A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem

Jintai Ding, Xiang Xie, Xiaodong Lin

University of Cincinnati
Chinese Academy of Sciences
Rutgers University

Abstract. We use the learning with errors (LWE) problem to build a new simple and provably secure key exchange scheme. The basic idea of the construction can be viewed as certain extension of Diffie-Hellman problem with errors. The mathematical structure behind comes from the commutativity of computing a bilinear form in two different ways due to the associativity of the matrix multiplications:

$$(\mathbf{x}^t \times \mathbf{A}) \times \mathbf{y} = \mathbf{x}^t \times (\mathbf{A} \times \mathbf{y}),$$

where $\mathbf{x}, \mathbf{y}$ are column vectors and $\mathbf{A}$ is a square matrix. We show that our new schemes are more efficient in terms of communication and computation complexity compared with key exchange schemes or key transport schemes via encryption schemes based on the LWE problem. Furthermore, we extend our scheme to the ring learning with errors (RLWE) problem, resulting in small key size and better efficiency.

Keywords: lattice, key exchange protocol

1 Introduction

Key exchange protocol enables two users to exchange keys in untrusted channels without sharing secret materials in advance. The first and celebrated key exchange protocol is the Diffie-Hellman key exchange protocol [14] which is also a fundamental construction in public key cryptography. It is simple and elegant, after its invention, countless applications based on Diffie-Hellman key exchange protocol or the Diffie-Hellman problem were proposed.

1.1 DH Protocol vs Encryption

Diffie and Hellman [14] also introduced the notion of public key encryption, and Rivest, Shamir and Adleman [26] gave the first concrete public key encryption scheme. Namely, the well-known RSA encryption. With public key encryption in hand, one can construct a key exchange protocol as follows: Suppose Alice and Bob want to share some secret value. Alice encrypts a uniformly random message m_A using Bob's public key and sends the ciphertext to Bob. Bob does the same thing by encrypting a uniformly random message m_B using Alice's public key and sends the ciphertext to Alice. Once they get the ciphertexts, Alice and Bob decrypt them using the secret keys and compute $m_A \oplus m_B$. It is easy to see that Alice and Bob will share the same value $m_A \oplus m_B$. Instantiating with the RSA algorithm, the above construction results a very efficient key exchange protocol. However, the encryption-type key exchange protocol may have an important side-effect in practice: This approach relies on the user's private key to protect all the session keys, anyone with access to a copy of the private key can also uncover the session keys and thus decrypt everything.

The Diffie-Hellman protocol offers an alternative algorithm to RSA for cryptographic key exchange. The Diffie-Hellman protocol generates more secure session keys that can't be recovered simply by knowing the user's private key, a protocol security feature called *forward security*. In

order to decrypt all communication, now the adversary can no longer compromise just the user's private key, but the adversary has to compromise the session keys belonging to every individual communication session. In other words, using the diffie-hellman protocol, even the adversary knows the session key of some particular session, he still can not learn anything about the session keys established before this particular session. Actually, SSL also uses the Diffie-Hellman protocol to support forward security.

1.2 Lattice-Based Key Exchange Protocol

Since the hard number theory based problems like the discrete logarithm problem and the integer factorization problem are vulnerable to quantum computer attacks, it is important to find constructions based on problems believed to be resistant to quantum attacks. For instance, post-quantum key exchange is considered as a high priority by NIST [13].

The motivation of this paper is to build simple Diffie-Hellman like key exchange protocols based on lattices. Lattice-based public key cryptography has become a promising potential alternative to public key cryptography based on traditional number theory assumptions. One building block of lattice-based cryptography, especially in encryption, is the learning with errors (LWE) problem. After the introduction of the LWE problem by Regev [25], it has attracted a lot of attentions in theory and applications due to its good asymptotical efficiency and strong security guarantee. In a nutshell, the (decisional) LWE problem is to distinguish polynomially many noisy inner-product samples of the form $(\mathbf{a}, b \approx \langle \mathbf{a}, \mathbf{s} \rangle)$ from uniformly random ones, where $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ are uniformly random.[1] An attractive property of the LWE problem shown by Regev [25] is that to solve the average-case LWE problem is at least as hard as to (quantumly) solve some worst-case hard lattice problems. Many lattice-based primitives based on LWE have been discovered, such as public-key encryption [25,16,20], (hierarchical) identity-based encryption [16,1,12], functional encryption [3,2,5,17] and fully homomorphic encryption [9,7,6,10].

In the constructions mentioned above, a matrix form of the LWE problem is always used (*i.e.*, need sufficient many samples). The drawback of that is that it results in large (say quadratic) key size. To further improve the efficiency, Lyubashevsky, Peikert and Regev [21] introduced the ring learning with errors (RLWE) problem, which is to distinguish polynomially many noisy ring multiplications $(a, b \approx a \cdot s)$ from uniform distribution, where " $\cdot$" is the multiplicative operation over some ring and a, s are uniformly random from this ring. It is shown in [21] that to solve the RLWE problem is at least as hard as to solve some worst-case problems in *ideal* lattices, instead of general lattices.

What motivates the work in this paper is to try to build a simple key exchange protocol using the basic idea of Diffie-Hellman protocol but based on the LWE and RLWE problem. There are already related works in [18,19,11,15], but as far as we know there is not yet until very recently any provably secure key exchange protocols based on the LWE problem as a direct generalization of the Diffie-Hellman key exchange protocol, which is elegant in terms of its simplicity. We would like to point out that a very recent work by Peikert [24] also presents an efficient key exchange protocol, but with totally different techniques from ours. To achieve our goal, we first use the normal form of LWE problem suggested in [4] which means that the secret vector (ring element) in the LWE (RLWE) samples can be sampled from the error distribution. Then we introduce a notion called robust extractor, which may be of independent interest, to agree on an identical value from two close ones.

The key idea behind our new construction can be viewed as a way to share a secret given by the value of the bilinear function of two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{Z}_q^n$, where q, n are some integers, via

[1] $\mathbf{s}$ is secret and remains the same in all the samples.

the bilinear form:

$$Q(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{A} \mathbf{y},$$

where $\mathbf{A}$ is an $n \times n$ matrix in $\mathbb{Z}_q$. Surely in order to make the system provably secure, we need to introduce small errors to achieve our goal. The main contribution of this paper is to use this simple idea to build a simple and provably secure key exchange scheme. In the ring setting, this bilinear form is more direct, since the ring multiplicative operation is already commutative. Therefore, we extend our construction further based on the RLWE problem and resulting in a more efficient key exchange protocol.

1.3 Our Contributions and Techniques

The DH key exchange protocol was invented ahead of the RSA encryption scheme, and these two system are based on two very different mathematical principles The DH key exchange protocol is based on the commutativity of the power maps and the hardness of the discrete logarithm problem, while the RSA cryptosystems is based on special group automorphism and the hardness of prime factorization of integers.

Due to versatile applications of LWE in cryptography including a very elegant encryption scheme, it is a natural question to ask if we can construct a simple and elegant key exchange just like the DH scheme which, however, is not based on the same mathematical principles as that of the LWE encryption scheme. Our paper gives a positive answer. The fundamental difference is that we use the quantities of the usual LWE constructions, which serves the purpose of hiding the plaintext, and later canceled out, to serve the purpose as the exchanged key, and we rely also on the commutativity of matrix multiplication to compute bilinear maps, which was not used in the LWE constructions. Thus, from the point view of structural constructions of cryptosystems, we further demonstrate the versatility of the LWE problem, but in a way different from any previous construction before. The simplicity of the construction is very striking, though the elegance is slightly affected due to extra bits needed. This method should open possible doors to other applications, in particular, key distribution systems and new identity-based encryption systems.

More precisely, let's first recall the standard way to encrypt a message using LWE. Taking Regev's encryption for example, the public key consists of a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^m$, where $\mathbf{u}$ is an LWE sample, *i.e.*, $\mathbf{u} = \mathbf{A}^T \mathbf{s} + \mathbf{e}$ with uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ and error vector $\mathbf{e} \in \mathbb{Z}^m$. To encrypt, the user chooses uniformly random vectors $\mathbf{x} \in \{0,1\}^m$, and computes $\mathbf{c}_1 = \mathbf{A}\mathbf{x} \bmod q, c_2 = \langle \mathbf{u}, \mathbf{x} \rangle + m \cdot \lfloor q/2 \rfloor \bmod q$. When decrypting, the user computes $c_2 - \mathbf{s}^T \mathbf{c}_1 \bmod q$ to remove the common part $\mathbf{s}^T \mathbf{A} \mathbf{x}$ and recover the message from the "error". Instead, our constructions retrieve a shared secret from the common part for each party. More specifically, suppose the system is built with a uniformly random public parameter $\mathbf{M} \in \mathbb{Z}_q^{n \times n}$. In the key exchange stage, Alice and Bob choose two secret vectors $\mathbf{s}_A \in \mathbb{Z}_q^n$ and $\mathbf{s}_B \in \mathbb{Z}_q^n$ whose Euclidean norm is very small (much smaller than q), respectively. Alice and Bob then send $\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + \mathbf{e}_A \bmod q$ and $\mathbf{p}_B = \mathbf{M}^T \mathbf{s}_B + \mathbf{e}_B \bmod q$ to each other, where $\mathbf{e}_A$ and $\mathbf{e}_B$ are error vectors and with small norm. When receiving $\mathbf{p}_B$, Alice computes $\mathbf{s}_A^T \mathbf{p}_B$. Similarly, Bob computes $\mathbf{s}_B^T \mathbf{p}_A$. Note that $\mathbf{s}_A^T \mathbf{p}_B$ and $\mathbf{s}_B^T \mathbf{p}_A$ are very close to $\mathbf{s}_A^T \mathbf{M}^T \mathbf{s}_B$ because $\mathbf{s}_A, \mathbf{s}_B, \mathbf{e}_A, \mathbf{e}_B$ are small. We then propose a method to extract a shared secret from the two values which are very close.

To achieve this goal, we need a robust extractor. Before discussing the details, we need to slightly change the key exchange protocol. We set $\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + 2\mathbf{e}_A \bmod q$ and $\mathbf{p}_B = \mathbf{M}^T \mathbf{s}_B + 2\mathbf{e}_B \bmod q$. For simplicity, let $K_A = \mathbf{s}_A \mathbf{p}_B \bmod q$ and $K_B = \mathbf{s}_B \mathbf{p}_A \bmod q$. Note that $K_A - K_B$ is even and small and $\mathbf{p}_A$ and $\mathbf{p}_B$ are pseudorandom under the LWE assumption as long as $\gcd(2, q) = 1$. Once Alice and Bob hold K_A and K_B, respectively, our robust extractor enables Alice and Bob to

agree on the same value. More specifically, it works in the following way. In addition to K_A, Alice needs to receive a signal value from Bob to indicate K_B lies in the interval $[-q/4, q/4] \cap \mathbb{Z}$ or not. Assume K_B lies in $[-q/4, q/4] \cap \mathbb{Z}$, and denote $K_A = K_B + 2\delta \bmod q$. Observe that $2\delta \ll q/4$, then $K_B + 2\delta \bmod q = K_B \bmod q + 2\delta$. This gives us

$$K_A \bmod q = K_B \bmod q + 2\delta \text{ (in } \mathbb{Z}),$$

and a further modulo operation on 2 yields $(K_A \bmod q) \bmod 2 = (K_B \bmod q) \bmod 2$. Similar analysis could be taken when $K_B \in \mathbb{Z}_q$ lies outside the interval $[-q/4, q/4] \cap \mathbb{Z}$.

However, this method actually results in another problem. To get a secure key exchange protocol, we need the extracted key to be uniformly random. Assuming K_B is uniformly random. According to our robust extractor mentioned before, the adversary will know that K_B lies in $[-q/4, q/4] \cap \mathbb{Z}$ or not. The problem is that in our construction we need to choose odd q, which means that $\mathbb{Z}_q$ has different numbers of even values and odd values. Therefore, the distribution of $(K_B \bmod q) \bmod 2$ always has bias when $K_B \bmod q$ is uniformly random and q is polynomial in the security parameter. To overcome this problem, we introduce a randomized algorithm to generate the signal, this new randomized algorithm allows our robust extractor to remove the bias of the distribution of the extracted key, while preserving the functionality. With this randomized signal generation algorithm, we show that when K_B is uniformly random in $\mathbb{Z}_q$, the extracted value on K_B is uniformly random in $\{0, 1\}$, even the adversary knows the signal of K_B.

Using the property of the randomized signal, we now only need to show that K_B is pseudorandom under the LWE assumption. First, due to the squared form of the matrix $\mathbf{M}$, the transcripts $\mathbf{p}_A$ and $\mathbf{p}_B$ are just LWE samples with independent secrets. This implies that they can be replaced by uniformly random vectors in $\mathbb{Z}_q^n$ under the LWE assumption. In order to make the extracted key look random, we additionally add noises to $\mathbf{s}_A^T\mathbf{p}_B$ and $\mathbf{s}_B^T\mathbf{p}_A$, respectively. Notice that if $\mathbf{p}_A$ and $\mathbf{p}_B$ are uniformly random, the "noisy" form of $\mathbf{s}_A^T\mathbf{p}_B$ and $\mathbf{s}_B^T\mathbf{p}_A$ are LWE samples, which are again pseudorandom under LWE assumption. In the security proof, we use standard hybrid games and deal exclusively with the squared matrix $\mathbf{M}$.

In terms of practical applications, one may argue that we can always construct easily a key exchange scheme using a public key encryption scheme, and why do we need a new key exchange scheme? One possible reason is the forward security we discussed at the beginning. Besides, we can compare our scheme with a key exchange based on the LWE-type encryption, but this comparison surely depends on the assumption what is overhead cost and what is the real key exchange cost. We can show that there could be indeed substantial advantage in our scheme in terms of communication cost and (or) computation cost, we will illustrate the point by using our LWE based one and RLWE based one respectively.

Besides, we also give an interactive multiparty key exchange protocol. This protocol can be viewed as a generalization of our two party protocol. Although the provable security of the protocol seems plausible but we do not know how to prove it, and we leave it as an open problem.

1.4 Organization

In Section 2, we give some basic notations and facts. The protocol based on LWE problem is given in Section 3, and the more efficient protocol based on RLWE problem is given in Section 4. In Section 5, we describe our interactive key exchange scheme. In the last section, we will present the conclusion and the discussion.

2 Preliminaries

Notations. We use bold capital letters to denote matrices, and bold lowercase letters to denote vectors. The notation $\mathbf{A}^T$ denotes the transpose of the matrix $\mathbf{A}$. A function $\mathrm{negl}(n)$ is *negligible*, if it vanishes faster than the inverse of any polynomial in n. The *statistical distance* between two distributions X, Y over some finite or countable set S is defined as $\Delta(X,Y) = \frac{1}{2}\sum_{s\in S}\big|\Pr[X = s] - \Pr[Y = s]\big|$. X and Y are statistically indistinguishable if $\Delta(X,Y)$ is negligible.

Let Λ be a discrete subset of $\mathbb{Z}^m$. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any positive parameter $\sigma \in \mathbb{R}_{>0}$, let $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2)$ be the Gaussian function on $\mathbb{R}^m$ with center $\mathbf{c}$ and parameter σ. Denote $\rho_{\sigma,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda}\rho_{\sigma,\mathbf{c}}(\mathbf{x})$ be the discrete integral of $\rho_{\sigma,\mathbf{c}}$ over Λ, and $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ be the discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter σ. Specifically, for all $\mathbf{y} \in \Lambda$, we have $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{y})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}$. For notional convenience, $\rho_{\sigma,\mathbf{0}}$ and $\mathcal{D}_{\Lambda,\sigma,\mathbf{0}}$ are abbreviated as ρ_σ and $\mathcal{D}_{\Lambda,\sigma}$, respectively.

The Learning with Errors Problem We recall the learning with errors (LWE) problem, a classical hard problem on lattices defined by Regev [25].

Definition 1. *Let $n \geq 1$ and $q \geq 2$ be integers, let $\alpha \in (0,1)$. For $\mathbf{s} \in \mathbb{Z}_q^n$, let $A_{\mathbf{s},\alpha}$ be the distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, $e \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, and outputting $(\mathbf{a}, \langle\mathbf{a},\mathbf{s}\rangle + e)$.*

The LWE problem is : for uniformly random $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, given poly(n) number of samples that are either from $A_{\mathbf{s},\alpha}$ or uniformly random in $\mathbb{Z}_q^n \times \mathbb{Z}_q$, output 0 if the former holds and 1 if the latter holds.

It is known that when $\alpha q \geq 2\sqrt{n}$ and $q = \mathrm{poly}(n)$, this decision problem is at least as hard as approximating several problems on n-dimensional lattices in the *worst-case* to within $\widetilde{O}(n/\alpha)$ factors with a quantum computer [25] or on a classical computer for a subset of these problems [23]. A very recent work by Brakerski *et al.* [8] even shows the classical hardness of LWE for any polynomial q but the security losses with about a $\sqrt{n}$ factor. A simple analysis shows that for any $t \in \mathbb{Z}^+$ and $gcd(t,q) = 1$, then the LWE assumption still holds if we choose $b = \langle\mathbf{a},\mathbf{s}\rangle + te$. The HNF-LWE assumption [4] says that the hardness preserves even if we choose the secret from the error distribution, i.e. $\mathbf{s} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. We will exclusively use this assumption.

We will use a bound of the norm of the Gaussian distribution as follows.

Lemma 1 ([22]). *For any $s \geq \omega(\sqrt{\log n})$, then we have*

$$\Pr_{x\leftarrow\mathcal{D}_{\mathbb{Z}^n,s}}[\|x\| > s\sqrt{n}] \leq 2^{-n}.$$

2.1 Robust Extractors

We now put forward a notion called robust extractor. Informally, a robust extractor enables two parties to extract an identical information from two close elements with some additional hint.

Definition 2 (Robust Extractors). *An algorithm E is a robust extractor on $\mathbb{Z}_q$ with error tolerance δ with respect to a hint function S, if the following holds:*

- *The deterministic algorithm E takes as input an $x \in \mathbb{Z}_q$ and a signal $\sigma \in \{0,1\}$, outputs $k = E(x,\sigma) \in \{0,1\}$.*
- *The hint algorithm S takes as input a $y \in \mathbb{Z}_q$ and outputs a signal $\sigma \leftarrow S(y) \in \{0,1\}$.*

– *For any $x, y \in \mathbb{Z}_q$ such that $x - y$ is even and $|x - y| \leq \delta$, then it holds that $E(x, \sigma) = E(y, \sigma)$, where $\sigma \leftarrow S(y)$.*

We use an robust extractor to guarantee the correctness of our protocol. More specifically, in our protocol, the parties will compute two very close values in $\mathbb{Z}_q$. In order to agree on a common value, one party additionally send a signal of his value. The both parties computes the shared key using the robust extractor.

We also note that the errors of x, y in the definition can be set to be multiple of t, where t is a small integer. For simplicity, we only focus on the case $t = 2$. Our construction can be extended to any small integer t.

Signal Functions. We now define two "signal" functions, which are used in the robust extractor. For prime $q > 2$, we define $\sigma_0(x), \sigma_1(x)$ from $\mathbb{Z}_q$ to $\{0, 1\}$ as follows.

$$\sigma_0(x) = \begin{cases} 0, x \in [-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor]; \\ 1, \text{otherwise.} \end{cases} ; \quad \sigma_1(x) = \begin{cases} 0, x \in [-\lfloor \frac{q}{4} \rfloor + 1, \lfloor \frac{q}{4} \rfloor + 1]; \\ 1, \text{otherwise.} \end{cases}$$

In our robust extractor, we define the hint algorithm S as: for any $y \in \mathbb{Z}_q$, $S(y) = \sigma_b(y)$, where $b \xleftarrow{\$} \{0, 1\}$. The robust extractor is defined as:

$$E(x, \sigma) = (x + \sigma \cdot \frac{q-1}{2} \bmod q) \bmod 2.$$

Lemma 2. *Let $q > 8$ be an odd integer, the function E defined above is a robust extractor with respect to S with error tolerance $\frac{q}{4} - 2$.*

Proof. For any $x, y \in \mathbb{Z}_q$ such that $x - y = 2\varepsilon$ and $|2\varepsilon| \leq \frac{q}{4} - 2$. Let $\sigma \leftarrow S(y)$, due to the definition of S, it is not hard to see that $|y + \sigma \cdot \frac{q-1}{2} \bmod q| \leq \frac{q}{4} + 1$ for any b used in S to generate σ. We have

$$x + \sigma \cdot \frac{q-1}{2} \bmod q = y + \sigma \cdot \frac{q-1}{2} + 2\varepsilon \bmod q = (y + \sigma \cdot \frac{q-1}{2}) \bmod q + 2\varepsilon,$$

because $|(y + \sigma \cdot \frac{q-1}{2}) \bmod q + 2\varepsilon| \leq \frac{q}{4} + 1 + |2\varepsilon| \leq \frac{q-1}{2}$. This implies

$$E(x, \sigma) = (x + \sigma \cdot \frac{q-1}{2} \bmod q) \bmod 2 = (y + \sigma \cdot \frac{q-1}{2} \bmod q) \bmod 2 = E(y, \sigma).$$

The claim follows □

Our robust extractor enjoys a very nice property which says that for uniformly random $x \in \mathbb{Z}_q$, $E(x, \sigma)$ is uniform in $\{0, 1\}$ even conditioned on σ, where $\sigma \leftarrow S(x)$. This property is crucial in our security proof.

Lemma 3. *For any odd $q > 2$, if x is uniformly random in $\mathbb{Z}_q$, then $E(x)$ is uniformly random conditioned on σ, where $\sigma \leftarrow S(x)$.*

Proof. For any $\sigma, b' \in \{0, 1\}$, we have

$$\begin{aligned} &\Pr_{x \xleftarrow{\$} \mathbb{Z}_q, b \xleftarrow{\$} \{0,1\}} [E(x, \sigma) = b' \mid \sigma_b(x) = \sigma] \\ &= \frac{1}{2} \Pr_{x \xleftarrow{\$} \mathbb{Z}_q} [E(x, \sigma) = b' \mid \sigma_0(x) = \sigma] + \frac{1}{2} \Pr_{x \xleftarrow{\$} \mathbb{Z}_q} [E(x, \sigma) = b' \mid \sigma_1(x) = \sigma] \\ &= \frac{1}{2} \cdot \frac{\Pr_{x \xleftarrow{\$} \mathbb{Z}_q} [E(x, \sigma) = b' \wedge \sigma_0(x) = \sigma]}{\Pr_{x \xleftarrow{\$} \mathbb{Z}_q} [\sigma_0(x) = \sigma]} + \frac{1}{2} \cdot \frac{\Pr_{x \xleftarrow{\$} \mathbb{Z}_q} [E(x, \sigma) = b' \wedge \sigma_1(x) = \sigma]}{\Pr_{x \xleftarrow{\$} \mathbb{Z}_q} [\sigma_1(x) = \sigma]} \end{aligned}$$

Denote $I = [-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor]$ the interval such that σ_0 equals 0, then $I+1$ is the interval such that σ_1 equals 0. It is easy to see that $|I| = |I+1| = 2\lfloor \frac{q}{4} \rfloor + 1$. We separately consider two cases, when $\sigma = 0$ and $\sigma = 1$. For $\sigma = 0$, we have that

$$\Pr_{x \xleftarrow{\$} \mathbb{Z}_q}[\sigma_0(x) = 0] = \Pr_{x \xleftarrow{\$} \mathbb{Z}_q}[\sigma_1(x) = 0] = \frac{2\lfloor \frac{q}{4} \rfloor + 1}{q}.$$

Let $I_0 = \{x : x \in I \wedge x \bmod 2 = 0\}$ and $I_1 = \{x : x \in I \wedge x \bmod 2 = 1\}$ and similarly for $(I+1)_0, (I+1)_1$. Then we have $|I_0| + |(I+1)_0| = |I|$ and $|I_1| + |(I+1)_1| = |I|$. Therefore,

$$\Pr_{x \xleftarrow{\$} \mathbb{Z}_q}[E(x,0) = b' \wedge \sigma_0(x) = 0] = \Pr_{x \xleftarrow{\$} \mathbb{Z}_q}[x \in I_{b'}] \text{ and } \Pr_{x \xleftarrow{\$} \mathbb{Z}_q}[E(x,0) = b' \wedge \sigma_1(x) = 0] = \Pr_{x \xleftarrow{\$} \mathbb{Z}_q}[x \in (I+1)_{b'}].$$

This implies that

$$\Pr_{x \xleftarrow{\$} \mathbb{Z}_q, b \xleftarrow{\$} \{0,1\}}[E(x,0) = b' | \sigma_b(x) = 0] = \frac{1}{2} \cdot \frac{q}{2\lfloor \frac{q}{4} \rfloor + 1} \cdot \frac{|I_{b'}| + |(I+1)_{b'}|}{q} = \frac{1}{2}.$$

For $\sigma = 1$, we first note that the intervals $\mathbb{Z}_q \backslash I$ and $\mathbb{Z}_q \backslash (I+1)$ have even numbers, *i.e.,* $q - (2\lfloor \frac{q}{4} \rfloor + 1)$. Therefore, we have:

$$\Pr_{x \xleftarrow{\$} \mathbb{Z}_q}[E(x,1) = b' \wedge \sigma_0(x) = 1] = \Pr_{x \xleftarrow{\$} \mathbb{Z}_q}[E(x,1) = b' \wedge \sigma_1(x) = 1] = \frac{q - (2\lfloor \frac{q}{4} \rfloor + 1)}{2q}.$$

A routine calculation shows that $\Pr_{x \xleftarrow{\$} \mathbb{Z}_q, b \xleftarrow{\$} \{0,1\}}[E(x,1) = b' | \sigma_b(x) = 1] = \frac{1}{2}$. This completes the proof □

3 Key Exchange Protocol from LWE

Key exchange protocols are very important cryptographic protocols. The original Diffien-Hellman key exchange protocol [14] is built on the fact that the exponential maps are commutative, namely

$$g^{ab} = (g^a)^b = (g^b)^a,$$

over some multiplicative group $\mathbb{G}$ with large order p. If we look carefully why the key exchange above works, one realizes we may do the same thing using the associativity and commutativity of computing the value of bilinear form, namely,

$$\mathbf{x}^T \mathbf{M} \mathbf{y} = (\mathbf{x}^T \mathbf{M}) \mathbf{y} = \mathbf{x}^T (\mathbf{M} \mathbf{y}),$$

where $\mathbf{M}$ is an $n \times n$ matrix in $\mathbb{Z}_q$ and $\mathbf{x}, \mathbf{y}$ are vectors in $\mathbb{Z}_q^n$. Here, this computation can be viewed a pairing of the two vector $\mathbf{x}, \mathbf{y}$ via the corresponding bilinear form.

Surely we need to introduce small errors. Namely, the idea of LWE problem, to make the scheme secure. Our basic idea is that we can use the Hermit normal form of LWE (HNF-LWE) problem to build a key exchange protocol like the Diffie-Hellman key exchange protocol. The protocol can be set up as follows.

3.1 Construction

Two parties Alice and Bob decide to do a key exchange over an open channel. Let E be the robust extractor with respect to S as defined in Section 2.

- The system first generates the public parameters q, n, α, where $q > 2$ is prime. Sample a uniformly random matrix $\mathbf{M} \leftarrow \mathbb{Z}_q^{n\times n}$.

- Alice chooses a secret vector $\mathbf{s}_A \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then, Alice computes $\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + 2\mathbf{e}_A \mod q$, where $\mathbf{e}_A \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Sends $\mathbf{p}_A$ to Bob.

- Receiving $\mathbf{p}_A$, Bob first chooses a secret vector $\mathbf{s}_B \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and an error $e'_B \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$. Computes $K_B = \mathbf{p}_A^T \cdot \mathbf{s}_B + 2e'_B \mod q$. Then he samples $\mathbf{e}_B \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$, computes $\mathbf{p}_B = \mathbf{M}^T \cdot \mathbf{s}_B + 2\mathbf{e}_B \mod q$. Bob computes $\sigma \leftarrow S(K_B)$ and obtains the shared key $SK_B = E(K_B, \sigma)$. Finally Bob sends $(\mathbf{p}_B, \sigma)$ to Alice.

- Once getting $(\mathbf{p}_B, \sigma)$, Alice samples $e'_A \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$ and computes $K_A = \mathbf{s}_A^T \mathbf{p}_B + 2e'_A \mod q$, and obtains $SK_A = E(K_A, \sigma)$.

Correctness We now show that if Alice and Bob run the protocol honestly, they will share an identical key.

Lemma 4. *If $8(\alpha q)^2 \cdot n \le \frac{q}{4} - 2$, then $SK_A = SK_B$ with overwhelming probability.*

Proof. From the form of K_A, K_B, we have $K_A = \mathbf{s}_A^T(\mathbf{M}^T \cdot \mathbf{s}_B + 2\mathbf{e}_B) + 2e'_A = \mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + 2(\mathbf{s}_A^T\mathbf{e}_B + e'_A)$, $K_B = (\mathbf{s}_A^T\mathbf{M}^T + 2\mathbf{e}_A^T)\mathbf{s}_B + 2e'_B = \mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + 2(\mathbf{e}_A^T\mathbf{s}_B + e'_B)$. Hence, we have

$$K_A = K_B + 2(\mathbf{s}_A^T\mathbf{e}_B + e'_A - \mathbf{e}_A^T\mathbf{s}_B - e'_B) \bmod q.$$

From Lemma 1, we have that

$$|2(\mathbf{s}_A^T\mathbf{e}_B + e'_A - \mathbf{e}_A^T\mathbf{s}_B - e'_B)| \le 8 \cdot (\alpha q\sqrt{n}) \cdot (\alpha q\sqrt{n}) = 8(\alpha q)^2 \cdot n \le \frac{q}{4} - 2,$$

with overwhelming probability. Because E is a robust extractor with respect to S with error tolerance $\frac{q}{4} - 2$ due to Lemma 2. We have $SK_A = E(K_A, \sigma) = E(K_B, \sigma) = K_B$.

Moreover, we show that $SK_A = SK_B = (\mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + \sigma(K_B) \cdot \frac{q-1}{2} \bmod q) \bmod 2$. This is because $\mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + \sigma(K_B) \cdot \frac{q-1}{2} = K_B + \sigma(K_B) \cdot \frac{q-1}{2} - 2(\mathbf{e}_A^T\mathbf{s}_B + e'_B) \bmod q$, and $|K_B + \sigma(K_B) \cdot \frac{q-1}{2}| < \frac{q}{4} + 1$. Hence $\left(\mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + \sigma(K_B) \cdot \frac{q-1}{2}\right) \bmod q = \left(K_B + \sigma(K_B) \cdot \frac{q-1}{2}\right) \bmod q - 2(\mathbf{e}_A^T\mathbf{s}_B + e'_B)$ (in $\mathbb{Z}$), and we have $SK_A = SK_B = (\mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + \sigma(K_B) \cdot \frac{q-1}{2} \bmod q) \bmod 2$. □

Remark. We note that for the correctness of our protocol, Bob has to send a signal to Alice to tell her that the resulting K_B is in the range close to $[-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor]$ or not. The reason is to make sure that the error terms in K_B and K_A do not result different modulo q operations. The drawback of the signal is that the adversary will also know the "main" part, say $\mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B \bmod q$ or $\mathbf{s}_A^T\mathbf{M}^T\mathbf{s}_B + \frac{q-1}{2} \bmod q$, lies close to $[-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor]$ or not. However, this does not harm the security, since if we can show K_B is (pseudo)random in $\mathbb{Z}_q$, the additional modulo 2 operation makes the shared key uniform in $\mathbb{Z}_2$ due to Lemma 3.

Parameter Selection. A reasonable way to select the parameters is $n = \lambda$, $q = \lambda^4$, $\alpha = 1/\lambda^3$. It's easy to verify that $\alpha q \ge \sqrt{n}$ and the correctness holds.

3.2 Security

We now define the passive security of a key exchange protocol. Intuitively, any PPT adversary should not distinguish a real shared key to a random one even if he gets the transcripts of the protocol. More specifically, we define the advantage of an adversary $\mathcal{A}$:

$$\mathbf{Adv}_{\mathcal{A}} = \Pr[b' \leftarrow \mathcal{A}(\text{transcripts}, K_b), b \xleftarrow{\$} \{0,1\}, K_0 \text{ is real}, K_1 \text{ is random} : b = b'] - 1/2.$$

Definition 3. *We say a key exchange protocol is secure under passive adversary, if for any PPT adversary the above advantage is negligible.*

We now slightly change the definition according to our construction, we do not need the adversary to distinguish the shared key, instead we want it to distinguish K_A or K_B from uniformly random in $\mathbb{Z}_q$. Namely, we prove that

$$\Pr[b' \leftarrow \mathcal{A}(\mathbf{M}, \mathbf{p}_A, \mathbf{p}_B, K_b), b \xleftarrow{\$} \{0,1\}, K_0 = K_B, K_1 \xleftarrow{\$} \mathbb{Z}_q : b = b'] - 1/2$$

is negligible (we can also replace $K_0 = K_A$). Lemma 3 guarantees that this definition is sufficient.

Theorem 1. *The construction above is secure against passive PPT adversaries, if the HNF-LWE assumption holds.*

Proof. We prove the security by a series of games. The first game $\mathbf{Game}_0$ is the real game which the adversary gets the real K_B, while in the last game $\mathbf{Game}_4$ the adversary gets a uniformly random K_B. We show that the views of $\mathbf{Game}_0$ and $\mathbf{Game}_4$ are computational indistinguishable for any PPT adversaries, under the HNF-LWE assumption.

$\mathbf{Game}_0$. This is the real game between the protocol challenger and the passive adversary $\mathcal{A}$. That is the adversary obtains $\mathbf{M}, \mathbf{p}_A, \mathbf{p}_B, \sigma_b(K_B), K_B$, where $\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + 2\mathbf{e}_A$, $\mathbf{p}_B = \mathbf{M}^T\mathbf{s}_B + 2\mathbf{e}_B$ and $K_B = \mathbf{p}_A^T\mathbf{s}_B + 2e'_B$. Then $\mathcal{A}$ outputs a guess b'.

$\mathbf{Game}_1$. This game is identical to $\mathbf{Game}_0$ except that instead of setting $\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + 2\mathbf{e}_A$ and $K_B = \mathbf{p}_A^T\mathbf{s}_B + 2e'_B$. The challenger sets $\mathbf{p}_A = \mathbf{b}_A$ and $K_B = \mathbf{b}_A^T \cdot \mathbf{s}_B + 2e'_B$, where $\mathbf{b}_A \xleftarrow{\$} \mathbb{Z}_q^n$.

In Lemma 5, we show that under the HNF-LWE assumption, the views in $\mathbf{Game}_0$ and $\mathbf{Game}_1$ are computationally indistinguishable for any PPT passive adversaries. Note that here $\mathbf{s}_A$ is chosen according to the error distribution.

$\mathbf{Game}_2$. This game is identical to $\mathbf{Game}_1$ except that instead of setting $\mathbf{p}_B = \mathbf{M}^T\mathbf{s}_B + 2\mathbf{e}_B$ and $K_B = \mathbf{b}_A^T \cdot \mathbf{s}_B + 2e'_B$. The challenger sets $\mathbf{p}_B = \mathbf{b}_B$ and $K_B = u$, where $\mathbf{b}_B \xleftarrow{\$} \mathbb{Z}_q^n$ and $u \xleftarrow{\$} \mathbb{Z}_q$.

We show the views for any PPT passive adversaries in $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are computationally indistinguishable, if the HNF-LWE assumption holds. The proof is given in Lemma 6.

$\mathbf{Game}_3$. This game is identical to $\mathbf{Game}_2$ except that instead of setting $\mathbf{p}_A = \mathbf{b}_A$. The challenger sets $\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + 2\mathbf{e}_A$.

In Lemma 7, we prove the views in $\mathbf{Game}_2$ and $\mathbf{Game}_3$ are computationally indistinguishable, if the HNF-LWE assumption holds.

$\mathbf{Game}_4$. This game is identical to $\mathbf{Game}_3$ except that instead of setting $\mathbf{p}_B = \mathbf{b}_B$. The challenger sets $\mathbf{p}_B = \mathbf{M}^T\mathbf{s}_B + 2\mathbf{e}_B$.

In Lemma 8, we prove that the views in $\mathbf{Game}_3$ and $\mathbf{Game}_4$ are indistinguishable, if the HNF-LWE assumption holds. The claim follows from Lemma 5,6,7,8 directly. □

Lemma 5. *Any PPT passive adversary can not distinguish* $\boldsymbol{Game}_0$ *and* $\boldsymbol{Game}_1$*, if the HNF-LWE assumption holds.*

Proof. We prove the lemma by showing that if there exists an adversary $\mathcal{A}$ who can distinguish $\mathbf{Game}_0$ and $\mathbf{Game}_1$, then we can construct another adversary $\mathcal{B}$ to distinguish the HNF-LWE samples from uniform. $\mathcal{B}$ works as follows. Once obtaining challenges $(\mathbf{M}, \mathbf{b}_A) \in \mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^n$ from the HNF-LWE challenger, where $\mathbf{b}_A$ is either $\mathbf{Ms} + 2\mathbf{e}$ or uniformly random in $\mathbb{Z}_q^n$, $\mathcal{B}$ samples $\mathbf{s}_B \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and sets $K_B = \mathbf{b}_A^T\mathbf{s}_B + 2e'_B$ and computes $\mathbf{p}_B = \mathbf{M}^T\mathbf{s}_B + 2\mathbf{e}_B$. Finally $\mathcal{B}$ sends $(\mathbf{M}, \mathbf{p}_A = \mathbf{b}_A, \mathbf{p}_B, \sigma_b(K_B), K_B)$ to $\mathcal{A}$. $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. We note that $\mathcal{B}$ can sample the errors and compute $\sigma_b(K_B)$ by himself.

If $\mathbf{b}_A$ is an LWE sample, then what $\mathcal{A}$ obtains are exactly the same as in $\mathbf{Game}_0$; if $\mathbf{b}_A$ is uniformly random in $\mathbb{Z}_q^n$, then what $\mathcal{A}$ obtains are exactly the same as in $\mathbf{Game}_1$. This implies that if $\mathcal{A}$ can distinguish $\mathbf{Game}_0$ and $\mathbf{Game}_1$ with noticeable advantage, then $\mathcal{B}$ can distinguish HNF-LWE samples from uniformly random with the same advantage. This finishes the proof. □

Lemma 6. *Any PPT passive adversary can not distinguish* $\boldsymbol{Game}_1$ *and* $\boldsymbol{Game}_2$*, if the HNF-LWE assumption holds.*

Proof. We prove this lemma by showing that if there exists an adversary $\mathcal{A}$ distinguishes $\mathbf{Game}_1$ and $\mathbf{Game}_2$, then we can construct a PPT adversary $\mathcal{B}$ to distinguish the HNF-LWE samples from uniform. $\mathcal{B}$ works as follows. Once obtaining challenges $(\mathbf{M}, \mathbf{b}_B) \in \mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^n$ and $(\mathbf{b}_A, u) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{u}$ and u are either $\mathbf{M}^T\mathbf{s} + 2\mathbf{e}, \mathbf{b}_A^T\mathbf{s} + 2e$ or uniformly random in $\mathbb{Z}_q^n$ and $\mathbb{Z}_q$ respectively, $\mathcal{B}$ sets $\mathbf{p}_A = \mathbf{b}_A$, let $\mathbf{p}_B = \mathbf{b}_B$ and $K_B = u$, and computes $\sigma_b(K_B)$. $\mathcal{B}$ sends $(\mathbf{M}, \mathbf{p}_A, \mathbf{p}_B, \sigma_b(K_B), K_B)$ to $\mathcal{A}$, and outputs whatever $\mathcal{A}$ outputs. Note that if $\mathbf{b}_B, u$ are LWE samples, then what $\mathcal{A}$ gets are exactly the same as in $\mathbf{Game}_1$; if $\mathbf{b}_B, u$ are uniformly random, then what $\mathcal{A}$ gets are exactly the same as in $\mathbf{Game}_2$. Therefore, if $\mathcal{A}$ can distinguish the two games with noticeable advantage, then $\mathcal{B}$ can break the HNF-LWE problem with noticeable advantage. This complete the proof. □

Lemma 7. *Any PPT passive adversary can not distinguish* $\boldsymbol{Game}_2$ *and* $\boldsymbol{Game}_3$*, if the HNF-LWE assumption holds.*

Proof. The proof is similar to Lemma 5, except we still choose K_B uniformly from $\mathbb{Z}_q$. □

Lemma 8. *Any PPT passive adversary can not distinguish* $\boldsymbol{Game}_3$ *and* $\boldsymbol{Game}_4$*, if the HNF-LWE assumption holds.*

Proof. The proof is similar to Lemma 6, except we still choose K_B uniformly from $\mathbb{Z}_q$. □

Key Exchange Protocol with Multiple Bits. In order to get multiple shared secret bits in the protocol, one can use the matrix secret form of LWE assumption. More specifically, Alice and Bob choose secret matrix $\mathbf{S}_A, \mathbf{S}_B \in \mathbb{Z}_q^{n\times n}$ instead of $\mathbf{s}_A, \mathbf{s}_B$ (still from the error distribution). It's easy to extend the other part to get multiple shared secret bits. The security is straightforwardly from the underlying HNF-LWE assumption by standard hybrid argument.

Comparisons . We now give some comparisons with directly using public key encryption scheme to do key exchange. The main idea of using PKE is as follows: for two parties A and B with key pair (pk_A, sk_A) and (pk_B, sk_B), respectively. A chooses a bit a uniformly at random, and encryption it by using B's public key $c_B = Enc(pk_B, a)$ and sends c_B to B. Similarly B chooses a uniform bit b and sends $c_A = Enc(pk_A, b)$ to A. A and B decrypt the ciphertext by using their own secret keys and compute $a \oplus b$. We note that, by using the PKE based key exchange, the users need to first download the public key of the party he/she wants to communicate. Therefore, it incurs more communication complexity. While in our scheme, the public parameter $\mathbf{M}$ is generated once for all, namely a public authority like NIST can generate one matrix $\mathbf{M}$ that any two parties can use the same $\mathbf{M}$, while the security is not affected. We focus on LWE-based encryptions and estimate the complexity for 1 bit secret key. The comparisons are given in Table 1:

Table 1. Comparisons between LWE-based ones for 1-bit secret key

	Pub. Param.	Commun. Comp.	Comput. Comp.	Assumption
Reg'05 [25]	$4(n+1)n\log^2 q$	$(4n^2+6n)\log^2 q + 2\log q$	$4n^2 \log q$	$\mathsf{SIVP}_{\tilde{O}(n^3)}$
LP'11 [20]	$4n^2 \log q$	$4(n^2+n)\log q$	$6n^2$	$\mathsf{SIVP}_{\tilde{O}(n^3)}$
Ours	$n^2 \log q$	$2n\log q + 1$	$2n^2$	$\mathsf{SIVP}_{\tilde{O}(n^4)}$

Pub. Param. means the size of public parameter; Commun. Comp. means the communication complexity; Comput. Comp. means the computation complexity and is estimated by the number of multiplications in $\mathbb{Z}_q$.

We also compare the efficiency of our scheme with key transport schemes based on PKE from LWE. Intuitively, in a key transport scheme, party A chooses a uniformly random bit s and encrypts it by using B's public key to $c = Enc(pk_B, s)$ and then sends c to B. B uses its secret key to decrypt c and recover s. The session key between A and B is s. A drawback of key transport schemes is that the shared key is totally determined by one party, while in our protocol, the shared key is determined by both involved parties. Another method to construct key transport schemes is to use key encapsulation mechanism (KEM). By now (except [24]), the efficiency of all the KEM schemes based on LWE is almost the same as the encryption scheme. We would like to point out that our protocol can also be easily extended to a more efficient KEM scheme. Therefore, when considering key transport schemes, the communication complexity and computation complexity will be half as the key exchange schemes based on PKE. From the results in Table 1, even in such a scenario, the efficiency of our scheme is still substantially better in terms of communication complexity and computation complexity.

4 Key Exchange Protocol from Ring-LWE

In this section, we show how to get a more efficient key exchange protocol from the ring learning with errors (RLWE) problem [21]. Consider the ring $R = \mathbb{Z}[x]/f(x)$, where $f(x) = x^n + 1$ and n is a power of 2. For an integer q, let $R_q = R/qR$. Any element in R_q is represented by a degree $n-1$ polynomial, which can also be viewed as a vector with its corresponding coefficients as its entries. For an element

$$a(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1},$$

we define $\|a\| = \max|a_i|$, the ℓ_∞ norm of the vector $(a_0, a_1, ..., a_{n-1})$. Furthermore, it's easy to get that $\|x \cdot y\| \leq n\|x\| \cdot \|y\|$ for any $x, y \in R$. For convenience, we do not give the specific description of the error distribution, since we only care the norm of the element from the distribution. Denote

χ (whose support is R) to be β-bounded, if $\Pr[\|x\| > \beta : x \leftarrow \chi] \leq \mathrm{negl}(n)$. We recall the definition of RLWE proposed by Lyubashevsky, Peikert and Regev [21].

Definition 4. *Let $n \geq 1$ be a power of 2 and $q \geq 2$ be an integer, let $R = \mathbb{Z}[x]/f(x)$, where $f(x) = x^n + 1$, and $R = R/qR$. Let χ be β-bounded. For $s \in R_q$, let $A_{s,\chi}$ be the distribution on $R_q \times R_q$ obtained by choosing $a \leftarrow R_q$ uniformly at random, $e \leftarrow \chi$, outputting $(a, a \cdot s + e \mod q)$.*

The RLWE problem is : for uniformly random $s \leftarrow R_q$, given poly(n) number of samples that are either from $A_{s,\chi}$ or uniformly random in $R_q \times R_q$, output 0 if the former holds and 1 if the latter holds.

The $\mathrm{RLWE}_{n,q,\chi}$ assumption says that the $\mathrm{RLWE}_{n,q,\chi}$ problem is infeasible. Denote the assumption by $\mathrm{RLWE}^{(m)}_{n,q,\chi}$ when we require the indistinguishability to hold given only m samples. We state the hardness of the special case of $\mathrm{RLWE}^{(m)}_{n,q,\chi}$ described in [21] as follows.

Theorem 2 ([21]). *For the ring $R = \mathbb{Z}[X]/f(x)$, $f(x) = x^n + 1$, where n is a power of 2, and a prime integer $q = q(n) = 1 \mod 2n$, and $\beta = \omega(\sqrt{n \log n})$, there is an efficiently samplable distribution χ that outputs elements of R with norm at most β with overwhelming probability, such that if there exists an efficient algorithm that solves $\mathrm{RLWE}^{(m)}_{n,q,\chi}$, then there is an efficient quantum algorithm for solving $n^{2.5} \cdot (q/\beta) \cdot (nm/\log(nm))^{1/4}$-approximate worst-case SVP for ideal lattices over R.*

The HNF-RLWE assumption [4] says that the hardness preserves even if we choose the secret from the error distribution, i.e. $s \leftarrow \chi$.

We also extend our "signal" algorithms and key algorithm to the ring case. For any $a = \sum_{i=0}^{n-1} a_i x^i \in R_q$, we extend $\sigma_0(a), \sigma_1(a) : R_q \to R_2$ as follows:

$$\sigma_0(a) = \sum_{i=0}^{n-1} \sigma_0(a_i) x^i;\ \sigma_1(a) = \sum_{i=0}^{n-1} \sigma_1(a_i) x^i.$$

The algorithm S now is defined as $S(a) = \sigma_b(a)$, where $b \xleftarrow{\$} \{0, 1\}$. Similarly, we also extend the robust extractor $E(a, \sigma) : R_q \to R_2$ as follows:

$$E(a, \sigma) = (a + \sigma \cdot \frac{q-1}{2} \mod q) \bmod 2.$$

4.1 Construction

We now describe the key exchange protocol based on RLWE assumption.

- The system first generates the public parameters $q, n, \chi, \beta, R = \mathbb{Z}[x]/f(x)$, where $f(x) = x^n + 1$ and n is a power of 2. Sample a uniformly random element $m \leftarrow R_q$.

- Alice first chooses an secret element $s_A \leftarrow \chi$. Then, Alice computes $p_A = m s_A + 2e_A \mod q$, where $e_A \leftarrow \chi$. Sends p_A to Bob.

- Receiving p_A, Bob first chooses a secret element $s_B \leftarrow \chi$ and an error $e'_B \leftarrow \chi$. Bob then computes $K_B = p_A \cdot s_B + 2e'_B \mod q$ and $\sigma \leftarrow S(K_B)$. He samples $e_B \leftarrow \chi$ and computes $p_B = m \cdot s_B + 2e_B \mod q$. Finally, Bob sends (p_B, σ) to Alice and obtains the shared key $SK_B = E(K_B, \sigma)$.

- Once getting (p_B, σ), Alice samples $e'_A \leftarrow \chi$ and computes $K_A = s_A p_B + 2e'_A \mod q$, and obtains $SK_A = E(K_A, \sigma)$.

Correctness We now show that if Alice and Bob run the protocol honestly, they will share an identical key.

Lemma 9. *If $8n\beta^2 \leq \frac{q}{4} - 2$, then $SK_A = SK_B$ with overwhelming probability.*

Proof. The form of K_A, K_B are as follows. $K_A = ms_As_B + 2(s_Ae_B + e'_A) \mod q$. $K_B = ms_As_B + 2(e_As_B + e'_B) \mod q$. Therefore, we have:

$$K_A = K_B + 2(e_As_B + e'_B - s_Ae_B - e'_A) \bmod (q, x^n + 1).$$

Note that from Lemma 1, we have $\|2(s_Ae_B + e'_A - e_As_B - e'_B)\| \leq 8n\beta^2 \leq \frac{q}{4} - 2$, with overwhelming probability. Therefore, from Lemma 2, we have $SK_A = E(K_A, \sigma) = E(K_b, \sigma) = SK_B$.

Moreover, we can show that $SK_A = SK_B = (ms_As_B + \sigma \cdot \frac{q-1}{2} \bmod (q, x^n + 1)) \mod 2$. The proof is the same as the one in Section 3.1. □

Parameter Selection. A reasonable way to select the parameters is $n = \lambda$, $q = \lambda^4$, $\beta = \lambda$.

Theorem 3. *The construction above is secure against passive PPT adversaries, if the HNF-RLWE assumption holds.*

Proof. The proof is almost the same as in 1, we omit it here. □

Comparisons. Here, we give comparisons between our scheme and other key exchange schme based on public key encryption from RLWE. We use two examples, one is the RLWE-based scheme from Lyubashevsky *et al.* [21], and the other one is the NTRU variant from Stehlé and Steinfeld [27]. Due to the property of RLWE, our scheme can agree on n bit secret key once. We note that the public parameter m can be produced once for all, therefore, it significantly reduce the communication cost. The comparisons are given in Table 2. When comparing to the key transport schemes based on PKE, where the communication and computation cost will be cut to half for encryption based schemes, the efficiency of our scheme is still better than the LPR'10 [21] scheme in communication cost (1/2) but worse in computation cost (4/3); and our scheme is slightly worse than the SS'11 [27] scheme. But we note that the assumption of the SS'11 [27] is much stronger. Therefore, to obtain same security, one needs to increase the security parameter in SS'11 [27], which results much worse efficiency. This means our scheme could still have substantial advantage in terms of practical applications.

Table 2. Comparisons between RLWE-based ones for n bit secret key

	Pub. Param.	Commun. Comp.	Comput. Comp.	Assumption
LPR'10 [21]	$4n \log q$	$8n \log q$	6	$\mathsf{Ideal\text{-}SIVP}_{\tilde{O}(n^3)}$
SS'11 [27]	$2n \log q$	$4n \log q$	4	$\mathsf{Ideal\text{-}SIVP}_{\tilde{O}(n^8)}$
Ours	$n \log q$	$2n \log q + n$	4	$\mathsf{Ideal\text{-}SIVP}_{\tilde{O}(n^{4.5})}$

Pub. Param. means the size of public parameter; Commun. Comp. means the communication complexity; Comput. Comp. means the computation complexity and is estimated by the number of multiplications in the ring R_q.

5 Interactive Multiparty Key Exchange Protocol

In this section, we describe an interactive multiparty key exchange protocol based on RLWE problem. Although the provable security of the protocol seems plausible, we still can not do it, and we leave it as an open problem.

We now describe the interactive multiparty key exchange protocol.

- For a set of k users, the system first generates the public parameters $q, n, \chi, \beta, R = \mathbb{Z}[x]/f(x)$, where $f(x) = x^n + 1$ and n is a power of 2. Samples a uniformly random element $m \leftarrow R_q$.

- For $i \in \{0, ..., k-1\}$, the user i chooses random $s_i \leftarrow \chi$ and $e_i^0 \leftarrow \chi$. Computes $p_i^0 = ms_i + 2e_i^0 \in R_q$, and sends p_i^0 to the user $i+1$. Then for $1 \leq j \leq k-2$, user $i+j \mod k$ computes $p_i^j = s_{i+j \mod k} \cdot p_i^{j-1} + 2e_i^j$, where $e_i^j \in \chi$, and sends p_i^j to the user $i+j+1 \mod k$.

- For the user 0, he or she first chooses $\hat{e}_0 \leftarrow \chi$ and computes $K_0 = p_1^{k-2} \cdot s_0 + 2\hat{e}_0$, then he or she computes $\sigma \leftarrow S(K_0)$. The user 0 obtains the shared key $SK_0 = E(K_0, \sigma)$. Finally, the user 0 broadcasts σ.

- For users $1 \leq i \leq k-1$. they first choose $\hat{e}_i \leftarrow \chi$ and compute $K_i = p_{i+1}^{k-2 \mod k} \cdot s_i + 2\hat{e}_i$ and each obtains the shared secret key $SK_i = E(K_i, \sigma)$.

Correctness We now show that if Alice and Bob run the protocol honestly, they will share an identical key.

Lemma 10. *If $4k \cdot n^k \beta^{k+1} \leq \frac{q}{4} - 2$, then all the k parties share the same secret key with overwhelming probability.*

Proof. The correctness is very similar to the RLWE base two party key exchange protocol. Let's first look at K_0, we can rewrite it in the form: $K_0 = m \prod_{i=0}^{k-1} s_i + \Delta_0$, where:

$$\Delta_0 = s_0 \prod_{i=2}^{k-1} s_i \cdot 2 \cdot e_1^0 + s_0 \prod_{i=3}^{k-1} s_i \cdot 2 \cdot e_1^1 + \cdots + s_0 \cdot s_{k-1} \cdot 2 \cdot e_1^{k-3} + s_0 \cdot 2 \cdot e_1^{k-2} + 2\hat{e}_0.$$

Note that $\|\Delta_0\| \leq k \cdot 2 \cdot n^k \beta^{k+1}$. Similarly, we can compute $K_j = m \prod_{i=0}^{k-1} s_i + \Delta_j$, where $\|\Delta_j\| \leq k \cdot 2 \cdot n^k \beta^{k+1}$ and $1 \leq j \leq k-1$. Notice that since $\|\Delta_0 - \Delta_j\| \leq 4k \cdot n^k \beta^{k+1} \leq \frac{q}{4} - 2$ for $1 \leq j \leq k-1$, we can apply Lemma 2 to finish the remaining part of this lemma. □

6 Conclusion

In this paper, we use the LWE problem to build a new, simple and provably secure key exchange protocol with the help of robust extractors. We show that our scheme have substantial advantages in practical applications when compared with similar scheme derived from the encryption schemes based on the LWE problem. We also extend the construction to the RLWE case. Our construction is a significant additional step in showing how versatile the LWE assumption can be.

The LWE problem itself can be viewed as certain form of inner product with small errors that somehow can be eliminated for certain applications. Our construction can be viewed as an extension of this idea to the case of a bilinear pairing, namely a pairing of bilinear forms with errors. In addition, the reason why the scheme works well actually depends on the associativity and the

commutativity of the multiplications in both the non-commutative rings (the LWE problem) and the commutative rings (the RLWE problem). We believe that exploring further algebraic properties of the non-commutative rings could yield even more interesting cryptographic protocols, such as certain homomorphic properties over non-commutative operations over matrices.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (h) ibe in the standard model. In *EUROCRYPT*, pages 553–572. Springer, 2010.
2. S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. In *PKC*, pages 280–297, 2012.
3. S. Agrawal, D. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *ASIACRYPT*, pages 21–40. Springer, 2011.
4. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618. Springer, 2009.
5. X. Boyen. Attribute-based functional encryption on lattices. In *TCC*, pages 122–142. Springer, 2013.
6. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *CRYPTO*, pages 868–886. Springer, 2012.
7. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.
8. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584. ACM, 2013.
9. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS*, pages 97–106. IEEE, 2011.
10. Z. Brakerski and V. Vaikuntanathan. Lattice-based fhe as secure as pke. In *ITCS*, pages 1–12. ACM, 2014.
11. R. Canetti, D. Dachman-Soled, V. Vaikuntanathan, and H. Wee. Efficient password authenticated key exchange via oblivious transfer. In *PKC*, pages 449–466. Springer, 2012.
12. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *EUROCRYPT*, pages 523–552, 2010.
13. Lily Chen. Practical impact of quantum computing. In *Quantum-Safe-Crypto Workshop, http://docbox.etsi.org/Workshop/2013/201309_CRYPTO/S05_DEPLOYMENT/NIST_CHEN.pdf*. The European Telecommunications Standards Institute, 2013.
14. W. Diffie and M. E. Hellman. New directions in cryptography. *Information Theory*, 22(6):644–654, 1976.
15. A. Fujioka, K. Suzuki, K. Xagawa, and K. Yoneyama. Strongly secure authenticated key exchange from factoring, codes, and lattices. In *PKC*, pages 467–484. Springer, 2012.
16. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. ACM, 2008.
17. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554. ACM, 2013.
18. J. Katz and V. Vaikuntanathan. Smooth projective hashing and password-based authenticated key exchange from lattices. In *ASIACRYPT*, pages 636–652. Springer, 2009.
19. J. Katz and V. Vaikuntanathan. Round-optimal password-based authenticated key exchange. In *TCC*, pages 293–310. Springer, 2011.
20. R. Lindner and C. Peikert. Better key sizes (and attacks) for lwe-based encryption. In *CT-RSA*, pages 319–339, 2011.
21. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.
22. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267, 2007.
23. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342. ACM, 2009.
24. C. Peikert. Lattice cryptography for the internet. Cryptology ePrint Archive, Report 2014/070, 2014.
25. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93. ACM, 2005.
26. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
27. D. Stehlé and R. Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In *EUROCRYPT*, pages 27–47, 2011.