Hindawi
Security and Communication Networks
Volume 2022, Article ID 8299232, 7 pages
https://doi.org/10.1155/2022/8299232

*Research Article*

# Module-LWE-Based Key Exchange Protocol Using Error Reconciliation Mechanism

**Wenjuan Jia**,[1] **Guanhao Xue**,[1] **Baocang Wang**,[2] **and Yupu Hu**[2]

[1]*School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi 710071, China*
[2]*The State Key Laboratory of Integrated Services Network, Xidian University, Xi'an, Shaanxi 710071, China*

Correspondence should be addressed to Wenjuan Jia; 18368910175@163.com

Received 3 October 2021; Accepted 23 December 2021; Published 1 February 2022

Academic Editor: Rongmao Chen



Lattice-based key exchange protocols have attracted tremendous attention for its post-quantum security. In this work, we construct a Module-LWE-based key exchange protocol using Peikert's error reconciliation mechanism. Compared with Kyber.KE, our key exchange protocol reduces the total communication cost by 96-byte, i.e., 3.2% ~ 6.1%, under the different parameter sets, and without reducing the post-quantum security levels. Moreover, our key exchange protocol slightly reduces the probability of session key agreement failure and the time consumed by modular multiplication of numbers and ring elements by approximately 30%. Thus, the key exchange protocol in this paper is more suitable for the lightweight communication systems.

## 1. Introduction

Key exchange protocol, which enables secure communications over an untrusted network by deriving and distributing shared keys between two or more parties, is one of the most fundamental cryptographic primitives and is widely applied in modern Internet protocols such as TLS [1] and SSL [2]. However, Shor [3] discovered an efficient quantum solving algorithm for the integer factorization and discrete logarithm problems in 1994, which would render number-theoretic cryptosystems insecure if large-scale quantum computers become available. With the rapid developments of quantum technology and quantum computer, we are getting closer to the quantum crisis of current public key cryptosystems. Therefore, it is urgent to propose post-quantum cryptographic schemes, such as public key encryptions (PKE), signatures, and key exchanges, that can resist quantum computer attacks. Lattice-based cryptography is one of the main directions in this field and has become the most promising post-quantum cryptography (PQC) candidate for standardization.

Lattice-based key exchange protocols are generally constructed using the *learning with errors* (LWE) problem and its variants. In 2005, Regev [4, 5] introduced the LWE problem and showed that solving the LWE problem with a Gaussian error distribution is at least as hard as *quantumly* solving the approximates shortest vector problem (GapSVP) and shortest independent vector problem (SIVP) on lattices in the worst case. Later, Peikert [6] gave a *classical* reduction from the approximate GapSVP (and its variants) to the search version of LWE, but with somewhat worse parameters.

Although LWE provides provably secure cryptosystems, most LWE-based schemes are inefficient which motivates the research around more efficient LWE variants. These variants improve the asymptotic and practical efficiency by considering the ring of integers of a number field [7, 8], a ring of polynomials [9], or a module over a number field [10, 11]. Lyubashevsky et al. [7] introduced the *ring learning with errors* (Ring-LWE) problem and proved its hardness is related to the hardness of the lattice problems based on ideal lattices. Later, the *module learning with errors* (Module-LWE) problem was introduced by Langlois and Stele [11] in 2015, and Module-LWE comes with the hardness guarantees given by lattice problems based on module lattices. Since the algebraic structures of module lattices are more complicated than ideal lattices, Module-LWE might be able to provide a better level of security than Ring-LWE, while still providing

performance advantages over LWE. In this paper, we focus on the key exchange protocols based on Module-LWE, as Module-LWE provides a nice security-efficiency trade-off by bridging LWE and Ring-LWE.

Lattice-based key exchange protocols generally include two types of protocols constructed using error reconciliation mechanism or key encapsulation mechanism (KEM). Most LWE-based (and its variants) key exchange protocols are constructed using error reconciliation mechanism, such as Ding's key exchange [12], BCNS [13], NewHope [14], Frodo [15], etc. Ding et al. [12] proposed an LWE-based Diffie-Hellman-like key exchange protocol and gave its security proof in 2012. Later, for Peikert's tweaked version [16] of Ding's key exchange protocol [12], Bos et al. [13] presented a concrete instantiation whose security is based on Ring-LWE problem and gave an implementation integrated into OpenSSL, with the affirmed goal of providing post-quantum security for TLS. Unfortunately, the performance of BCNS seems quite disappointing. In 2015, Alkim et al. [14] improved and generalized Peikert's error reconciliation mechanism [16] using an analog error-correction approach and presented an unauthenticated key exchange protocol that solved the performance and security issues in BCNS [13]. Subsequently, Bos et al. [15] proposed the Frodo protocol based on similar ideas to the LWE-based protocol in [12], but as in the Ring-LWE-based key exchange protocols BCNS [13] and NewHope [14], Bos et al. incorporated and extended Peikert's error reconciliation mechanism [16] and further modified the protocol to save bandwidth.

Key exchange protocols constructed using KEM include NewHope-simple [17] and Kyber.KE [18], etc. Most of the exiting lattice-based key exchange protocols are constructed using KEM for its simplicity and modularity, although it will cause more communication cost. Alkim et al. [17] introduced NewHope-simple in 2016, which is a variant of the NewHope [14]. The main advantage of NewHope-simple over NewHope is simplicity; in particular, NewHope-simple avoids the error reconciliation mechanism. In 2018, Bos et al. [18] presented Kyber.KE that was constructed using a IND-CCA-secure KEM, and the security of Kyber.KE is based on the hardness of Module-LWE in the classical and quantum random oracle models. Recently, Xue et al. [19] presented an authentication key exchange (AKE) protocol following a generic construction with a KEM and a signature scheme in 2021. Compared with the Kyber.AKE [18], Xue's AKE protocol reduced the communication overhead under the same post-quantum security levels.

In this work, we propose a key exchange protocol constructed using error reconciliation mechanism, its security based on the hardness of Module-LWE problem. Compared with Kyber.KE, our key exchange protocol reduces the total communication cost by 96 bytes, i.e., 3.2% ~ 6.1%, under the same post-quantum security levels and different parameter sets, and the time consumed by modular multiplication of ring elements and numbers by approximately 30%. Secondly, the number of the most time-consuming operations (such as discrete binomial sampling and modular multiplication of ring elements) is reduced in our key exchange protocol since the reencryption is not used. Thus, our key exchange protocol is more suitable for lightweight communication protocol, such as Internet of Vehicles environment and smart home terminals. Thirdly, our protocol slightly reduces the probability of the agreement failure for the compression algorithm used is less than that in Kyber.KE. Moreover, the key exchange protocol proposed in this paper is relatively symmetric: the process of the protocol is symmetric, and the computational as well as communication costs of two parties are nearly the same. Finally, our key exchange protocol inherits the parameter sets of Kyber.KE, which lead to the same post-quantum security strength, and the computational efficiency is almost the same as Kyber.KE according to the performance analysis.

Section 2 gives the necessary preliminaries and definitions. Then, Section 3 describes our key exchange protocol, analyzes the correctness and security, and gives parameter sets and its performance. Finally, Section 4 makes a conclusion of our work.

## 2. Preliminaries

Denote the security parameter by $\lambda$, and the negligible function by $\mathrm{negl}(\lambda) \in \lambda^{-\omega(1)}$. Let $q$ be a prime, $n$ be a power of two, positive integer $k$ be the rank of Module-LWE, and $q \equiv 1 \bmod 2n$. We write $\mathbb{Z}$ for the set of integers, $\mathbb{Q}$ for the set of rational numbers, and $\mathbb{R}$ for the set of reals. Let $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$, $R := \mathbb{Z}[x]/(x^n+1)$, and $R_q = R/qR := \mathbb{Z}_q[x]/(x^n+1)$. We use bold lowercase letters $\mathbf{a}$ for column vectors, bold uppercase letters $\mathbf{A}$ for matrices, and $(\cdot)^T$ for the transpose of vectors/matrices. Denote probability distributions by calligraphic letters $\mathcal{S}$, and discrete set by uppercase letters $S$. We write $x \leftarrow \mathcal{S}$ to denote sampling $x$ from the distribution $\mathcal{S}$, and $x \leftarrow S$ to denote that $x$ is chosen uniformly at random from a set $S$. For an even (resp., odd) positive integer $p$, we define $\widehat{x} = x \bmod^{\pm} p$ to be unique element $\widehat{x}$ in the range $-p/2 < \widehat{x} \le p/2$ (resp., $-(p-1)/2 < \widehat{x} \le (p-1)/2$ such that $\widehat{x} = x \bmod p$, and $\widehat{x} = x \bmod^{+} p$ to be unique element $\widehat{x}$ in the range $0 \le \widehat{x} < p$ such that $\widehat{x} = x \bmod p$. Assume that $\mathrm{Sample}(\cdot)$ is an *extendable output function*, that is, a function on bit strings in which the output can be extended to any desired length. Let $y \sim \mathcal{S} = \mathrm{Sample}(x)$ (resp., $y \sim S = \mathrm{Sample}(x)$); i.e., if function $\mathrm{Sample}(\cdot)$ takes $x$ as input, then its output is $y$ according to distribution $\mathcal{S}$ (resp., uniformly over a set $S$).

*2.1. Module-LWE Problem and Compression Algorithm.* The Module-LWE problem was first defined by Brakerski et al. [10] and studied in detail by Langlois and Stehlé [11]. Let $K$ be a number field of degree $n$, and $R$ be the ring of integers of $K$. Let $K_{\mathbb{R}} := K \otimes_{\mathbb{Q}} \mathbb{R}$, $\mathbb{T}_R := K_{\mathbb{R}}/R$, and $\mathbb{T}_{qR} := K_{\mathbb{R}}/qR$. We refer the reader to [20] and [11, 21–24] for the thorough introduction to algebraic number theory. Let $\chi$ be a distribution on $K_{\mathbb{R}}$.

The search Module-LWE problem $\mathrm{MLWE}_{q,k,m,n,\chi}$ is to find $\mathbf{s} \in (R_q)^k$ given $(\mathbf{A}, b := As + e)$, where $\mathbf{A} \leftarrow (R_q)^{m \times k}$, $\mathbf{s} \leftarrow (R_q)^k$, and $\mathbf{e} \leftarrow \chi^m$, whereas the decision variant of the Module-LWE problem $\mathrm{dMLWE}_{q,k,m,n,\chi}$ asks to distinguish

the distribution $(\mathbf{A}, b{:}\ = As + e)$ from uniform distribution $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow (R_q)^{m\times k}$, $\mathbf{s} \leftarrow (R_q)^k$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{u} \leftarrow (\mathbb{T}_{qR})^m$.

It can be shown that the *normal form* of the above problems where the secret distribution is a discretized version of the error distribution is no easier than the case where the secret is chosen uniformly at random. When the error distribution $\chi$ is a Gaussian distribution of parameter $\eta > 0$ or a centered binomial distribution of parameter $\eta > 0$, we write $\mathrm{MLWE}_{q,k,m,n,\eta}$ ($\mathrm{dMLWE}_{q,k,m,n,\eta}$). We denote $\mathrm{Adv}^{\mathrm{MLWE}}_{q,k,m,n,\chi}[\mathscr{A}]$ by the advantage of the adversary $\mathscr{A}$ in solving the search Module-LWE problem, and $\mathrm{Adv}^{\mathrm{dMLWE}}_{q,k,m,n,\chi}[\mathscr{D}]$ by the advantage of the distinguisher $\mathscr{D}$ in distinguishing between the two distributions of the decision Module-LWE problem. More precisely, the Module-LWE problem is defined as follows.

*Definition 1.* Let $q \geq 2$ be a modulus, $m = \mathrm{poly}(\lambda)$ be the number of samples, $k > 0$ be the rank of Module-LWE, and $n = \mathrm{poly}(\lambda)$ be the degree of modular polynomial. Let $\chi$ be a distribution on $K_{\mathbb{R}}$.

We say that the search problem $\mathrm{MLWE}_{q,k,m,n,\chi}$ is hard, if it holds for every PPT adversary $\mathscr{A}$ that

$$\mathrm{Adv}^{\mathrm{MLWE}}_{q,k,m,n,\chi}[\mathscr{A}] = \Pr\left[\mathscr{A}\left(1^\lambda, A, b{:}\ = As + e\right) = \mathbf{s}\right] \leq \mathrm{negl}(\lambda), \tag{1}$$

where $\mathbf{A} \leftarrow (R_q)^{m\times k}$, $\mathbf{s} \leftarrow (R_q)^k$, and $\mathbf{e} \leftarrow \chi^m$.

We say that the decision problem $\mathrm{dMLWE}_{q,k,m,n,\chi}$ is hard, if it holds for every PPT distinguisher $\mathscr{D}$ that

$$\mathrm{Adv}^{\mathrm{dMLWE}}_{q,k,m,n,\chi}[\mathscr{D}] = \left|\Pr\left[\mathscr{D}\left(1^\lambda, A, b{:}\ = As + e\right) = 1\right] - \Pr\left[\mathscr{D}\left(1^\lambda, \mathbf{A}, \mathbf{u}\right) = 1\right]\right| \leq \mathrm{negl}(\lambda), \tag{2}$$

where $\mathbf{A} \leftarrow (R_q)^{m\times k}$, $\mathbf{s} \leftarrow (R_q)^k$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{u} \leftarrow (\mathbb{T}_{qR})^m$.

Let integer $\eta > 0$; the *central binomial distribution* $\beta_\eta$ is defined as follows: randomly choosing samples $(x_i, y_i)_{i=1}^{\eta}$ from $\{0,1\}^\eta$, and output $\sum_{i=1}^{\eta}(x_i - y_i)$. For $v \in R$, $v \leftarrow \beta_\eta$ means that each of its coefficients is sampling from $\beta_\eta$ independently. Next, we review the compression algorithm in [18].

*Definition 2.* Let $0 < d < \lceil \log(q) \rceil$ be an integer and $q > 0$ be a modulus. The compression algorithm consists of two functions: $\mathrm{Compress}_q(\cdot, d)$ and $\mathrm{Decompress}_q(\cdot, d)$. These two functions are defined as follows:

$$\begin{aligned}\mathrm{Compress}_q(x,d){:}\ &= \left\lceil \frac{2^d}{q}\cdot x \right\rfloor \mathrm{mod}^+ 2^d \text{ for } x \in \mathbb{Z}_q,\\ \mathrm{Decompress}_q(x,d){:}\ &= \left\lceil \frac{q}{2^d}\cdot x \right\rfloor \text{ for } x \in \mathbb{Z}_{2^d}.\end{aligned} \tag{3}$$

If $\mathrm{Compress}_q(\cdot, d)$ or $\mathrm{Decompress}_q(\cdot, d)$ is used with vector $\mathbf{v} \in (R_q)^k$, then the function is applied to each coefficient individually.

*2.2. Error Reconciliation Mechanism.* When constructing key exchange protocol using LWE problem and its variants, a serious matter is that there usually are errors in the protocol, which leads to similar values instead of the same values. These errors are significant to the post-quantum security and should be handled since the key exchange protocol requires communication parties get common session key. The *error reconciliation mechanism*, first introduced by Ding et al. [12], is the key technique to deal with errors. It mainly include Ding's error reconciliation mechanism [12], Peikert's error reconciliation mechanism (and its multibit variant) [16], and $\widetilde{D}_4$ lattice decoding [14] so far. Peikert's error reconciliation mechanism is widely used because of its simplicity and efficiency, such as BCNS [13] and Frodo [15], and the detailed process of reconciliation mechanism and its correctness is described in [16]. Next, we recall Peikert's reconciliation mechanism.

Defining $\lceil x \rfloor{:}\ = \lfloor x + (1/2) \rfloor \in \mathbb{Z}$ for $x \in \mathbb{R}$, for an integer $p$ that divides $q$ (typically $p = 2$), the *modular rounding function* $\lceil \cdot \rfloor_{q,p}{:}\ \mathbb{Z}_q \longrightarrow \mathbb{Z}_p$ is defined by $\lceil x \rfloor_{q,p}{:}\ = \lceil x \cdot (p/q) \rfloor \mathrm{mod} p$. Defining $I_0{:}\ = \{0, 1, \ldots, \lceil (q/4) \rfloor - 1\}$, $I_1{:}\ = \{-\lfloor q/4 \rfloor, \ldots, -2, -1\} \mathrm{mod} q$, and $E = [(-q/8), t(q/8)) \cap \mathbb{Z}$. The *cross-rounding function* $\langle\cdot\rangle_{q,2}{:}\ \mathbb{Z}_q \longrightarrow \mathbb{Z}_2$ is defined by $\langle x \rangle_{q,2}{:}\ = \lfloor (4/q) \cdot x \rfloor \mathrm{mod} 2$, and the *reconciliation function* $\mathrm{Rec}(\cdot,\cdot){:}\ \mathbb{Z}_q \times \mathbb{Z}_2 \longrightarrow \mathbb{Z}_2$ is defined by $\mathrm{Rec}(y, \sigma){:}\ = 0$ if $y \in I_\sigma + E \mod q$; otherwise $\mathrm{Rec}(y, \sigma){:}\ = 1$, where $\sigma \in \mathbb{Z}_2$.

Both modular rounding and cross-rounding functions are extended to polynomials $f \in R_q$ coefficientwise. Lemma 1 shows that the modular rounding $\lceil x \rfloor_{q,2}$ of a uniform random element $x \in \mathbb{Z}_q$ is uniform random in $\mathbb{Z}_2$ given cross-rounding $\langle x \rangle_{q,2}$ of $x$; i.e., $\langle x \rangle_{q,2}$ hides $\lceil x \rfloor_{q,2}$. Lemma 2 shows that one can recover $\lceil x \rfloor_{q,2}$ from an element $y \in \mathbb{Z}_q$ close to an element $x \in \mathbb{Z}_q$, given only $y$ and the cross-rounding $\langle x \rangle_{q,2}$.

**Lemma 1** (see [16], Claim 3.1). *For even $q$, if $x \in \mathbb{Z}_q$ is uniformly random, then $\lceil x \rfloor_{q,2}$ is uniformly random given $\langle x \rangle_{q,2}$.*

**Lemma 2** (see [16], Claim 3.2). *For even $q$, if $y = x + e \mathrm{mod} q$ for some $x \in \mathbb{Z}_q$ and $e \in E$, then $Rec(y, \langle x \rangle_{q,2}) = \lceil x \rfloor_{q,2}$.*

When modulus $q$ is odd, it is necessary to work in $\mathbb{Z}_{2q}$ rather than $\mathbb{Z}_q$ to avoid bias in the derived bits. Since we use odd $q$ in this paper, we need to introduce the *randomized doubling function* from [16]. The randomized doubling function $\mathrm{dbl}(\cdot){:}\ \mathbb{Z}_q \longrightarrow \mathbb{Z}_{2q}$ is defined by $dbl(x){:}\ = 2x - \overline{e}$,

where $\overline{e}$ is sampled from $\{-1, 0, 1\}$ with probabilities $\Pr[\overline{e} = -1] = \Pr[\overline{e} = 1] = (1/4)$ and $\Pr[\overline{e} = 0] = (1/2)$. The randomized doubling function is extended to polynomials $f \in R_q$ by applying it to each of $f$ 's coefficients.

Lemma 3 shows that, for a uniform random element $x \in \mathbb{Z}_q$, the modular rounding $\lceil \mathrm{dbl}(x) \rceil_{2q,2}$ of $\mathrm{dbl}(x) \in \mathbb{Z}_{2q}$ is uniform random in $\mathbb{Z}_2$ given cross-rounding $\langle \mathrm{dbl}(x) \rangle_{2q,2}$; i.e., $\langle \mathrm{dbl}(x) \rangle_{2q,2}$ hides $\langle \mathrm{dbl}(x) \rangle_{2q,2}$. Moreover, if $y, x \in \mathbb{Z}_q$ are close, then so are $2y, \widehat{x} := \mathrm{dbl}(x) \in \mathbb{Z}_{2q}$; i.e., if $y = x + e \bmod q$ for some $e \in E$, then we have $2y = \overline{x} + (2e + \overline{e}) \bmod 2q$. Thus, one can recover $\lceil \overline{x} \rceil_{2q,2}$ of a random element $x \in \mathbb{Z}_q$ from an element $y \in \mathbb{Z}_q$ close to $x$ and the cross-rounding $\langle \overline{x} \rangle_{2q,2}$, as described by Lemma 4.

**Lemma 3** (see [16], Claim 3.3). *For odd $q$, if $x \in \mathbb{Z}_q$ is uniformly random and $\overline{x} := dbl(x) \in \mathbb{Z}_{2q}$, then $\lceil \overline{x} \rceil_{2q,2}$ is uniformly random given $\langle \overline{x} \rangle_{2q,2}$.*

**Lemma 4** (see [16], Section 3.2). *For odd $q$, let $\overline{x} := dbl(x) \in \mathbb{Z}_{2q}$; if $y = x + e \bmod q$ for some $x \in \mathbb{Z}_q$ and $e \in E$, then* $\mathrm{Rec}(2y, \langle \overline{x} \rangle_{2q,2}) = \lceil \overline{x} \rceil_{2q,2}$.

## 3. Key Exchange Protocol

In this section, we propose a Module-LWE-based unauthenticated key exchange protocol using Peikert's error reconciliation mechanism, which is a variant of Kyber.KE [18]. We first describe the concrete process of the key exchange protocol and then prove its correctness and security. Finally, we give the parameter sets and analyze the performance of our key exchange protocol, including communication cost and computation overhead.

*3.1. Key Exchange Protocol Using Peikert's Error Reconciliation Mechanism.* We present a Module-LWE-based key exchange protocol using Peikert's error reconciliation mechanism, instead of using IND-CCA-secure KEM as in Krber.KE [18]. In particular, Alice (initiator) sends $(\mathbf{b}, \rho)$ to Bob (responder) in both our key exchange protocol and Kyber.KE, where $\mathbf{b}$ is the output of the compression function. However, Bob sends $(\mathbf{u}, c)$ to Alice in the second round of our key exchange protocol, where $\mathbf{u}$ is the output of the compression function and $c$ is the output of the cross-rounding function. But Bob sends $(\mathbf{u}, v, d)$ to Alice in the second round of Kyber.KE, where both $\mathbf{u}$ and $v$ are the output of the compression function and $d$ is a 256-bit random bit string. The specific description of the protocol is shown in Figure 1.

Compared with Kyber.KE [18], our key exchange protocol has the following differences.

*3.1.1. Our Key Exchange Protocol Is Relatively Symmetric and Reduces the Communication Cost in the Second Round.* The Kyber.KE is asymmetric: Alice generates key pair and sends public key to the Bob, then Bob encrypts random session key with public key and sends the ciphertext back to Alice, and finally Alice decrypts the received ciphertext to get the session key. It is known from [18] that the communication costs in the first and second rounds of Kyber.KE are not equal. However, the communication costs of both rounds are equal in our key exchange protocol, and we reduce the communication cost in the second round. See Section 3.4 for detailed analysis.

*3.1.2. Our Key Exchange Protocol Slightly Reduces the Probability of Session Key Agreement Failure.* The Kyber.KE always compresses public key and ciphertext using compression algorithm; this is done not only to save communication traffic but also to ensure correctness. Generally, the least significant bits are discarded and the other bits are retained using the compression function. Thus, the probability of the session key agreement failure can be effectively reduced without using the compression algorithm or reducing the number of times of the compression algorithm is used. In Kyber.KE, the compression algorithm will add an extra error term on sent messages, which means the encoded messages are not uniformly at random and then may leak some information. However, the ideal situation (no compression algorithm is used) and the real situation are indistinguishable under certain parameter sets according to the analysis of [18]. Compared with Kyber.KE, our key exchange protocol reduces the number of the compression functions used by 4 times and the decompression functions used by 2 times; thus it can slightly reduce the probability of session key agreement failure of the key exchange protocol. If the compression algorithm is not used, it will not affect the correctness of the protocol but increase the additional communication traffic. Therefore, we still use the compression algorithm and prove that it has no effect on the correctness and security of the protocol in Sections 3.2 and 3.3.

*3.2. Correctness.* This section gives the correctness proof of our Module-LWE-based key exchange protocol. According to Section 2.2, when $q$ is odd, there will be an additional randomized doubling function $\mathrm{dbl}(\cdot)$ in Peikert's error reconciliation mechanism, and it maps $x \in \mathbb{Z}_q$ to $\overline{x} := \mathrm{dbl}(x) \in \mathbb{Z}_{2q}$.

Assume that the tiny error between $\mathbf{b} \in R_q^k$ and $\mathbf{b}'$ after using compression algorithm is $\mathbf{c}_b$; i.e.,

$$\begin{aligned} \mathbf{b}' &= \mathrm{Decompress}_q\left(\mathrm{Compress}_q(\mathbf{b}, d_t), d_t\right) \\ &= \mathbf{A}\mathbf{s} + \mathbf{e} + \mathbf{c}_b. \end{aligned} \tag{4}$$

Similarly, suppose that $\mathbf{u}' = \mathrm{Decompress}_q(\mathrm{Compress}_q(\mathbf{u}, d_u), d_u) = \mathbf{A}^T\mathbf{s}' + \mathbf{e}_1 + \mathbf{c}_u$ for $\mathbf{u} \in R_q^k$.

Thus the difference between $v'$ and $\overline{v}$ is

$$\begin{aligned} \overline{v} - v' &= 2 \cdot \left(\mathbf{b}'^T\mathbf{s}' + e_2\right) - \overline{e} - 2 \cdot \mathbf{s}^T\mathbf{u}' \\ &= 2 \cdot \left(\mathbf{e}^T\mathbf{s}' + \mathbf{c}_b^T\mathbf{s}' + e_2 - \mathbf{s}^T\mathbf{e}_1 - \mathbf{s}^T\mathbf{c}_u\right) - \overline{e}. \end{aligned} \tag{5}$$

Peikert's error reconciliation mechanism shows that the error tolerance range is $\lfloor q/2 \rfloor$ when $q$ is odd. Therefore, the output of reconciliation function is $\mathrm{Rec}(v', c) = \lceil \overline{v} \rceil_{2q,2}$ if the

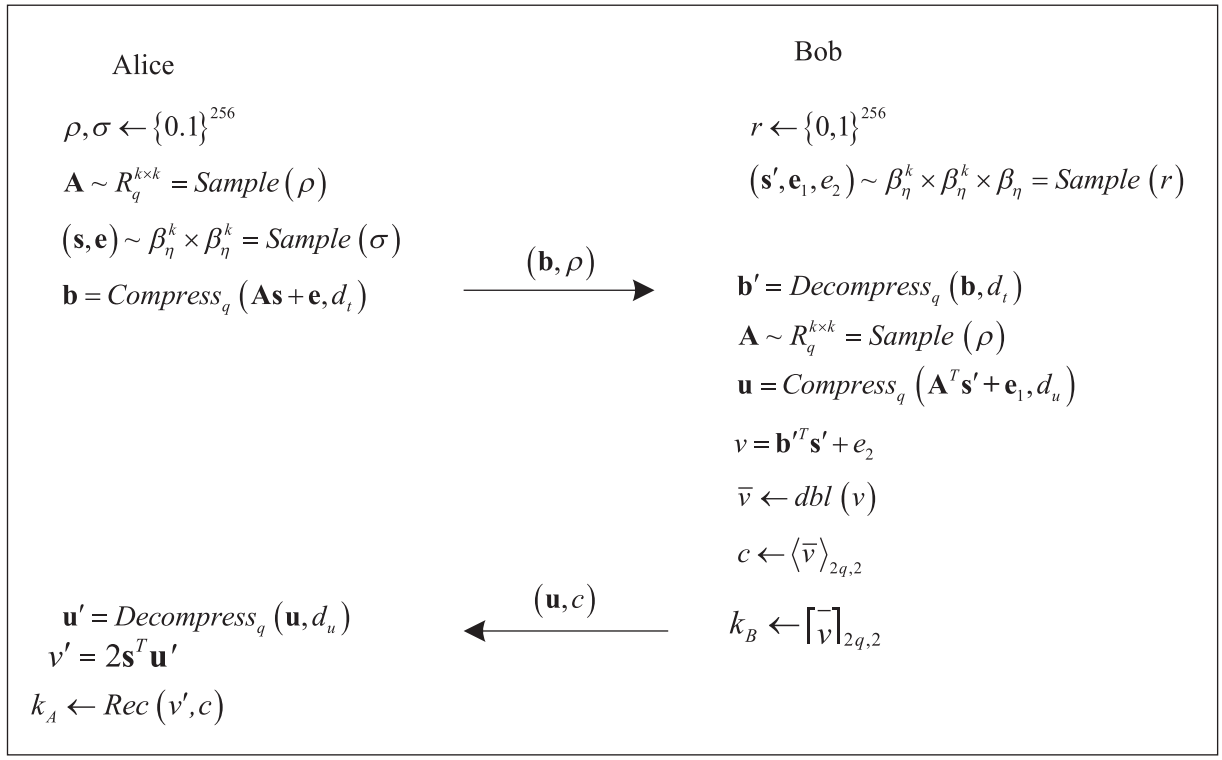


Figure 1: Key exchange protocol using Peikert's error reconciliation mechanism.

difference between $v'$ and $\bar{v}$ satisfies $|\bar{v} - v'| \le \lfloor q/2 \rfloor$. Then the inequation above turns to

$$2 \cdot \left(\mathbf{e}^T\mathbf{s}' + \mathbf{c}_b^T\mathbf{s}' + e_2 - \mathbf{s}^T\mathbf{e}_1 - \mathbf{s}^T\mathbf{c}_u\right) - \bar{e} \le \lfloor \frac{q}{2} \rfloor, \tag{6}$$

i.e.,

$$\mathbf{e}^T\mathbf{s}' + \mathbf{c}_b^T\mathbf{s}' + e_2 - \mathbf{s}^T\mathbf{e}_1 - \mathbf{s}^T\mathbf{c}_u \le \lceil \frac{q}{4} \rfloor. \tag{7}$$

Compared with the correctness proof in Kyber.KE, inequation (1) is almost the same. Thus, we have that the probability that inequation (1) holds is no less than $1 - 2^{-128}$ by choosing appropriate parameter sets, which means that the probability of session key agreement failure is less than $2^{-128}$.

Note that there is a slight difference between inequation (1) and the inequation in Kyber.KE [18], because $v$ is not compressed in our key exchange protocol; i.e., the error term $c_v$ of $v$ is missed in inequation (1). Even though the norm of $c_v$ is relatively small, it increases the probability of session key agreement failure in Kyber.KE under the same parameter sets. In other words, the probability of session key agreement failure in our key exchange protocol is smaller than that in Kyber.KE.

*3.3. Security Proof.* This section gives the security proof of our key exchange protocol by designing a sequence of games. The Module-LWE-based key exchange protocol described in Figure 1 is constructed using Peikert's error reconciliation mechanism; its security relies on the hardness of Module-LWE problem. One can prove that the generated session key is undistinguishable from equal-length random bit string.

**Theorem 1.** *Let $q$ be an odd prime, $n, k$ be public parameters, and $\eta$ be the parameter of binomial distribution. Then the key exchange protocol described in Figure 1 is secure, provided that the decision Module-LWE problem $dMLWE_{q,k,k+1,n,\eta}$ is hard. More precisely, if $\mathscr{D}$ is an distinguisher for $dMLWE_{q,k,k+1,n,\eta}$, then*

$$\mathrm{Adv}^{\mathrm{KE}}_{q,k,k,n,\eta}(\mathscr{A}) \le 2 \cdot \mathrm{Adv}^{\mathrm{dMLWE}}_{q,k,k+1,n,\eta}(\mathscr{D}), \tag{8}$$

*where $\mathscr{A}$ is an adversary for the key exchange protocol described in Figure 1.*

Proof. Let $b^*$ be the bit guessed by adversary, and $\mathscr{A}$ an adversary for the key exchange protocol described in Figure 1. Consider the following sequence of games. **Game 0**. This is the original game, where the messages are honestly generated according to the description in Figure 1. Our goal is to bound $\mathrm{Adv}^{\mathrm{KE}}_{q,k,k,n,\eta}(\mathscr{A}) * = |\Pr[b = b^* \mathrm{inGame0}] - (1/2)|$. Note that, in Game 0, the Module-LWE samples are $(\mathbf{A}, \mathbf{b}_0 := \mathbf{As} + \mathbf{e})$, $(\mathbf{A}^T, \mathbf{u}_0 := \mathbf{A}^T\mathbf{s}' + \mathbf{e}_1)$, and $(\mathbf{b}'^T, v := \mathbf{b}'^T\mathbf{s}' + e_2)$.

**Game 1**. In this game, assume that $\mathbf{b}_0$ is chosen uniformly at random from $R_q^k$; i.e., $(\mathbf{A}, \mathbf{b}_0)$ is chosen uniformly at random from $R_q^{k\times k} \times R_q^k$ instead of a Module-LWE sample. In Game 1, the Module-LWE samples are $(\mathbf{A}^T, \mathbf{u}_0 := \mathbf{A}^T\mathbf{s}' + \mathbf{e}_1)$ and $(\mathbf{b}'^T, v := \mathbf{b}'^T\mathbf{s}' + e_2)$. By the assumption that the decision Module-LWE problem $\mathrm{dMLWE}_{q,k,k+1,n,\eta}$ is hard, we know that Game 0 and Game 1 are computationally indistinguishable. In other words, there exists a Module-LWE

Table 1: Parameter sets and performance of our key exchange protocol.

| Parameter sets | $n$ | $k$ | $q$ | $\eta$ | $(d_t, d_u)$ | pq sec | Alice $\longrightarrow$ Bob (bytes) | Bob $\longrightarrow$ Alice (bytes) |
|---|---|---|---|---|---|---|---|---|
| Light | 256 | 2 | 7681 | 5 | (11, 11) | 102 | 736 | 736 |
| Default | 256 | 3 | 7681 | 4 | (11, 11) | 161 | 1088 | 1088 |
| Paranoid | 256 | 4 | 7681 | 3 | (11, 11) | 218 | 1440 | 1440 |

distinguisher $\mathscr{D}$ with the same running times as that of $\mathscr{A}$, such that $|\Pr[b = b^* \text{inGame0}] - \Pr[b = b^* \text{inGame1}]| \leq A\, dv^{\text{dMLWE}}_{q,k,k,n,\eta}(\mathscr{D}) \leq A\, dv^{\text{dMLWE}}_{q,k,k+1,n,\eta}(\mathscr{D})$.

**Game 2**. In this game, assume that $(\mathbf{u}_0, v)$ is chosen uniformly at random from $R_q^k \times R_q$; i.e., both $(\mathbf{A}^T, \mathbf{u}_0)$ and $(\mathbf{b}'^T, v)$ are chosen uniformly at random from $R_q^{k\times k} \times R_q^k$ and $R_q^k \times R_q$, respectively. By the assumption that the decision Module-LWE problem $\text{dMLWE}_{q,k,k+1,n,\eta}$ is hard, we know that Game 1 and Game 2 are computationally indistinguishable. In other words, there exists a Module-LWE distinguisher $\mathscr{D}$ with the same running times as that of $A$, such that $|\Pr[b = b^* \text{in Game1}] - \Pr[b = b^* \text{in Game2}]| \leq A\, dv_{q,k,k+1,n,\eta}{}^{\text{dMLWE}}(\mathscr{D})$.

In Game 2, since $v \in R_q$ is uniformly random and $\overline{v} := dbl(v)$, according to Lemma 3 in Section 2.2, we know that real session key $k := \lceil \overline{v} \rceil_{2q,2}$ is uniformly random in $\{0,1\}^n$ given $c := \langle \overline{v} \rangle_{2q,2}$. Therefore, $\Pr[b = b^* \text{inGame2}] = (1/2)$.

Collecting the probabilities yields the required bound.

*3.4. Parameter Sets and Performance.* In this section, we give the parameter sets of the protocol described in Figure 1 and analysis of their performance. Based on the analysis of Section 3.1, the parameter sets of Kyber.KE can perfectly satisfy the correctness of our key exchange protocol. The parameter sets of the protocol described in Figure 1 and their performance are listed in Table 1, where "Alice $\longrightarrow$ Bob" (resp. "Bob $\longrightarrow$ Alice") denotes the communication cost in the first (resp., second) round.

It is known that Alice sends $(\mathbf{b}, \rho)$ to Bob in both the key exchange protocol described in Figure 1 and Kyber.KE [18]. However, Bob sends $(\mathbf{u}, c)$ to Alice in the second round of the key exchange protocol described in Figure 1 and sends $(\mathbf{u}, v, d)$ in Kyber.KE. We take the parameter set "Default" as an example to calculate the reduced communication cost in the key exchange protocol described in Figure 1. Both in our key exchange protocol and in Kyber.KE, $\mathbf{u}$ is a vector of three polynomials with 256 11-bit coefficients. In Kyber.KE, $v$ is a polynomial with 256 3-bit coefficients and $d$ is a 256-bit random string; i.e., $32 \times 3 + 32 = 128$ bytes are required to store $v$ and $d$. But in our key exchange protocol, each coefficient of $\overline{v} = \text{dbl}(v) \in R_{2q}$ is in $\mathbb{Z}_{2q}$; then the cross-rounding function takes $\overline{v}$ as input and outputs a 256-bit (32-byte) string $c$. According to the analysis above, our key exchange protocol reduces the total communication cost by 96-byte, i.e., 4.2%. Note that no matter which parameter sets we choose, the total communication cost reduced is invariant. Therefore, compared with Kyber.KE, our key exchange protocol reduces the total communication cost by 3.2% ~ 6.1% for different parameter sets.

In terms of computational efficient, the number of the most time-consuming operations, such as discrete binomial sampling and modular multiplication of ring elements, used in our key exchange protocol is less than that in Kyber.KE, since our key exchange protocol does not use the reencryption. In relatively time-consuming operations, we mainly talk about the modular multiplication of ring elements and numbers. Note that the randomized doubling function, cross-rounding function, modular rounding function, and compression algorithm are all modular multiplications of ring elements and numbers. Moreover, the time consumed by these operations is the same for vectors of the same dimension. In particular, Kyber.KE includes 6 compression functions and 4 decompress functions, whereas our key exchange protocol includes 2 compression functions, 2 decompress functions, 1 randomized doubling function, 1 cross-rounding function, and 1 modular rounding function. Therefore, compared with Kyber.KE, our key exchange protocol reduces the time consumed by modular multiplication of numbers and ring elements by approximately 30%, and the only difference between two protocols is that some operations are transferred from the initiator to the responder.

From the aspect of security, both our key exchange protocol and Kyber.KE are based on Module-LWE problem, and the scale of the problem is equal. Thus the security strength of our key exchange protocol is the same as that of Kyber.KE.

Table 1 shows that the communication costs of both rounds are equal, which means our key exchange protocol is a Diffie-Hellman-like symmetric key exchange protocol. Since symmetric key exchange protocols can ensure that the computation and communication costs of the two parties are roughly the same instead of occupying the computing resources of one party, it is more suitable to be deployed among users of the same level, such as the Internet of Vehicles (IOV) environment.

## 4. Conclusion

In this paper, we propose a Module-LWE-based key exchange protocol using Peikert's error reconciliation mechanism. Compared with Kyber.KE, our key exchange protocol reduces the total communication cost by 96 bytes, i.e., 3.2% ~ 6.1%, under the same post-quantum security levels and different parameter sets. Furthermore, our key exchange protocol slightly reduces the probability of session key agreement failure due to the reduction in the use of compression algorithms, has the less number of the most time-consuming operations (such as discrete binomial sampling and modular multiplication of ring elements) since the reencryption is not used, and reduces the time consumed by modular multiplication of numbers and ring elements by approximately 30%. Unlike the protocol using the KEM, our key exchange protocol is a Diffie-Hellman-like symmetric

protocol, which means the computation and communication costs of the two parties are roughly the same. With the advantages and properties above, our key exchange protocol is more suitable for the lightweight communication protocol, such as deployed in the IOV environment and smart home terminals.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References


[1] T. Dierks and E. Rescorla, "The transport layer security (tls) protocol version 1.2," *RFC*, vol. 5246, pp. 1–104, 2008.

[2] A. O. Freier, P. L. Karlton, and P. C. Kocher, "The secure sockets layer (ssl) protocol version 3.0," *RFC*, vol. 6101, pp. 1–67, 2011.

[3] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, Santa Fe, NM, USA, November 1994.

[4] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 84–93, STOC'05, Baltimore, MD, USA, May 2005.

[5] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, pp. 1–40, 2009.

[6] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 333–342, STOC'09, Bethesda, MD, USA, May 2009.

[7] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 6110, pp. 1–23, Monaco, France, May 2010.

[8] M. Rosca, D. Stehlé, and A. Wallet, "On the ring-LWE and polynomial-LWE problems,"vol. 10820, pp. 146–173, in *Proceedings of the 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 10820, EUROCRYPT'18, Tel Aviv, Israel, April 2018.

[9] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa, "Efficient public key encryption based on ideal lattices," in *Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security*, vol. 5912, pp. 617–635, Tokyo, Japan, December 2009.

[10] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping," pp. 309–325, 2012, https://eprint.iacr.org/2011/277.

[11] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 565–599, 2015.

[12] J. Ding, X. Xie, and X. Lin, "A simple provably secure key exchange scheme based on the learning with errors problem," *IACR Cryptology ePrint Archive*, vol. 20122688 pages, 2012.

[13] J. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 553–570, San Jose, CA, USA, May 2015.

[14] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange—a new hope," in *Proceedings of the 25th USENIX Security Symposium*, pp. 327–343, Austin, TX, USA, August 2016.

[15] J. Bos, C. Costello, L. Ducas et al., "Take off the ring! Practical, quantum-secure key exchange from LWE," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1006–1018, Vienna, Austria, October 2016.

[16] C. Peikert, "Lattice cryptography for the internet," in *Proceedings of the International Workshop on Post-Quantum Cryptography*, vol. 8772, pp. 197–219, Waterloo, Canada, October 2014.

[17] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "NEWHOPE without reconciliation," *IACR Cryptology ePrint Archive*, vol. 20161157 pages, 2016.

[18] J. Bos, L. Ducas, E. Kiltz et al., "CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM," in *Proceedings of the IEEE European Symposium on Security and Privacy*, pp. 353–367, London, UK, April 2018.

[19] G. Xue, B. Wang, Q. Qu, and W. Zhang, "Efficient lattice-based authenticated key exchange based on key encapsulation mechanism and signature," *IET Information Security*, vol. 15, no. 1, pp. 107–116, 2021.

[20] R. A. Mollin, *Algebraic Number Theory*, Chapman and Hall/CRC Press, New York, NY, USA, 2nd edition, 2011.

[21] M. R. Albrecht and A. Deo, "Large modulus ring-LWE >= module-LWE," in *Proceedings of the 23th International Conference on the Theory and Applications of Cryptology and Information Security*, vol. 10624, pp. 267–296, Hong Kong, China, January 2017.

[22] K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen, "Towards classical hardness of module-LWE: the linear rank case," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, vol. 12492, pp. 289–317, Daejeon, Republic of Korea, December 2020.

[23] K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen, "On the hardness of module-LWE with binary secret," in *Topics in Cryptology-CT-RSA 2021*, K. G. Paterson, Ed., vol. 12704, pp. 503–526, Springer, New York, NY, USA, 2021.

[24] H. Lin, Y. Wang, and M. Wang, "Hardness of Module-Lwe and Ring-Lwe on General Entropic Distributions," *IACR Cryptology ePrint Archive*, vol. 20201238 pages, 2020.