptography*, 86(1):55–83, Jan 2018.

[38] ISO. ISO/IEC 11770-3:2008, information technology - security techniques - key management - part 3: Mechanisms using asymmetric techniques, 2009.

[39] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, pages 19–34, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[40] Jonathan Katz and Vinod Vaikuntanathan. Smooth projective hashing and password-based authenticated key exchange from lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, pages 636–652, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[41] Hugo Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, pages 546–566, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[42] Brian LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger security of authenticated key exchange. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *Provable Security*, pages 1–16, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[43] Laurie Law, Alfred Menezes, Minghua Qu, Jerry Solinas, and Scott Vanstone. An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography*, 28(2):119–134, Mar 2003.

[44] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, pages 598–616, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[45] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 738–755, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[46] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 1–23, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[47] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 35–54, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[48] Tsutomu Matsumoto, Youichi Takashima, and Hideki Imai. On seeking smart public-key-distribution systems. *IEICE TRANSACTIONS (1976-1990)*, 69(2):99–106, 1986.

[49] A. Menezes, M. Qu, and S. Vanstone. Some new key agreement protocols providing mutual implicit authentication. In *Selected Areas in Cryptography*, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[50] D. Micciancio and O. Regev. Worst case to average case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[51] Chris Peikert. Lattice cryptography for the internet. In Michele Mosca, editor, *Post-Quantum Cryptography*, pages 197–219, Cham, 2014. Springer International Publishing.

[52] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[53] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, Jan 1991.

[54] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[55] Andrew Chi-Chih Yao and Yunlei Zhao. OAKE: A new family of implicitly authenticated Diffie-Hellman protocols. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, CCS '13, pages 1113–1128, New York, NY, USA, 2013. ACM.

[56] Jiang Zhang, Zhenfeng Zhang, Jintai Ding, Michael Snook, and Özgür Dagdelen. Authenticated key exchange from ideal lattices. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 719–751, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.