COMPARISON ANALYSIS OF DING’S RLWE-BASED KEY EXCHANGE PROTOCOL AND NEWHOPE VARIANTS

XINWEI GAO

Beijing Key Laboratory of Security and Privacy in Intelligent Transportation
Beijing Jiaotong University
No.3 ShangYuanCun, Haidian District, Beijing 100044, China

(Communicated by Jintai Ding)

Abstract. In this paper, we present a comparison study on three RLWE key exchange protocols: one from Ding et al. in 2012 (DING12) and two from Alkim et al. in 2016 (NewHope and NewHope-Simple). We compare and analyze protocol construction, notion of designing and realizing key exchange, signal computation, error reconciliation and cost of these three protocols. We show that NewHope and NewHope-Simple share very similar notion as DING12 in the sense that NewHope series also send small additional bits with small size (i.e. signal) to assist error reconciliation, where this idea was first practically proposed in DING12. We believe that DING12 is the first work that presented complete LWE & RLWE-based key exchange constructions. The idea of sending additional information in order to realize error reconciliation and key exchange in NewHope and NewHope-Simple remain the same as DING12, despite concrete approaches to compute signal and reconcile error are not the same.

1. Introduction

Diffie and Hellman (DH) proposed the first public key cryptography algorithm in 1976 by introducing a key exchange protocol [9]. DH key exchange allows two parties to agree on same key without any prior knowledge over public (insecure) channel. In DH key exchange, both parties send their own public key to the other side. By taking advantage of the property of commutativity, both parties can compute \((g^a)^b \mod p = (g^b)^a \mod p\) to derive final shared key. It has been over 40 years since Diffie-Hellman key exchange was introduced. This protocol alongside with numerous variants have been deployed in real-world applications for secure communication, e.g. TLS, SSH, IPSec etc. It is known that hardness of Diffie-Hellman key exchange is built on hardness of discrete logarithm problem (DLP), i.e. given \(g^a \mod p\), it is very hard to compute \(a\) when parameters are properly chosen. DLP, together with hard problems like integer factoring problem (IFP), elliptic curve DLP (ECDLP) etc. serve as fundamental hard problems and tools to build current public key cryptosystems. With carefully designed cryptosystem and hard parameter choices, it is very hard to break these cryptosystems with current computers. Other works related to key exchange include [12], [20], [18] etc.

2010 Mathematics Subject Classification: Primary: 94A60, 11T71; Secondary: 14G50.
Key words and phrases: Post-quantum, key exchange, RLWE, error reconciliation, comparison, analysis.
* Corresponding author: Xinwei Gao.
With the advent of quantum computers, cryptographic constructions based on DLP, IFP, ECDLP etc. are no longer considered as secure even with larger and reasonable parameters. In 1994, Peter Shor proposed a quantum algorithm which can efficiently solve such problems on a sufficiently large quantum computer [26]. According to estimation results, a sufficiently large quantum computer can break most currently deployed public key cryptographic algorithms efficiently. In 2015, NSA announced their plan of switching to quantum-resistant cryptography in near future. In 2016, NIST formally called for post-quantum cryptographic algorithms for next generation public key cryptography standards. These developments stress strong and urgent need for post-quantum cryptography algorithms.

There are several approaches to construct post-quantum cryptography primitives: lattice-based, multivariate-based, hash-based, code-based etc. Lattice-based constructions are considered to have very strong hardness, smaller key size and better computational efficiency compared with other constructions. Hardness of most modern lattice-based constructions can be reduced to solving hard problems in lattices (e.g. closest vector problem, shortest vector problem etc.). There are various constructions based on hard problems including Learning with Errors (LWE) problem [25], Ring-LWE problem [19] etc. More importantly, lattice-based cryptography constructions are truly versatile. Public key encryption, key exchange, digital signature, attribute based encryption, pseudorandom function, homomorphic encryption etc. can be built based on LWE/RLWE or other lattice problems. In real-world security protocols (e.g. TLS, SSH, VPN etc.), one important application of public key cryptography is key exchange, since encrypting actual large amount of data is done using symmetric encryption, not public key encryption. Public key encryption is much more costly than symmetric encryption. Therefore, it is of great importance to develop and optimize truly efficient RLWE-based key exchange protocol for real world deployment.

1.1. LWE & RLWE-based key exchange protocols. In 2012, Ding et al. introduced the first LWE and RLWE key exchange protocols with a novel reconciliation mechanism [11] (denoted as DING12). This is the first work which constructs complete analogues of classic Diffie-Hellman key exchange over LWE and RLWE problem practically. Before DING12 was proposed, various attempts on building Diffie-Hellman-like key exchange protocol were made. However, those works presented a general sketch of the protocol, rather than presenting a concrete protocol construction. Some works implied that a DH-like key exchange protocol over LWE/RLWE can be achieved with an efficient error reconciliation mechanism, but no concrete approach on error reconciliation was proposed.

It is known that one technical challenge to construct key exchange over LWE and RLWE is how to agree on same value. Key exchange computation values in LWE/RLWE-based constructions are perturbed by small error terms, therefore they are approximately equal. This is different from rigorously equal values in Diffie-Hellman key exchange. In DING12, they invented “robust extractor” (i.e. error reconciliation mechanism) to extract least significant bit from each coefficient with the help of small additional information (i.e. signal value) to agree on same key. Since both sides can extract same bits using robust extractor with properly chosen parameters, therefore key exchange can be realized. The idea of sending small additional information (signal) to assist error reconciliation is completely new when [11] was published.
Directly following DING12, Peikert proposed a similar RLWE key exchange which fixed a minor bias issue in early version of DING12 in 2014 [22] (denoted as PKT14). In 2015, Bos et al. chose parameters and implemented PKT14 with integration into TLS as post-quantum TLS ciphersuite [8] (denoted as BCNS15). In 2016, Alkim et al. introduced an optimized version of BCNS15 called “NewHope” [6]. They chose more compact parameters, present new error reconciliation mechanism and highly optimized implementation. A detailed comparison analysis on DING12 and BCNS15 can be found in [16]. In late 2016, Alkim et al. introduced a variant of NewHope called “NewHope-Simple” [5]. It is a Key Encapsulation Mechanism (KEM) variant of NewHope in order to “avoid the error-reconciliation mechanism originally proposed by Ding” as [5] wrote. In 2016, Bos et al. presented “Frodo” LWE key exchange which uses modified reconciliation mechanism in DING12 with implementation [7]. DING12, PKT14, BCNS15, NewHope, Frodo construct Diffie-Hellman-like LWE/RLWE key exchange protocols while NewHope-Simple is a KEM-based RLWE key exchange protocol. There are also various works on building Authenticated Key Exchange (AKE), Password-based AKE (PAKE) protocols, applications and implementations, including [27, 10, 17, 15, 14, 13] etc.

1.2. RELATED WORKS. In DING12, the first LWE & RLWE-based key exchange protocols were proposed. They invented a new notion (i.e. reconciliation mechanism) to achieve error cancellation. The notion of sending small additional bits (i.e. signal value) and reconcile error with “robust extractor” (reconciliation function) is new by the time [11] was published. In reconciliation-based key exchange protocols, one party need to send signal values to the other side. In DING12, both sides use robust extractor to mod 2 simultaneously in order to agree on same value (extract least significant bit). They divide $\mathbb{Z}_q$ into inner and outer region equally and signal value is computed according to which region does coefficient lies in. More details are given in following sections.

In NewHope, they introduced a new reconciliation which works in $D_4$ lattice. Unlike DING12 and BCNS15, they present a more geometric approach to generate signal: they divide Voronoi cell of $D_4$ lattice (icositetrachoron, or 24-cells) and use an encoded difference vector between polynomial coefficient vector (consisted of four coefficients of $R_q$) and the center of closest Voronoi cell as signal. For error reconciliation, both parties add signal vector with polynomial coefficient vector and determine generated key bit to be 0 or 1. They claimed that with new reconciliation, NewHope has higher error tolerance. NewHope chooses parameter $n = 1024$, size of signal value is 2048 bits, key length is 256 bits.

In NewHope-Simple, they adopted same idea of using signal value to indicate which region does a coefficient lie in and reconcile errors. NewHope-Simple is a KEM variant of NewHope RLWE key exchange. Despite that authors of NewHope-Simple claim that they do not use reconciliation, the idea of sending small additional bits (i.e. signal value) to assist error reconciliation remain the same as DING12. We will elaborate this in following sections. NewHope-Simple chooses parameter $n = 1024$, size of signal value is 3072 bits, key length is 256 bits.

As RLWE-based key exchange protocol is an important cryptographic primitive towards the post-quantum era, it is important to understand the fundamental design principles of RLWE-based key exchange and major differences between some important works. Since error reconciliation is an important technique to construct RLWE-based key exchange protocol (which is also the major differences between various works), we mainly focus on discussing this technique in this paper. With
such theoretical comparison analysis, we can understand such schemes more clearly and pave the way for future improvements.

1.3. CONTRIBUTION. We present a comparison analysis on three RLWE-based key exchange protocols: DING12, NewHope and NewHope-Simple. We recall and analyze principle of protocol construction, notion to realize key agreement over RLWE problem, signal computation, error reconciliation mechanism, overall cost etc. of these three constructions. We show that DING12 and NewHope are error reconciliation-based Diffie-Hellman-like RLWE key exchange protocols, while NewHope-Simple is a KEM construction with reduced ciphertext size. Similarities and differences between these three protocols are explained and compared. We conclude that despite with some differences, NewHope and NewHope-Simple adopt same notion of sending additional information (i.e. signal value) to assist error reconciliation to realize key exchange, while mechanisms of signal computation and error reconciliation are not the same.

1.4. ORGANIZATION. We recall background knowledge in section 2. In section 3, we analyze and compare DING12, NewHope and NewHope-Simple RLWE key exchange protocol. We analyze and compare protocol construction, signal computation, how error reconciliation is realized and differences between these approaches carefully with intuitive comparison charts. We conclude the paper in section 4.

2. PRELIMINARIES

2.1. RING LEARNING WITH ERRORS (RLWE). RLWE problem was introduced by Lyubashevsky et al. in 2010 [19]. In this work, they presented a ring variant of Learning with Errors (LWE) problem [25]. Define ring $R = \mathbb{Z}[x]/f(x)$, where $f(x) = x^n + 1$ with $n$ be a power of 2. Let $q$ be modulus, quotient ring $R_q = R/qR$. Generally, ring $R$ is cyclotomic ring, $\chi$ is an error distribution. $\chi \leftarrow Z_{\chi}$ denotes a random sampling according to some distribution $\chi$. $D_{\mathbb{Z}_q, \sigma}$ be discrete Gaussian distribution with standard deviation $\sigma$. Define $\|a\|_{\infty} = \max(|a_i|)$, which is the $l_\infty$ norm of $a$. RLWE distribution $A_{s, \chi}$ over $R_q \times R_q$ is defined as uniformly random chosen vector $a \in R_q$, small secret vector $s \in R_q$, error $e \in R_q$ sampled from discrete Gaussian distribution, outputs $(a, b = a \cdot s + e \mod q)$. RLWE has search and decision versions like LWE. Goal of search-RLWE is to recover secret $s$ given multiple RLWE samples. For decision-RLWE, if one can distinguish $b = a \cdot s + e \mod q$ from uniformly random generated ones, then RLWE problem and underlying hard lattice problems can be solved.

In [25], they showed that solving LWE can be reduced to solving hard lattice problems. It is known that LWE-based constructions use large matrix, therefore it is not practical enough compared with RLWE-based ones due to slower computation and larger communication cost. RLWE uses computations and polynomials in ring $R_q$, therefore it is more efficient. LWE and RLWE are known to be quantum resistant since hardness of these problem can be directly reduced to hard problems in lattice (e.g. Shortest Vector Problem (SVP) etc.), which classic and quantum computer cannot solve efficiently. There are quantum [25] and classical [21] reduction between LWE problem in average-case and worst-case hard lattice problems. If there exists a polynomial-time algorithm to solve LWE/RLWE problem, then there exists algorithms to solve hard lattice problems. Currently, there is no known publicly known algorithm that can solve LWE/RLWE problem or underlying lattice problems efficiently on both classic and quantum computers [23]. Hardness of LWE/RLWE
serves as the solid foundation to numerous cryptographic schemes. There are various works that estimate the security of concrete LWE/RLWE instances, including \cite{4, 2, 3} etc. Due to high security and efficiency of RLWE, constructions based on RLWE are regarded as very promising towards post-quantum cryptography.

3. **Comparison analysis of DING12, NewHope and NewHope-Simple**

3.1. **DING12 RLWE key exchange.** Ding et al. introduced LWE and RLWE-based key exchange protocols in 2012 \cite{11}. This work constructs analogues of classic Diffie-Hellman key exchange protocols over LWE and RLWE problem, which are known to be quantum-resistant. These two key exchange protocols are provably secure and directly relying on hardness of LWE and RLWE problem.

An important contribution of DING12 is presenting a reconciliation algorithm that allows two parties to agree on same value over error perturbed and approximately equal values. DING12 is the first work to present complete key exchange protocols over LWE and RLWE. Some previous works only give overall structure of “approximate key exchange”, claiming that Diffie-Hellman-like key exchange over LWE/RLWE can be realized, but they did not give concrete algorithms on how to reconcile error and generate same key bits for two parties. The idea of sending small additional bits (i.e. signal) is new by the time \cite{11} was published. Moreover, error reconciliation is done by mod 2 for both sides simultaneously, which is very efficient.

Figure 1 illustrates DING12 RLWE key exchange protocol.

![Diagram](image)

**Figure 1. DING12 RLWE key exchange protocol**

Element \(a, s, e\) in Figure 1 are polynomials in \(R_q\), where \(a\) is an uniformly random public parameter, \(s\) and \(e\) are private polynomials sampled from \(D_{Z^n, \sigma}\). Note that in DING12, they represent \(Z_q\) as \(\{-q^{-1}, \cdots, q^{-1}\}\). We will follow this notation throughout this paper when analyzing DING12 protocol.

**Signal function**. A signal function \(\text{Sig}\) is defined as:

\[
\text{Sig}(y) = \begin{cases} 0, & x \in [-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor], \\ 1, & \text{otherwise} \end{cases}
\]

**Hint functions**. Hint functions \(\sigma_0(x), \sigma_1(x)\) from \(Z_q\) to \(\{0, 1\}\) are defined as:

\[
\sigma_0(x) = \begin{cases} 0, & x \in [-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor], \\ 1, & \text{otherwise} \end{cases}, \quad \sigma_1(x) = \begin{cases} 0, & x \in [-\lfloor \frac{q}{2} \rfloor + 1, \lfloor \frac{q}{2} \rfloor + 1], \\ 1, & \text{otherwise} \end{cases}
\]

For any \(y \in Z_q\), \(\text{Sig}(y) = \sigma_b(y)\), where \(b \overset{\$}{\leftarrow} \{0, 1\}\). We denote that \(y\) is in the outer region if \(\text{Sig}(y) = 1\), otherwise \(y\) is in the inner region.
Signal function for \( a \in R_q \) is computed by applying \( \text{Sig}() \) for each coefficient \( a_i \in \mathbb{Z}_q \).

**Reconciliation function.** \( \text{Mod}_2() \) is a deterministic function with error tolerance \( \delta \). \( \text{Mod}_2() \) is defined as: for any \( x \in \mathbb{Z}_q \) and \( w = \text{Sig}(x) \), \( \text{Mod}_2(x, w) = (x + w \cdot \frac{q-1}{2} \mod q) \mod 2 \).

We define the error tolerance \( \delta \), as the largest integer such that for any \( x, y \in \mathbb{Z}_q \), if \( \|x - y\|_\infty \leq \delta \), then \( \text{Mod}_2(x, w) = \text{Mod}_2(y, w) \), where \( w = \text{Sig}(y) \). Error tolerance \( \delta \) of error reconciliation mechanism in DING12 is \( \frac{q}{4} - 2 \), which is the key to ensure correctness of key exchange over RLWE with overwhelming probability.

Reconciliation function is defined for an integer \( x \in \mathbb{Z}_q \). Reconciliation function for \( a \in R_q \) is computed by applying \( \text{Mod}_2() \) for each coefficient \( a_i \in \mathbb{Z}_q \). We use the same notation “\( \text{Mod}_2() \)” for both reconciliation functions over \( \mathbb{Z}_q \) and \( R_q \).

In DING12 key exchange, party \( j \) computes and sends signal to the other party. Both parties utilize the error reconciliation function \( \text{Mod}_2() \) to agree on an identical value, i.e. same key bits. Let \( q > 8 \) be an odd integer, function \( \text{Mod}_2() \) defined above is a robust extractor with error tolerance \( \delta = \frac{q}{4} - 2 \). They also proved that for any odd \( q > 2 \), if \( x \) is uniformly random in \( \mathbb{Z}_q \), then \( \text{Mod}_2(x, w) \) is uniformly random conditioned on \( w \), where \( w \leftarrow \text{Sig}(x) \). For detailed proofs, please refer to [11].

Signal and shared key size are estimated to be 1024 bits for parameter choice \( n = 1024 \) respectively. Note that in [11], they did not specify concrete parameter choice. Here we only choose \( n = 1024 \) since this is a rather common and secure choice for RLWE instance. We do not specify other parameters, including error distribution, modulus \( q \) etc. For the rest of this paper, we fix \( n = 1024 \) for DING12.

We believe that the structure of key exchange protocol and the notion of sending small additional bits (i.e. signal) to assist error reconciliation process are very efficient. The first work to propose complete solution is DING12 [11]. The idea of sending small additional bits to assist error reconciliation has been applied in various RLWE key exchange protocols, including PKT14 [22], BCNS15 [8], NewHope [6] and NewHope-Simple [5]. In BCNS15 [8], they modify reconciliation in DING12 slightly to extract most significant bit, instead of least significant bit in DING12. A detailed comparison study on DING12 and BCNS15 is given in [16]. This work also shows the potential of truly efficient implementation of DING12. In NewHope [6], they adopt the notion of sending signal value to reconcile errors, but they use a geometric approach to compute signal. In NewHope-Simple [5], they construct a KEM which also uses the idea of sending signal value to assist error reconciliation.

### 3.2. NewHope RLWE Key Exchange

NewHope RLWE key exchange protocol was proposed by Alkim et al. [6]. It improves BCNS15 construction with several new techniques, including: a different error reconciliation mechanism, more compact parameter choice, choosing alternative sampling algorithm for higher efficiency etc. This work claims that with all above approaches, their portable C implementation is 9x faster than BCNS15 and AVX2 implementation is 24x faster. Structure of NewHope remains the same as DING12 and BCNS15.

Figure 2 illustrates NewHope RLWE key exchange protocol. Functions and variables highlighted in red color are the major differences compared with DING12.

NewHope uses a different approach to reconcile errors. This idea is similar as DING12 and BCNS15 in the sense that one party need to send signal to the other...
side to assist error reconciliation, but NewHope adopts a geometric way (i.e. computation in $D_4$ lattice) to achieve this. The $D_4$ lattice is a rotated and scaled version of the root lattice $D_4$. The Voronoi cells of $\tilde{D}_4$ lattice are icositetrachoron, or known as 24-cells [1]. They extend technique in [24] to encode information from four coordinates of the key computation material $k_j$ into one bit, therefore a 256-bit key is derived by both parties. [24] is originally designed to encode 2 coordinates into one bit and NewHope extended this technique to encode 4 coordinates into one bit.

In NewHope, signal value is an encoded difference vector between coefficient vector and center of its closest Voronoi cell. One party computes signal vector, encodes it and sends to the other party so that error can be reconciled. The other party first decodes signal value to a vector, then adds this difference vector to its corresponding coefficient vector. Figure 3 illustrates NewHope reconciliation mechanism in $d = 2$ case, which is easier to demonstrate and understand (i.e. in $D_2$ lattice). $d = 4$ case is used by NewHope but more complex. Since it is an direct extension of $d = 2$ case and shares similar idea, we demonstrate $d = 2$ case for simplicity. If the sum of two vectors is in the grey Voronoi cell, key bit is 1, otherwise key bit is 0. Possible range of sum is the square with vertex (0, 0), (0, 1), (1, 0), (1, 1).

NewHope takes four coefficients in $k_i$ (or $k_j$) and forms vector $k'_i$ (or $k'_j$). The notion of NewHope reconciliation is to compute difference between vector $k'_i$ and nearest Voronoi cell as signal value. For reconciliation, both parties add signal vector with $k'_i$ (or $k'_j$) so that sum vector gets close to the nearest Voronoi cell. Region in $\tilde{D}_4$ is divided evenly so that key bits are generated uniformly random. Signal function of NewHope is defined as: $\text{HelpRec}(x, b) = \text{CVP}_{\tilde{D}_4}(\frac{x}{r} + b\mathbf{g}) \mod 2^r$, reconciliation function is defined as $\text{Rec}(x, r) = \text{Decode}(\frac{1}{q}x - \frac{1}{2q}\mathbf{Br})$ where $r = 2$. In lemma C.3 of [6], they claimed that if $\|k'_i - k'_j\| < \frac{2r}{4} - 2$, then same key bit is generated with overwhelming probability.

Signal and shared key size are estimated to be 2048 bits and 256 bits respectively, since NewHope’s reconciliation mechanism generates $r \cdot d = 2 \cdot 4 = 8$ bits signal vector per key bit. Note that HelpRec function uses mod $2^r$ computation where $r = 2$ for NewHope, therefore it generates 2-bit signal for each coefficient of $R_q$. In comparison, DING12 ($n = 1024$) generates 1024-bit signal and 1024-bit key.
3.3. **NewHope-Simple RLWE Key Exchange.** Alkim et al. proposed a simplified version of NewHope RLWE key exchange in [5] (denoted as “NewHope-Simple”). Compare with NewHope, NewHope-Simple shares most components of NewHope’s protocol construction, but it uses different approaches to compute signal value and error reconciliation mechanism. NewHope-Simple is a KEM-based key agreement protocol where one party decides then encrypts the key, unlike DING12 and NewHope which are Diffie-Hellman-like and both parties decide the key together. Compared with other KEM constructions, NewHope-Simple uses signal value to reduce communication cost. Authors of NewHope-Simple claim that it is a non-reconciliation version of NewHope.

Figure 4 illustrates NewHope-Simple RLWE key exchange protocol. Functions and variables highlighted in red color are the major differences compared with DING12.

**Figure 3.** Illustrated figure of simplified NewHope error reconciliation ($d = 2$)

**Figure 4.** NewHope-Simple RLWE key exchange protocol
For correctness of NewHope-Simple, since NHSEncode() and NHSDecode() works on a 4-dimension chunk, they claimed that for rounding error \( r \in \mathbb{Z}^4 \), \( \| r \|_1 \leq q/4 + 4 \). Failure probability is estimated as \( 2^{-60} \) using same parameters and estimation approach as NewHope.

Major differences include: hashing uniform random bits as encapsulated key, adding/subtracting encoded encapsulated key etc. We believe that hashing \( v \) to \( v' \) is not necessary in terms of security but making the scheme more complex. Compared with the simple and elegant error reconciliation mechanism in DING12, encryption/decryption computations are more complex than mod 2 simultaneously while giving no additional advantages. Moreover, signal \( \tau \) is 3x larger than DING12. We believe that approaches to realize KEM in NewHope-Simple give additional unnecessary overheads compared with simple and efficient design in DING12.

Variable \( \tau \) in NewHope-Simple is the signal value. How to compute \( \tau \) and reconcile error are very different from NewHope. Signal value \( \tau \) is computed using NHSCompress() function and we recall it as follows: for each coefficient in \( k_j \) (a polynomial in \( R_q \) with degree \( n = 1024 \)), \( \tau_i = \lfloor (k_{ji} \cdot 8)/q \rfloor \mod 8 \) with \( i \) from 0 to 1023. \( \tau \) is an indicator for \( k_{ji} \), implying that which region does each coefficient of \( k_j \) lie in, and signal \( \tau \) is sent to the other party to reconcile the errors, which shares same idea as DING12 [11] and NewHope [6], despite that how to compute signal \( \tau \) and error reconciliation are different.

Figure 5 illustrates signal value of NewHope-Simple and DING12.

Error reconciliation mechanism of NewHope-Simple is relative similar with RLWE-based encryption, but one part is different: unlike early versions of RLWE decryption algorithms, Alice computes \( k_j' = c' - p_j s_i \), then get 0/1 key bit using NHSDecode() function. NHSDecode() function is a variant of RLWE decryption algorithm, which uses sum of absolute value of difference between four coefficients and \( \lfloor q/2 \rfloor \) to recover encapsulated key bit. If the sum is smaller than \( q \), then key bit is 1, otherwise 0.

Signal size of NewHope-Simple is \( 3 \cdot 1024 = 3072 \) bits since each coefficient gives 3-bit signal value, which is larger than 1024-bit signal of DING12 and 2048-bit signal of NewHope. Final shared key size is 256 bits, which is same as NewHope but shorter than DING12 (1024-bit).
3.4. **Summary.** As our analysis shows, we can see that general structure of DING12, NewHope and NewHope-Simple are very similar. To be more precise, all three protocols take advantage of: (1) Properties of RLWE problem and commutativity; (2) Sending small additional information (i.e. signal value) to assist error reconciliation; (3) Use certain algorithm to reconcile errors and derive final shared key. The most important part to realize key exchange over RLWE problem is how to reconcile errors, where DING12 is the first to propose comprehensive and practical solution. NewHope and NewHope-Simple follow same idea of sending signal value from one side to the other to assist error reconciliation. Despite algorithms on computing signal and error reconciliation are not the same for these three works, the notion of sending signal remain the same.

For signal computation, DING12 generates 1-bit signal for each coefficient in $R_q$ using function `Sig()`. NewHope generates 8-bit signal for four coefficients in $R_q$ using function `HelpRec()`. NewHope-Simple generates 3-bit signal for each coefficient in $R_q$ using function `NHSCompress()`. As Figure 5 shows, it is clear that DING12, NewHope and NewHope-Simple are very similar in the sense that all three protocols send small additional information to the other side for error reconciliation. As for NewHope, despite it is more complex than above two constructions, the idea remain exactly the same. Four coefficients generates 8-bit signal value.

For error reconciliation, DING12 simply mod 2 for both sides with the help of signal value (i.e. `Mod2()` function) to extract least significant bit of each coefficient as key bit. A 1024-bit key is derived. NewHope adopts a different reconciliation mechanism in $D_4$ lattice in a geometric approach, which uses an encoded vector as signal value. This vector incidates how the error can be reconciled, therefore both parties can agree on same 256-bit key with the help of signal value using `Rec()` function. NewHope-Simple generates a 3-bit signal for each coefficient for a variant of basic RLWE encryption/decryption computation with reduced ciphertext size. It encrypts a randomly generated key using `NHSEncode()` function and decrypts the encapsulated key bits using `NHSDecode()` function. As a KEM construction, NewHope-Simple decrypts the 256-bit key in a slightly different approach compared with old-fasion RLWE decryption.

To summarize, NewHope and NewHope-Simple share same idea as DING12 in the sense that these two constructions also need to send small additional bits (i.e. signal) to assist error reconciliation, which is first practically proposed in DING12. We summarize the differences on signal computation, error reconciliation and their sizes in Table 1. In addition, despite the comparison between DING12 and BCNS15 (PKT14) has been detailed in [16], we will add BCNS15 (PKT14) in Table 1 as well. Note that [11] did not specify concrete parameter choice for their protocol. Here we choose $n = 1024$ since this is considered as a safe choice for RLWE-based key exchange and directly comparable with other works.

4. **Conclusion**

In this paper, we discuss and compare core notions to realize key exchange over RLWE problem: protocol construction using commutativity, signal computation, error reconciliation etc. in DING12, NewHope and NewHope-Simple RLWE key exchange protocols. In order to reconcile errors, an important notion is to send additional information so that both parties can get advantage of such information and agree on same key. This notion is first proposed practically by Ding et al. in 2012 and utilized to construct first complete LWE & RLWE key exchange protocols.
Comparison Analysis

**Signal computation**
- Divide $\mathbb{Z}_q$ into 2 regions.
- Different region gives corresponding value.
- Signal value $\in \{0, 1\}$.
- Function: $\langle \cdot \rangle_q^2$.

**Error reconciliation**
- Add then mod 2 on each coefficient.
- Extract least significant bit.
- Function: $\text{Mod}_2()$.

**Degree $n$ of $R_q$**
- 1-bit per coefficient.
- 1024 bits.
- Function: $\text{Rec()}$.

---

**DING12**
- Divide $\mathbb{Z}_q$ into 2 regions.
- Different region gives corresponding value.
- Error reconciliation.
- Signal value $\in \{0, 1, 2, 3\}$.
- Function: $\text{HelpRec()}$.

**BCNS15**
- Divide $\mathbb{Z}_q$ into 2 regions.
- Different region gives corresponding value.
- Error reconciliation.
- Signal value $\in \{0, 1\}$.
- Function: $\text{HelmRec()}$.

**NewHope**
- Divide $\mathbb{Z}_q$ into 8 regions.
- Different region gives corresponding value.
- Error reconciliation.
- Signal value $\in \{0, \ldots, 7\}$.
- Function: $\text{NHSCompress()}$.

**NewHope-Simple**
- Divide $\mathbb{Z}_q$ into 2 regions.
- Different region gives corresponding value.
- Error reconciliation.
- Signal value $\in \{0, 1\}$.
- Function: $\text{NHSDecomp}()$.

---

**Table 1. Summary Chart of DING12, BCNS15, NewHope, NewHope-Simple**

| Category          | Signal size | Key size | Degree $n$ of $R_q$ | Error reconciliation | Function: $\text{Rec}()$ | Function: $\text{NHSDecomp}()$ |
|-------------------|-------------|----------|---------------------|----------------------|--------------------------|-------------------------------|
| Diffie-Hellman-like | 1-bit per coefficient | $n$ bits | 1024 | 1-bit per coefficient | $\text{Rec}()$ | $\text{NHSDecomp}()$ |
| Diffie-Hellman-like | 2-bit per coefficient | $2n$ bits | 2048 | 2-bit per coefficient | $\text{Rec}()$ | $\text{NHSDecomp}()$ |
| Diffie-Hellman-like | 3-bit per coefficient | $3n$ bits | 3072 | 3-bit per coefficient | $\text{Rec}()$ | $\text{NHSDecomp}()$ |
| KEM (Encryption)  | 3-bit per coefficient | $3n$ bits | 3072 | 3-bit per coefficient | $\text{Rec}()$ | $\text{NHSDecomp}()$ |
We show that two recent follow-up works: NewHope and NewHope-Simple share
same notion to realize key exchange, despite the approaches to compute signal and
reconcile errors are not the same.

ACKNOWLEDGMENT

This work is funded by China Scholarship Council and Fundamental Research
Funds for the Central Universities (Grant No. 2017YJS038) and National Natural
Science Foundation of China (Grant No. 61672092).

REFERENCES

[1] 24-cell, Page Version ID: 822760596. https://en.wikipedia.org/w/index.php?title=
24-cell&oldid=822760596
[2] M. R. Albrecht, On dual lattice attacks against small-secret lwe and parameter choices in
helib and seal, in Annual International Conference on the Theory and Applications of Crypt-
ographic Techniques, Springer, 10211 (2017), 103–129.
[3] M. R. Albrecht, F. Göpfert, F. Virdia and T. Wunderer, Revisiting the expected cost of solving
usvp and applications to lwe, in International Conference on the Theory and Application of
Cryptography and Information Security, Springer, 10624 (2017), 297–322.
[4] M. R. Albrecht, R. Player and S. Scott, On the concrete hardness of learning with errors,
Journal of Mathematical Cryptology, 9 (2015), 169–203.
[5] E. Alkim, L. Ducas, T. Pöppelmann and P. Schwabe, Newhope without reconciliation IACR
Cryptology ePrint Archive, 2016 (2016), 1157.
[6] E. Alkim, L. Ducas, T. Pöppelmann and P. Schwabe, Post-quantum key exchange-a new
hope, in USENIX Security Symposium, 2016, 327–343.
[7] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan and
D. Stebila, FRODO: Take off the ring! practical, quantum-secure key exchange from lwe, in
Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications
Security, ACM, 2016, 1006–1018.
[8] J. W. Bos, C. Costello, M. Naehrig and D. Stebila, Post-quantum key exchange for the tls
protocol from the ring learning with errors problem, in Security and Privacy (SP), 2015 IEEE
Symposium on, IEEE, 2015, 553–570.
[9] W. Diffie and M. Hellman, New directions in cryptography, IEEE Transactions on Information
Theory, 22 (1976), 644–654.
[10] J. Ding, S. Alsayigh, J. Lancrenon, S. RV and M. Snook, Provably secure password authen-
ticated key exchange based on rlwe for the post-quantum world, in Cryptographers Track at
the RSA Conference, Springer, 10159 (2017), 183–204.
[11] 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, 2012 (2012), 688.
[12] M. S. Dousti and R. Jalili, Forsakes: A forward-secure authenticated key exchange protocol
based on symmetric key-evolving schemes, Advances in Mathematics of Communications, 9
(2015), 471–514.
[13] X. Gao, J. Ding, L. Li and J. Liu, Practical randomized rlwe-based key exchange against
signal leakage attack, IEEE Transactions on Computers, 67 (2018), 1584–1593.
[14] X. Gao, J. Ding, L. Li, S. RV and J. Liu, Efficient implementation of password-based authen-
ticated key exchange from rlwe and post-quantum tls, International Journal of Network
Security, 20 (2018), 923–930.
[15] X. Gao, J. Ding, J. Liu and L. Li, Post-quantum secure remote password protocol from
rlwe problem, in International Conference on Information Security and Cryptology, Springer,
10726 (2017), 99–116.
[16] X. Gao, J. Ding, S. RV, L. Li and J. Liu, Comparison analysis and efficient implementation of
reconciliation-based rlwe key exchange protocol, IACR Cryptology ePrint Archive, 2017
(2017), 1178.
[17] X. Gao, L. Li, J. Ding, J. Liu, S. RV and Z. Liu, Fast discretized gaussian sampling and
post-quantum tls ciphersuite, in International Conference on Information Security Practice
and Experience, Springer, 2017, 551–565.
S. González, L. Huguet, C. Martínez and H. Villaña, Discrete logarithm like problems and linear recurring sequences, **Advances in Mathematics of Communications**, 7 (2013), 187–195.

V. Lyubashevsky, C. Peikert and O. Regev, On ideal lattices and learning with errors over rings, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 6110 (2010), 1–23.

G. Micheli, Cryptanalysis of a noncommutative key exchange protocol, **Advances in Mathematics of Communications**, 9 (2015), 247–253.

C. Peikert, Public-key cryptosystems from the worst-case shortest vector problem, in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, ACM, 2009, 333–342.

C. Peikert, Lattice cryptography for the internet, in *International Workshop on Post-Quantum Cryptography*, Springer, 8772 (2014), 197–219.

C. Peikert et al., A decade of lattice cryptography, *Foundations and Trends® in Theoretical Computer Science*, 10 (2014), 283–424.

T. Pöppelmann and T. Güneysu, Towards practical lattice-based public-key encryption on reconfigurable hardware, in *International Conference on Selected Areas in Cryptography*, Springer, 2013, 68–85.

O. Regev, On lattices, learning with errors, random linear codes, and cryptography, *Journal of the ACM (JACM)*, 56 (2009), Art. 34, 40 pp.

P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review*, 41 (1999), 303–332.

J. Zhang, Z. Zhang, J. Ding, M. Snook and Ö. Dagdelen, Authenticated key exchange from ideal lattices, in *EUROCRYPT (2)*, 9057 (2015), 719–751.

Received January 2018; revised May 2018.

E-mail address: xinwei.gao.7@yandex.com