Lattice Based Group Key Exchange Protocol in the Standard Model

Parhat Abla

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing China,
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing China

Abstract. Group key exchange schemes allow group members to agree on a session key. Although there are many works on constructing group key exchange schemes, but most of them are based on algebraic problems which can be solved by quantum algorithms in polynomial time. Even if several works considered lattice based group key exchange schemes, believed to be post-quantum secure, but only in the random oracle model.
In this work, we propose a group key exchange scheme based on ring learning with errors problem. On contrast to existing schemes, our scheme is proved to be secure in the standard model. To achieve this, we define and instantiate multi-party key reconciliation mechanism. Furthermore, using known compiler with lattice based signature schemes, we can achieve authenticated group key exchange with post-quantum security.

Keywords: Group key exchange; Lattice; Ring LWE,

1 Introduction

Cryptographic key exchange protocols can establish a "secure channel" among the participants, connected by insecure communication networks, by enabling them agree on a session key. Through this channel, participants can transmit sensitive data or apply other higher-level cryptographic schemes. The confidentiality of this channel usually can be reduced to the security of the cryptographic protocols.

Since Diffie-Hellman's two party key exchange protocol [13], the work of [22, 24] are focus on designing two-party protocols based on various hard problems and improving their efficiency. There are many works [4, 8–11] on considering the multi-party scenario. However, aforementioned protocols's security are based on classically hard problems which can be solved in polynomial time by quantum algorithms [29].

Since the quantum resistance of lattice problems, especially the hardness of LWE problems [7, 23, 25, 27, 28], most of the recent works [1, 6, 14, 18, 19, 26, 30]

David C. Wyld et al. (Eds): CMC, NCO, SOFT, CDKP, MLT, ICAITA - 2021
pp. 157-174, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111113

mainly focus on designing and improving the quality of lattice based two party protocols. On the other hand, only a few works [2, 14] focus on designing lattice based group key exchange(GKE) protocols, but they have their own drawbacks as we show next.

The work of [14] is the first try of constructing lattice based key exchange scheme, but it is lack of standard security proof. Even if the work of [2] proposed a constant round GKE protocols base on plain LWE problem. But, their protocols only proven secure in the random oracle model [3], which usually replaced by cryptographic hash functions in a real world applications. But there exist cryptographic schemes that are secure in the Random Oracle Model, but for which any implementation yields insecure schemes [12]. Therefore, security in the standard model is more plausible for the cryptographic schemes. To our knowledge, designing and modular analyzing of a group key exchange protocols in the standard model are not considered yet. Even if a GKE scheme can be obtained by a two-party key exchange scheme, but this approach believed to be very impractical, hence we consider the direct construction.

1.1 Our Contributions

In this work, we analyze and construct a multi-party group key exchange protocol. As shown in the work of [18], the key reconciliation mechanism(KRM) is necessary for a LWE based key exchange protocols. Therefore, we first introduce the concept of multi-party KRM and show it's concrete instantiation. Our definition of the multi-party KRM can be regarded as the generalization of the two-party KRM [14, 19, 26]. In a multi-party key reconciliation mechanism, each party should own predetermined informations to ensure that each party have a same element after running the multi-party KRM. Its not hard to see that multi-party KRM is not enough to get a group key exchange protocol because the correctness of KRM need the pre-determined value(input to the KRM) satisfy some proper constraints. Therefore, this is why we need other additional tools to get GKE. For the security, we require that KRM's output should random even if the transactions are exposed.

To instantiate, we designe a new multi-party key reconciliation mechanism. Compared to a naive generalization of the two-party case [14], our instantiation can be applied to both for odd and even modulus. Meanwhile, the previous key reconciliation mechanism of [14] only fits for the odd modulus. Furthermore, our design is as efficient as [14] in the two-party settings. Roughly, we have following result.

Theorem 1.1 (informal) *For the integers p, q, g such that $p < q$, $q > p(g + 1)$ and $gcd(q, g) = 1$, there exist a multi-party KRM that is secure and correct.*

Additionally, we introduce a weaker version of GKE. In contrast to GKE, a weak GKE only enables the participants to agree on some approximately the same element. Obviously, any GKE protocol is also a weak GKE, but the reverse is not the case. Therefore, constructing this weak definition of GKE at most as hard as constructing a general GKE.

The correctness of a weak GKE is similar to the case of GKE except the final output of a weak GKE should belongs to some range with overwhelming probability. But the security of weak GKE is a crux. In a GKE protocol, there is no difference between the following two cases:(1) the adversary is given one key of the parties, and (2) the adversary is given all parties's keys. This is the case in a GKE, since the correctness of GKE guarantees that all parties's keys are equal. But this is not the case in the weak GKE. Because in a weak GKE, the participants will obtain an approximately the same keys. However, we define passive security of weak GKE and present the our instantiation of weak GKE for sake of such weak GKE's existence.

Finally, we construct a GKE in the standard model. Roughly speaking, we show that a secure multi-party KRM and a secure weak GKE implies passively secure GKE. Intuitively, the weak GKE ensures that each party have approximately the same element, and then applying the multi-party KRM, each party will agree on a same session key. The correctness of corresponding GKE can be reduced to the correctness of the KRM and the weak GKE. The security analysis is more subtle, and we elaborate it in section 5. Additionally, combining the previous instantiations, we show the instantiation of the GKE.

1.2 Related Works and Comparison

There are sequence of works [1, 6, 14, 19, 26, 30] are working on designing and improving the two-party KRM and authenticated key exchange schemes from lattices.

Even if the works of [14, 19, 26] designed KRM, but they only focused on two-party case. Our KRM design is applicable for both two party and multi-party case. Variants of above designs are submitted to the NIST post-quantum cryptography competition. But they mainly focused on designing KEMs, and then designed two-party key exchange protocols through this KEMs. Obviously, this approach seem to be more centralized and heavily rely on one party. Hence we didn't consider this research line in designing multi-party case.

Apon et.al [2] proposed constant round lattice based GKE, but their scheme only proven secure in the random oracle model. In contrast, our GKE protocol is proven secure in standard model which is a more plausible security for a cryptographic scheme.

Organizations In section 2 we present basic notations, definitions and some useful results from literatures. In section 3 we introduce multi-party KRM and its concrete instantiation. We define and instantiate a waker version of group key exchange protocol in section 4. finally we construct a secure group key exchange protocol in the standard model.

2 Preliminaries

Notations For a real $x \in \mathbb{R}$, denote the largest integer which smaller than x by $\lfloor x \rfloor$. For any natural integer $n \in \mathbb{N}$, the symbol $[n]$ denotes the index set $\{0, 1, \cdots, n-1\}$. For any positive integer q, let $\mathbb{Z}_q$ be the cyclic group $\{0, 1, 2, ..., q-1\}$ with addition modulo q. For any reals a, b, c such that $a \leq b$, the shifted set $c + (a, b)$ denotes the interval $(a + c, b + c)$. We abuse the notions for the half closed and closed intervals in $\mathbb{Z}_q$ in a similar way. For any two elements $x, y \in \mathbb{Z}_q$, we let $|x - y|$ be the value of $\min_{k \in \mathbb{Z}} |x - y + kq|$. Vectors are denoted with bold lower-case letters(e.g., $\boldsymbol{a}$). For any set S and $n \in \mathbb{N}$, the set of n-dimensional vectors with entries in S is denoted by S^n, and the set of n-by-m matrices with entries in S is denoted by $S^{n \times m}$. For any probability distribution χ with probability space Ω, the notion $x \overset{\chi}{\leftarrow} \Omega$ mean that x is sampled from Ω according to χ. If the probability space is clear from the context, we simplify the notion as $x \leftarrow \chi$. If χ is uniform distribution, we omit it for the sake of simplicity, e.g., $x \leftarrow \Omega$. We say a function $\epsilon(\lambda)$ is negligible if $\frac{1}{\epsilon(\lambda)}$ is larger than all polynomial $\mathsf{poly}(\lambda)$ from some point λ_0.

2.1 Group Key Exchange Protocol

In this section, we recall the concepts relevant to group key exchange and key reconciliation mechanisms.

GKE: A Group key exchange protocol enables the participated parties agree on a random session key. During the process, participants may run different scripts, but after all interaction and calculation processes, they will agree on a same session key. The security of GKE require that the agreed session key is indistinguishable from an equal-length random string. Here we recall the definition, correctness, and security of GKE as follow:

Definition 2.1 *A Group key exchange protocol* GKE *consists of three algorithms* $(\mathsf{GKE.Setup}, \mathsf{Interact}, \mathsf{KeyGen})$ *as follow:*

- $\mathsf{GKE.Setup}(1^\lambda, 1^N) \to \mathsf{pp}$: *On input the security parameter* λ *and number of participants* N*, it outputs a general public parameter* pp*.*
- $\mathsf{Interact}(P_i, \mathsf{pp})_{i \in [N]} \to \{\mathsf{trans}_i, \mathsf{st}_i\}_{i \in [N]}$: *After receiving the public parameter* pp*, each party* P_i *run its own script which calculate, receive, and broadcast data transmitted through public tunnel. Use* trans_i *to denote the data sets received and sent by* P_i*, and denote the after all state of* P_i *by* st_i*.*
- $\mathsf{KeyGen}(\mathsf{pp}, P_i, \{\mathsf{trans}_i, \mathsf{st}_i\})_{i \in [N]} = \{K_i\}_{i \in [N]}$: *On input public parameter* pp*, transaction* trans_i*, party* P_i *computes its own session key* K_i*.*

Definition 2.2 (Correctness.) *We say a* GKE *is correct if for some random string* K*, the probability*

$$\mathsf{Pr}\left[\bigwedge_{i\in[N]} K_i = K \middle| \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{GKE.Setup}(1^\lambda, 1^N) \\ \{\mathsf{trans}_i, \mathsf{st}_i\}_{i\in[N]} \leftarrow \mathsf{Interact}(P_i, \mathsf{pp})_{i\in[N]} \\ \{K_i\}_{i\in[N]} := \mathsf{KeyGen}\,(\mathsf{pp}, P_i, \{\mathsf{trans}_i, \mathsf{st}_i\})_{i\in[N]} \end{array}\right]$$

is $\mathsf{negl}(\lambda)$*, where probability is taken over the randomness of* KeyGen *algorithm and randomness of* $\mathsf{Interact}$ *algorithm.*

For a probabilistic polynomial time algorithm $\mathcal{A}$ and key space $\mathcal{K}$ of GKE, we define the advantage of $\mathcal{A}$ against GKE, denoted $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{GKE}}$, as

$$\mathsf{Pr}\left[b' = b \middle| \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{GKE.Setup}(1^\lambda, 1^N) \\ \{\mathsf{trans}_i, \mathsf{st}_i\}_{i\in[N]} \leftarrow \mathsf{Interact}(P_i, \mathsf{pp})_{i\in[N]} \\ \{K_i\}_{i\in[N]} := \mathsf{KeyGen}(\mathsf{pp}, P_i, \{\mathsf{trans}_i, \mathsf{st}_i\})_{i\in[N]} \\ b \xleftarrow{\$} \{0,1\},\ \text{if } b = 0, K^* :\xleftarrow{\$} \mathcal{K},\ \text{else } K^* := K_0 \\ b' \leftarrow \mathcal{A}(\{\mathsf{trans}_i\}_{i\in[N]}, \mathsf{pp}, K^*) \end{array}\right] - \frac{1}{2},$$

where the probability is taken over the randomness of KeyGen algorithm, randomness of $\mathsf{Interact}$ algorithm and random coin toss of b. we define the eavesdropper(passive) security of a GKE as follows.

Definition 2.3 (Security.) *We say protocol* GKE *is passively secure if the advantage* Adv *of any* PPT *algorithm* $\mathcal{A}$*(eavesdropper) is negligible in the security parameter* λ*, i.e.,* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{GKE}}(\lambda) \leq \mathsf{negl}(\lambda)$*.*

If a GKE protocol remain secure in a case where the adversary capable of completely controlling over all the communications in the network, We say GKE is adaptively secure. Fortunately, there is a compiler [21] transforms a passively secure GKE into an adaptive one. Note that this compiler need a secure signature scheme. Fortunately, there are lattice based signature schemes [15–17] which are strongly unforgeable under adaptive chosen message attack (EUF-CMA), and it's enough for the compiler. In other words, if there is a lattice based GKE, then we have a lattice based athenticated GKE. Hence in this work, we mainly focus on constructing GKE.

2.2 Gaussians and Ring LWE

Here, we recall definitions and some useful results of gaussian distributions and ring Learning With Errors(LWE) problems.

Lattice and Gaussian. A n-dimensional lattice L is the discrete subgroup of $\mathbb{R}^n$. A lattice can be generated by n linearly independent basis $B = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n\}$ as $L = L(\mathbf{B}) := \{\sum_{i=1}^{n} k_i \mathbf{b}_i | k_i \in \mathbb{Z}\}$. For a real $s > 0$, the gaussian distribution function on a real $x \in \mathbb{R}$ is defined as $\rho_s(x) = e^{-\pi \frac{x^2}{s^2}}$. For a positive matrix Σ, we extend the definition over a n-dimensional vector $\boldsymbol{x} \in \mathbb{R}^n$ by letting $\rho_{\Sigma}(\boldsymbol{x}) = e^{-\pi\|\boldsymbol{x}\Sigma\boldsymbol{x}\|^2}$. For a probability distribution ρ and a S subset of ρ's support, we let $\rho(S) := \sum_{x \in S} \rho(x)$. For a natural number n and a discrete set $S \subset \mathbb{Z}^n$, the discrete gaussian distribution $D_s : S^n \to [0,1]$ is defined as $D_{S,s}(\boldsymbol{x}) := \frac{\rho_s(\boldsymbol{x})}{\rho_s(S^n)}$. For a polynomial $\boldsymbol{a} = \sum_{i \in [n]} a_i x^i$, we say $\boldsymbol{a}$ is sampled from $D_{\mathbb{Z},s}^{\mathsf{Coeffs}}$, if the coefficient vector $\boldsymbol{a} = (a_0, a_1, \cdots, a_{n-1})$ is sampled from $D_{\mathbb{Z},s}$.

ring-LWE. Before recalling the definitions of ring LWE, we first define the rings that we work on in this paper. One thing need to be noticed that our construction of GKE and instantiation of key reconciliation mechanisms are independent of concrete instantiations. The reason of using ring LWE is because of its compactness and commutativity. One can instantiate the scheme with plain LWE or any version of learning with rounding problems following the constraints of GKE.

Let n be a power of 2, we define the polynomial ring $R := \mathbb{Z}[x]/(x^n + 1)$ and let R_q be the quotient R/qR for some positive integer q. For a $s \in R_q$ and gaussian parameter s, we say a pair (a, b) is sampled from the R-LWE distribution, denoted $A_{n,q,s}$, if a is uniformly sampled from R_q and $b = as + e$ for some error term e sampled from $D_{\mathbb{Z},s}^{\mathsf{Coeffs}}$. The goal of R-LWE problem is to distinguish the samples of $A_{n,q,s}$ from the same number of samples of $\mathcal{U}(R_q) \times \mathcal{U}(R_q)$.

Definition 2.4 (ring-LWE) *For any positive integers n, q and gaussian parameter s, we say R-$\mathsf{LWE}_{n,k,q,s}$ is hard if for all* PPT *adversary $\mathcal{A}$, the following holds :*

$$\mathsf{Pr}\left[b' = b \left| \begin{array}{c} b \xleftarrow{\$} \{0,1\}; \\ \textit{if } b = 1, (a_i, b_i)_{i \in [k]} \leftarrow A_{n,q,s}^k; \\ \textit{else }, (a_i, b_i)_{i \in [k]} \xleftarrow{\$} R_q^k \times R_q^k; \\ b' \leftarrow \mathcal{A}((a_i, b_i)_{i \in [k]}); \end{array} \right. \right] - \frac{1}{2} \leq \mathsf{negl}(\lambda),$$

where the probability is over the randomness of all the coin tosses.

Theorem 2.5 [20] *Let α be a positive real, m be a power of 2, l be an integer, $\Phi_m(X) = X^n + 1$ be the m-th cyclotomic polynomial where $m = 2n$, and $R = Z[X]/(\Phi_m(X))$. Let $q \equiv 3 \mod 8$ be a (polynomial size) prime such that there is another prime $p \equiv 1 \mod m$ satisfying $p \leq q \leq 2p$. Let also $\alpha q \geq n^{1.5} k^{0.25} \omega(\log^{2.25}(n))$. Then, there is a probabilistic polynomial-time quantum reduction from $O(n/\alpha)$-approximate SIVP (or SVP) to $RLWE_{n,k,q,\alpha q}$.*

Above theorem shows that for the parameters satisfying the constraints in above theorem, $RLWE_{n,k,q,\alpha q}$ problem is hard if assuming the $O(n/\alpha)$-approximate SIVP is hard. Furthermore, its believed that SIVP remains hard even if the large scale quantum computers are available. Therefore, it is reasonable to assume that the $RLWE_{n,k,q,\alpha q}$ based cryptographic schemes are post-quantum.

3 Multi-Party Key Reconciliation

3.1 Definition

KRM: A key reconciliation mechanism enables participated parties to obtain a key from roughly the same elements. A significant difference between KRM and GKE is that KRM requires all the parties should have some approximately the same elements beforehand. But a GKE not need this requirement at all. Here in what follows, we define the multi-party KRM with its correctness and security.

Definition 3.1 *A N-party key reconciliation mechanism* KeyRek *consist of tuples* $(\mathsf{KeyRek.Hint}, \mathsf{KeyRek.KeyGen})$*, described as follow:*

- $\mathsf{KeyRek.Hint}(b_i)_{i\in[N]} \to \{h_i\}_{i\in[N]}$: *On input b_i, each party P_i for $i \in [N]$ runs this algorithm to obtain a hint message h_i and broadcast it to other parties.*
- $\mathsf{KeyRek.KeyGen}(b_i, \{h_i\}_{i\in[N]})_{i\in[N]} :\to \{K_i\}_{i\in[N]}$: *On input b_i and $\{h_i\}_{i\in[N]}$, each party P_i runs this algorithm to obtain a key K_i.*

where the b_is are the predetermined approximately same elements.

Correctness. For a KRM, we require all the agreed keys are equal except with negligible probability. The formal definition is as follow.

Definition 3.2 *We say multi-party* KeyRek *is correct with respect to β if $\|b_i - b_j\| \le \beta$ for all $i, j \in [N]$ and for some random string K, the probability*

$$\mathsf{Pr}\left[\bigwedge_{i\in[N]} K_i = K \middle| \begin{array}{l} \{h_i\}_{i\in[N]} \leftarrow \mathsf{KeyRek.Hint}(b_i)_{i\in[N]}; \\ \{K_i\}_{i\in[N]} := \mathsf{KeyRek.KeyGen}(b_i, \{h_i\}_{i\in[N]}) \end{array}\right]$$

is at least $1 - \mathsf{negl}(\lambda)$, where the probability is taken over the randomness of b_i.

Security. For any PPT algorithm $\mathcal{A}$ and key space $\mathcal{K}$ of KeyRek, the advantage of $\mathcal{A}$ against the protocol KeyRek, denoted $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KeyRek}}$, is defined as

$$\mathsf{Pr}\left[b' = b \middle| \begin{array}{l} \{h_i\}_{i\in[N]} \leftarrow \mathsf{KeyRek.Hint}(b_i)_{i\in[N]}; \\ \{K_i\}_{i\in[N]} := \mathsf{KeyRek.KeyGen}(b_i, \{h_i\}_{i\in[N]}); \\ b \xleftarrow{\$} \{0,1\},\ \text{if } b = 0, K^* :\xleftarrow{\$} \mathcal{K},\ \text{else } K^* := K_0; \\ b' \leftarrow \mathcal{A}(\{h_i\}_{i\in[N]}, K^*) \end{array}\right] - \frac{1}{2},$$

where the probability is taken over the randomness of b_is and the random coin toss of b. We say KeyRek is secure if the above advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KeyRek}}$ is negligible in the security parameter λ. Note that b_is are approximately the same elements, but it's unknown and random to the adversary. A secure KeyRek should reveals nothing about K_i to the adversary except the public message h_i derived from b_i.

3.2 Instantiation

Here, we instantiate the KRM in Definition3.1 with more special case in which only one participant's hint message is suffice for all the participants to agree on the same session key. In below, we generalized the KRM of [14]. Our description of $\mathsf{KeyRek} = (\mathsf{Hint}, \mathsf{KeyGen})$ is as below where we omit the input integers q, p, g as they are implicitly contained in both algorithms.

Construction 3.3 *For the integers q, p, g such that $2 \leq p$, $p(g+1) < q$ and $gcd(q, g) = 1$, the construction of $\mathsf{KeyRek} = (\mathsf{Hint}, \mathsf{KeyGen})$ as follows:*

$\mathsf{Hint}(K') \to h$ *: On input $K' \in \mathbb{Z}$, it runs as follow:*

- *(1)* $i \xleftarrow{\$} \mathbb{Z} \cap (-\frac{g}{2}, \frac{g}{2}]$
- *(2)* $h = \lfloor p - \frac{p}{q}K' + \frac{1}{2} + \frac{p}{q}i \rfloor \mod p$
- *(3) Outputs* h

$\mathsf{KeyGen}(K, h) = k$ *: On input $K \in \mathbb{Z}$ and $h \in \mathbb{Z}_p$, it runs:*

- *(1)* $k = (K + \lfloor h\frac{q}{p} \rfloor \mod {}^{\pm}q) \mod g$
- *(2) Outputs* k

For any integer $x \in \mathbb{Z}$, we let $(x \mod {}^{\pm}q)$ be an integer in $(\frac{-q}{2}, \frac{q}{2}]$. In what follows, we prove the correctness and security of above KeyRek.

Theorem 3.4 *For the integer parameters as in Construction3.3, if for any $K', K \in \mathbb{Z}_q$, there is an integer d such that $K - K' = dg$ and $|dg| \leq q\frac{p-1}{2p} - \frac{g+1}{2}$, then we have*

$$\mathsf{KeyGen}(K', h) = \mathsf{KeyGen}(K, h),$$

where $h = \mathsf{Hint}(K')$.

Proof. Since $K = dg + K'$, we re-write $\mathsf{KeyGen}(K, h)$ as

$$\begin{aligned}
&\mathsf{KeyGen}(K, h) = \mathsf{KeyGen}(K' + dg, h) \\
&= \left(K' + \lfloor h\frac{q}{p} \rfloor + dg \mod {}^{\pm}q \right) \mod g \\
&= \left(\underbrace{((K' + \lfloor h\frac{q}{p} \rfloor \mod {}^{\pm}q)}_{\leq \frac{q}{2p} + \frac{g+1}{2} \leq \frac{q}{2} - |dg|} + \underbrace{dg}_{\leq |dg|}) \mod {}^{\pm}q \right) \mod g.
\end{aligned}$$

Since $|dg| \leq q\frac{p-1}{2p} - \frac{g+1}{2} < \frac{q}{2}$, we have $\frac{q}{2} - |dg| \geq \frac{q}{2p} + \frac{g+1}{2}$. So if $|(K' + \lfloor h\frac{q}{p} \rfloor \mod {}^{\pm}q)| \leq \frac{q}{2p} + \frac{g+1}{2}$, then we can remove the second $\mod {}^{\pm}q$ operation from the representation of $\mathsf{KeyGen}(K,h)$. That is to say, $\mathsf{KeyGen}(K,h)$ can be re-written as

$$\begin{aligned}\mathsf{KeyGen}(K,h) &= \left(K' + \lfloor h\frac{q}{p}\rceil \mod {}^{\pm}q\right) + dg \mod g\\ &= \left(K' + \lfloor h\frac{q}{p}\rceil \mod {}^{\pm}q\right) \mod g\\ &= \mathsf{KeyGen}(K',h)\end{aligned}$$

Therefore, to compete the proof, we need to show: $|(K'+\lfloor h\frac{q}{p}\rceil \mod {}^{\pm}q)| \leq \frac{q}{2p}+\frac{g+1}{2}$.

Replacing the h in $(K' + \lfloor h\frac{q}{p}\rceil \mod {}^{\pm}q)$ with explicit representation of h in Hint, it's easy to see the following

$$\begin{aligned}&|(K' + \lfloor h\frac{q}{p}\rceil \mod {}^{\pm}q)|\\ =\ &|\underbrace{\lfloor K' + \frac{q}{p}(\lfloor p - \frac{p}{q}K' + \frac{1}{2} + \frac{p}{q}i\rfloor \mod p)\rceil}_{\in\left(\kappa\cdot q - \frac{q}{2p}+i, \kappa\cdot q+\frac{q}{2p}+i\right]} \mod {}^{\pm}q|,\end{aligned}$$

where κ is some integer. In addition, we have that $(K'+\lfloor h\frac{q}{p}\rceil \mod {}^{\pm}q) \in \left(\frac{-q}{2p}+i, \frac{q}{2p}+i\right]$, and thus $|(K' + \lfloor h\frac{q}{p}\rceil \mod {}^{\pm}q)| \leq \frac{q}{2p} + \frac{g+1}{2}$. This completes the proof. □

The following theorem shows the uniformity of our KeyRek.

Theorem 3.5 *For the parameters as in Construction3.3, and a uniform K, the $\mathsf{KeyGen}(K,h)$ is uniformly distributed conditioned on $h = \mathsf{Hint}(K)$, i.e.,*

$$\Pr_{K\leftarrow\mathbb{Z}_q}[\mathsf{KeyGen}(K,h) = k|\mathsf{Hint}(K) = h] = \frac{1}{g},$$

where $k \in \mathbb{Z}_g$.

Proof. Let $\mathsf{Hint}(K,i)$ be the deterministic version of $\mathsf{Hint}(K)$(making the implicit randomness $i \in \mathbb{Z}_g$ as an explicit input), proving following two statements is suffice to complete the proof.

(1) For any $i \in \mathbb{Z}_g$, we have

$$\Pr_{K\leftarrow\mathbb{Z}_q}[\mathsf{Hint}(K,i) = h] = \frac{|T_h^i|}{q}, \text{ and}$$

$$\Pr_{K\leftarrow\mathbb{Z}_q}[\mathsf{KeyGen}(K,h) = k \wedge \mathsf{Hint}(K,i) = h] = \frac{|T_{h,k}^i|}{q},$$

where T_h^i and $T_{h,k}^i$ are defined as

$$T_h^i := \left(\frac{q}{p}(p-h-\frac{1}{2})+i, \frac{q}{p}(p-h+\frac{1}{2})+i\right],$$
$$T_{h,\kappa}^i := \{x \in T_h^i | \mathsf{KeyGen}(x,h)=\kappa\}.$$

(2) For any $i \in \mathbb{Z}_g$ and $T_h^i, T_{h,\kappa}^i$ defined above, we have

$$T_h^i := \bigcup_{\kappa \in \mathbb{Z}_g} T_{h,\kappa}^i, |T_h^i| = |T_h^0|, \text{ and}$$

$$|T_h^0| = \sum_{\kappa \in \mathbb{Z}_g} |T_{h,\kappa}^i| = \sum_{i \in \mathbb{Z}_g} |T_{h,k}^i|.$$

This is the case, since we have

$$\begin{aligned}
&\Pr_{K \leftarrow \mathbb{Z}_q}[\mathsf{KeyGen}(K,h)=k|\mathsf{Hint}(K)=h] \\
&= \frac{1}{g}\sum_{i \leftarrow \mathbb{Z}_g} \Pr_{K \leftarrow \mathbb{Z}_q}[\mathsf{KeyGen}(K,h)=k|\mathsf{Hint}(K,i)=h] \\
&= \frac{1}{g}\sum_{i \leftarrow \mathbb{Z}_g} \frac{\Pr_{K \leftarrow \mathbb{Z}_q}[\mathsf{KeyGen}(K,h)=k \wedge \mathsf{Hint}(K,i)=h]}{\Pr_{K \leftarrow \mathbb{Z}_q}[\mathsf{Hint}(K,i)=h]} \\
&= \frac{1}{g}\sum_{i \leftarrow \mathbb{Z}_g} \frac{\frac{|T_{h,k}^i|}{q}}{\frac{|T_h^i|}{q}} = \frac{1}{g}
\end{aligned}$$

where the first and second equality is by property of probability; the third equality is by statement (1); the last equality is by the statement (2). In what follows, we prove (1) and (2)

Now, we prove (1). We first show $\Pr_{K \leftarrow \mathbb{Z}_q}[\mathsf{Hint}(K,i)=h] = \frac{|T_h^i|}{q}$ as follow: From the definition of T_h^i, it is easy to verify that, for any $x \in T_h^i$ we have $\mathsf{Hint}(x,i)=h$; Furthermore, T_h^is are disjoint and $\mathbb{Z}_q = \cup_{\bar{h} \in \mathbb{Z}_p} T_{\bar{h}}^i$, and thus for any $x \notin T_h^i$, there is some $h' \neq h$ such that $x \in T_{h'}^i$ and $\mathsf{Hint}(x,i) = h' \neq h$. It is obvious from the definition of $T_{h,k}^i$ that $\Pr_{K \leftarrow \mathbb{Z}_q}[\mathsf{KeyGen}(K,h)=k \wedge \mathsf{Hint}(K,i)=h] = \frac{|T_{h,k}^i|}{q}$.

Next, we prove (2). Since KeyGen is deterministic algorithm of K and h, $T_{h,k}^i$s are the partitioning of T_h^i, that is $T_h^i = \cup_{\kappa \in \mathbb{Z}_g} T_{h,\kappa}^i$. Observing the definition of T_h^i, it's not hard to find that T_h^i is the shift of T_h^0(e.g., $T_h^i = T_h^0 + i$), and thus $|T_h^i| = |T_h^0|$. To show $\sum_{\kappa \in \mathbb{Z}_g} |T_{h,\kappa}^i| = \sum_{i \in \mathbb{Z}_g} |T_{h,k}^i|$, verifying the existence of a bijection between

$T^i_{h,k}$ and $T^{i-1}_{h,k-1}$ is suffice. This is the case, because we have $|T^i_{h,k}| = |T^{i-1}_{h,k-1}|$ in this case, and

$$\sum_{i\in\mathbb{Z}_g} |T^i_{h,k}| = \sum_{i\in\mathbb{Z}_g} |T^0_{h,k-i}| = \sum_{\kappa\in\mathbb{Z}_g} |T^0_{h,\kappa}| = |T^0_h|.$$

Here, we define the map $f : T^i_{h,k} \longrightarrow T^{i-1}_{h,k-1}$ as $f(x) = x - 1$ and prove this is a bijective map. We first show that for any $x \in T^i_{h,k}$, $f(x) \in T^{i-1}_{h,k-1}$. From the definition of the algorithms Hint and KeyGen, we have $\mathsf{Hint}(x-1, i-1) = \mathsf{Hint}(x, i) = h$ and $\mathsf{KeyGen}(x-1, h) = k - 1 \mod g$, and thus $f(x) \in T^{i-1}_{h,k}$. It's straight that f is bijective map. This completes the proof. □

The following is a multi-party KRM using the Construction3.3 as a building block.

Construction 3.6 *A N-party key reconciliation mechanism* KeyRek *is consist of algorithm tuples* $(\mathsf{KeyRek.Hint}, \mathsf{KeyRek.KeyGen})$ *as follow:*

- $\mathsf{KeyRek.Hint}(b_i)_{i\in[N]} \to \{h_i\}_{i\in[N]}$: *On input b_0, party P_0 computes $h_0 = \mathsf{Hint}(b_0)$ and broadcast h_0 to other parties, then each party P_i set $h_i = h_0$.*
- $\mathsf{KeyRek.KeyGen}(b_i, \{h_i\}_{i\in[N]})_{i\in[N]} :\to \{K_i\}_{i\in[N]}$: *On input b_i and $\{h_i\}_{i\in[N]}$, each party P_i runs $\mathsf{KeyGen}(b_i, \{h_i\}_{i\in[N]})$ to obtain a key k_i.*

where the b_is are the predetermined approximately same elements.

As described in above Construction3.6, this KeyRek is a special case of Definition3.1. In general, we have following result.

Theorem 3.7 *For the integers p, q, g such that $p < q$, $q > p(g+1)$ and $gcd(q, g) = 1$, there exist a multi-party KRM that is secure and correct respect to $q\frac{p-1}{2pg} - \frac{g+1}{2g}$.*

Proof. The Construction3.6 is the witness to the existence of such multi-party KRM. The security and correctness are simply followed from the Theorem3.5 and the Theorem3.4. □

4 A Weaker Version of GKE

In this section, we define a weak version of GKE, and then we show the RLWE based instantiation.

4.1 Weak GKE

The correctness of a GKE protocol guarantees that the participated parties can have the same session key. But, in this section, we degrade the correctness of the GKE protocol, and we call this new degraded protocol as *weak* GKE. More specifically, at the end of a *weak* GKE, the correctness of the *weak* GKE requires that the participants agree on an approximately the same element rather than an exactly the same element. The definition of a *weak* GKE is identical to the definition GKE(Definition 2.1), and thus we omit the formal definition here. We define the correctness of a *weak* GKE protocol as follows.

Definition 4.1 (Correctness.) *For a real $\gamma > 0$, we say a* weak-GKE *is correct respect to γ^3 if the probability*

$$\mathsf{Pr}\left[\bigwedge_{i,j\in[N]} \|K_i - K_j\| \le \gamma \,\middle|\, \begin{array}{l} \mathsf{pp} \leftarrow w\mathsf{GKE.Setup}(1^\lambda, 1^N) \\ \{\mathsf{trans}_i, \mathsf{st}_i\}_{i\in[N]} \leftarrow w\mathsf{Interact}(P_i, \mathsf{pp})_{i\in[N]} \\ \{K_i\}_{i\in[N]} := w\mathsf{KeyGen}(\mathsf{pp}, P_i, \{\mathsf{trans}_i, \mathsf{st}_i\})_{i\in[N]} \end{array}\right]$$

is negligible, where the probability is taken over the randomness of weak-KeyGen *algorithm and randomness of* weak-Interact *algorithm.*

The above correctness definition of a *weak*-GKE shows that all the agreed keys from a *weak*-GKE protocol should be near each other instead of requiring them to be equal. Intuitively, this relaxed version seems to be easily reached, and we will show an explicit instantiation in next section.

security Here we define the security of *weak* GKE which is slightly different from the security definition of GKE. Recall the security definition of a GKE protocol, all the keys should be exactly equal, and thus there is no difference either of the following two cases: (1) the adversary is only given a single key, or (2) the adversary has all the keys. But in the case of *weak* GKE, the approximate-equality is needed, and thus above two cases are different. Here, the adversary is asked to distinguish the derived keys of a *weak* GKE from the same number of random dense keys.

For a probabilistic polynomial time(PPT) algorithm $\mathcal{A}$ and the key space $\mathcal{K}$ of a $w\mathsf{GKE}$, we define the advantage of $\mathcal{A}$ against the protocol $w\mathsf{GKE}$, denoted $\mathsf{Adv}_{\mathcal{A}}^{w\mathsf{GKE}}$, as follow:

$$\mathsf{Pr}\left[b' = b \,\middle|\, \begin{array}{l} \mathsf{pp} \leftarrow w\mathsf{GKE.Setup}(1^\lambda, 1^N) \\ \{\mathsf{trans}_i, \mathsf{st}_i\}_{i\in[N]} \leftarrow w\mathsf{GKE.Interact}(P_i, \mathsf{pp})_{i\in[N]} \\ \{K_i\}_{i\in[N]} := w\mathsf{GKE.KeyGen}(\mathsf{pp}, P_i, \{w\mathsf{trans}_i, \mathsf{st}_i\})_{i\in[N]} \\ b \xleftarrow{\$} \{0,1\},\ \text{if } b = 1, K_i^* := K_i \text{ for all } i \in [N], \\ \text{else } K_0^* :\xleftarrow{\$} \mathcal{K}, \{K_i^*\}_{i\in 1,\dots,N-1} := K_0^* + \chi_{\gamma/2}, \\ b' \leftarrow \mathcal{A}(\{\mathsf{trans}_i\}_{i\in[N]}, \mathsf{pp}, \{K_i^*\}_{i\in[N]}) \end{array} \right] - \frac{1}{2},$$

where $\chi_{\gamma/2}$ is a distribution bounded by $\gamma/2$, the probability is taken over the randomness of $w\mathsf{GKE.KeyGen}$, $w\mathsf{GKE.Interact}$, χ and random coin toss of b. Our security definition of a *weak* GKE is as follows.

[3] We use the notion *correct respect to* γ rather than γ-correct due to the fact that the latter usually used to imply the correctness not holds with probability γ.

Definition 4.2 *We call a protocol* $w\mathsf{GKE}$ *is passively secure if the advantage of any* PPT *algorithm* $\mathcal{A}$*(eavesdropper) against the protocol* $w\mathsf{GKE}$ *is negligible in the security parameter* λ*, i.e.,* $\mathsf{Adv}_{\mathcal{A}}^{w\mathsf{GKE}}(\lambda) \leq \mathsf{negl}(\lambda)$*.*

Note that a GKE protocol is obviously a weak GKE. We instantiate this weaker version of GKE in the following section.

4.2 Instantiation of Weak GKE

In this section, instead of presenting a concrete instantiation of $w\mathsf{GKE}$, we give a high level description of its existence. In particular, we construct a $w\mathsf{GKE}$ using a similar way as in [5,18].

Construction 4.3 *The construction of* $w\mathsf{GKE}$ *is consist of three algorithm triples* $(w\mathsf{GKE.Setup}, w\mathsf{GKE.Interact}, w\mathsf{GKE.KeyGen})$ *as follows:*
$w\mathsf{GKE.Setup}(\lambda)$ *: On input the security parameter* λ *and number of participants* N*, it first choose a random ring element* a*, a PRF, and an obfuscated circuit* C*(described in Construction 4.4),*
$w\mathsf{GKE.Interact}(\mathsf{pp}, P_i)_{i\in[N]}$*: On input public parameter* pp*, party* P_i *choose a pair* $(s_i, e_i) \overset{\chi}{\leftarrow} R$*, and comput* $b_i = a \cdot s_i + e_i$*, then broadcast* b_i*.*
$w\mathsf{GKE.KeyGen}(\mathsf{pp}, P_i, \{b_i,\}_{i\in[N]}, (s_i, e_i))_{i\in[N]}$*: Party* P_i *evaluate the obfuscated circuit* C *on input* $\{b_j, s_j, e_j\}_{j\in[N]}$*, and let the sum of a random bounded value and the evaluation of* C *as it's session key.*

Construction 4.4 *Input :* a*,* $(b_i, s_i, e_i)_{i\in[N]}$*, PRF:*

```
For i =0 to n :
    if $b_i = a \cdot s_i + e_i$, output $PRF(b_1, \cdots, b_N)$.
Otherwise output $\perp$.
```

Hardness result of RLWE shows that b_i only leaks negligible information about the pair (s_i, e_i). Since the ith party has exact values of (s_i, e_i), $\{b_j\}_{j\in[N]}$, and incorrect values of s_j, e_j's for $j \neq i$, the statement $b_k = a \cdot s_k + e_k$ holds for $k = i$, and thus it will correctly have the value $PRF(b_1, \cdots, b_N)$. But those who hasn't any exact pair of (s_i, e_i) unable to have $PRF(b_1, \cdots, b_N)$.

5 Group Key Exchange Protocol from Ring LWE

In this section, we firstly construct a GKE protocol from a weak GKE and key reconciliation mechanism. Then, we will show its correctness and security. After all, we will instantiate the GKE protocol.

5.1 construction

In this section, we present the construction of group key exchange scheme $\mathsf{GKE} = (\mathsf{GKE.Setup}, \mathsf{GKE.Interact}, \mathsf{GKE.KeyGen})$ from a combination of weak group key exchange scheme $w\mathsf{GKE} = (w\mathsf{GKE.Setup}, w\mathsf{GKE.Interact}, w\mathsf{GKE.KeyGen})$ and a multiparty key reconciliation mechanism $\mathsf{KeyRek} = (\mathsf{KeyRek.Hint}, \mathsf{KeyRek.KeyGen})$. let λ be the security parameter and N be the number of participants, the construction if GKE is as follows.

Construction 5.1 *The description of* GKE *as follows:*

$\mathsf{GKE.Setup}(1^\lambda, N) \rightarrow \mathsf{pp}$**:** *On input the security parameter* λ *and number of participants* N*, it obtains* pp *by running* $w\mathsf{GKE.Setup}(1^\lambda, N)$*.*

$\mathsf{GKE.Interact}(\mathsf{pp}, P_i)_{i\in[N]} \rightarrow \{\mathsf{trans}_i, h_i, \mathsf{st}_i\}_{i\in[N]}$**:** *On input the public parameter* pp*, each party* P_i *do the followings :*

1. $(\mathsf{trans}_i, \mathsf{st}_i) \leftarrow w\mathsf{GKE.Interact}(\mathsf{pp}, P_i)$*, and brodcast* trans_i
2. $(K_i) \leftarrow w\mathsf{GKE.KeyGen}(\mathsf{trans}_i, \mathsf{st}_i, P_i)$
3. $(h_i) \leftarrow \mathsf{KeyRek.Hint}(K_i)$*, and brodcast it.*

$\mathsf{GKE.KeyGen}(\mathsf{pp}, (\mathsf{trans}_i, h_i, \mathsf{st}_i)_{i\in[N]}) = \{k_i\}_{i\in[N]}$**:** *On inputs* $\mathsf{pp}, (\mathsf{trans}_i, h_i, \mathsf{st}_i)_{i\in[N]}$ *generated from previous algorithms, it first generate* K_i *by running the algorithm* $w\mathsf{GKE.KeyGen}(\mathsf{trans}_i, \mathsf{st}_i, P_i)$*. Then it runs* $\mathsf{KeyRek.KeyGen}(K_i, \{h_i\}_{i\in[N]})$ *to get* k_i*.*

Correctness. Following theorem shows the correctness of above GKE.

Theorem 5.2 *The GKE protocol* GKE *presented in Construction5.1 is correct if the* $w\mathsf{GKE}$ *and* KeyRek *are correct respect to* γ*.*

Proof. From the correctness definition of GKE, we have following by union bound

$$\begin{aligned}\mathsf{Pr}\left[\underset{i,j\in[N]}{\wedge} k_i = k_j\right] &= 1 - \mathsf{Pr}\left[\underset{i,j\in[N]}{\wedge} k_i \neq k_j\right] \\ &\leq 1 - N^2 \max_{i,j\in[N]} \mathsf{Pr}\left[k_i \neq k_j\right].\end{aligned}$$

Hence, for any $i, j \in [N]$, showing $\mathsf{Pr}[k_i \neq k_j] \leq \mathsf{negl}(\lambda)$ is suffice to show the theorem. To show this, we rewrite the probability $\mathsf{Pr}[k_i \neq k_j]$ as

$$\begin{aligned}&\mathsf{Pr}[k_i \neq k_j \wedge \|K_i - K_j\| \leq \gamma] + \mathsf{Pr}[k_i \neq k_j \wedge \|K_i - K_j\| > \gamma] \\ &\leq \mathsf{Pr}[k_i \neq k_j | \|K_i - K_j\| \leq \gamma] + \mathsf{Pr}[\|K_i - K_j\| > \gamma]\end{aligned}$$

Since $w\mathsf{GKE}$ is correct respect to γ, thus $\|K_i - K_j\| \leq \gamma$ holds except with $\mathsf{negl}(\lambda)$ probability. In addition, the conditional probability of $k_i = k_j$ on $\|K_i - K_j\| \leq \gamma$ is at least $1 - \mathsf{negl}(\lambda)$ by the correctness of KeyRek. Therefore we have $\mathsf{Pr}[k_i \neq k_j] \leq \mathsf{negl}(\lambda)$. This completes the proof

□

5.2 Security and Instantioation

Let $\chi_{\gamma/2}$ be some bounded distribution, the following theorem shows the security of the GKE in Construction 5.1.

Theorem 5.3 *The GKE protocol* GKE *presented in Construction5.1 is(passively) secure assuming the(passive) security of* KeyRek *and security of* $w\mathsf{GKE}$ *respect to* $\chi_{\gamma/2}$.

Proof. We prove the theorem by contradiction. We start by assuming the theorem is false, that is there exists an adversary $\mathcal{A}$ which can break the protocol GKE, then we will show that at least one of the following two statements holds: (1) There is a simulator $\mathsf{Sim1}$ which breaks the security of $w\mathsf{GKE}$. (2) There is a simulator $\mathsf{Sim2}$ which breaks the security of KeyRek. This contradict with the theorem assumption that $w\mathsf{GKE}$ and KeyRek are secure, and thus the theorem holds.

To show the existence of an adversary $\mathcal{A}$ which can break the protocol GKE implies one of the above two statements is true, we need following sequence of games.

Game_0 This is the ordinary GKE security game between the $\mathcal{A}$ and the challenger.

Game_1 In this game, we modify the game so that challenger chooses K_0 from uniform distribution instead of generating it by running $w\mathsf{GKE.KeyGen}$, and then let $K_i = K_0 + \chi_\gamma$. Here we also require χ be some distribution bounded by $\gamma/2$ except with negligible probability.

Game_2 In this game, we change the way that the challenger compute the challenge key k^*. Here the challenge key k^* is chosen randomly instead of generating it by using $\mathsf{KeyRek.KeyGen}$.

In what follows, we show the advantage of $\mathcal{A}$ in the Game_0, denoted $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_0}$, is upper bounded by the advantage of corresponding algorithms($\mathsf{Adv}_{\mathsf{Sim1}}^{w\mathsf{GKE}}$ and $\mathsf{Adv}_{\mathsf{Sim2}}^{\mathsf{KeyRek}}$) plus a negligible function in security parameter λ. By our assumption, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_0}$ is noticeable, therefore $\mathsf{Adv}_{\mathsf{Sim1}}^{w\mathsf{GKE}}$ or $\mathsf{Adv}_{\mathsf{Sim2}}^{\mathsf{KeyRek}}$ is non-negligible, and this is what we want to prove. Now, we show it by following lemmas.

Lemma 5.4 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_0} \leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1} + \mathsf{Adv}_{\mathsf{Sim1}}^{w\mathsf{GKE}}$.

Proof. The algorithm $\mathsf{Sim1}$ works as follows:

At the beginning of the game, algorithm $\mathsf{Sim1}$ is given $(\mathsf{pp}, \{\mathsf{trans}_i, K_i^*\}_{i\in[N]})$ from its challenger, where pp is the public parameter of $w\mathsf{GKE}$, trans_is are the transactions and K_i^*s are derived keys of the $w\mathsf{GKE}$ if the challenger's coin toss $b = 1$, and $K_0^* :\overset{\$}{\leftarrow} \mathcal{K}, \{K_i^*\}_{i\in 1,\ldots,N-1} := K_0^* + \chi_{\gamma/2}$ otherwise. Then, $\mathsf{Sim1}$ computes $h_i := \mathsf{KeyRek.Hint}(K_i^*)$ for all $i \in [N]$, $k^* := \mathsf{KeyRek.KeyGen}(K_0, \{h_i\}_{i\in[N]})$, and send $(\mathsf{pp}, \{\mathsf{trans}_i, h_i\}_{i\in[N]}, k^*)$ to the adversary $\mathcal{A}$. At the end of the game, after receiving the $\mathcal{A}$'s guessing bit $b' \in \{0, 1\}$, $\mathsf{Sim1}$ outputs b' as its guess of b.

If the challenger's coin toss $b = 1$, then $\mathsf{Sim1}$ perfectly simulate the view of $\mathcal{A}$ as in Game_0. Otherwise, $\mathsf{Sim1}$ simulates the view of $\mathcal{A}$ as in Game_1. Therefore, the lemma follows.

□

Lemma 5.5 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1} \leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2} + \mathsf{Adv}_{\mathsf{Sim}}^{\mathsf{KeyRek}}$.

Proof. The algorithm $\mathsf{Sim2}$ runs as follows:

> At the beginning of the game, algorithm $\mathsf{Sim2}$ is given $(\{h_i\}_{i\in[N]}, K^*)$ from its challenger, where $h_i = \mathsf{KeyRek.KeyGen}(U_i)$ for uniformly distributed U_is such that $\|U_i - U_j\| \leq \gamma, \forall i, j \in [N]$. The key K^* is generated by the algorithm $\mathsf{KeyRek.KeyGen}(U_0, \{h_i\}_{i\in[N]})$ if the challenger's coin toss $b = 1$ and generated by randomly if $b = 0$. Then, $\mathsf{Sim2}$ obtain $(\mathsf{pp}, \{\mathsf{trans}_i\}_{i\in[N]})$ by running $w\mathsf{GKE.Setup}$ and $w\mathsf{GKE.Interact}$. Next, $\mathsf{Sim2}$ let $k^* := K^*$ and sends $(\mathsf{pp}, \{\mathsf{trans}_i, h_i\}_{i\in[N]}, k^*)$ to the adversary $\mathcal{A}$. At the end of the game, after receiving the $\mathcal{A}$'s guessing bit $b' \in \{0, 1\}$, $\mathsf{Sim2}$ outputs b' as its guess of b.

If the challenger's coin toss $b = 1$, then $\mathsf{Sim2}$ perfectly simulate the view of $\mathcal{A}$ as in Game_1. Otherwise, $\mathsf{Sim2}$ simulates the view of $\mathcal{A}$ as in Game_2. Therefore, the lemma follows. □

Complete the proof. Since the key k^* in Game_2 is uniformly selected, then the adversary's advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}$ is zero. Furthermore, combining the above lemmas, we have

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_0} &\leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1} + \mathsf{Adv}_{\mathsf{Sim1}}^{w\mathsf{GKE}}\\ &\leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2} + \mathsf{Adv}_{\mathsf{Sim2}}^{\mathsf{KeyRek}} + \mathsf{Adv}_{\mathsf{Sim1}}^{w\mathsf{GKE}}\\ &\leq \mathsf{Adv}_{\mathsf{Sim2}}^{\mathsf{KeyRek}} + \mathsf{Adv}_{\mathsf{Sim1}}^{w\mathsf{GKE}}.\end{aligned}$$

Since $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_0}$ is noticeable, at least on of $\mathsf{Adv}_{\mathsf{Sim1}}^{\mathsf{KeyRek}}$ and $\mathsf{Adv}_{\mathsf{Sim2}}^{w\mathsf{GKE}}$ is noticiable. This completes the proof.

□

Instantiation. Concrete Instantiation of GKE straightly obtained by combining the instantiations of multi-party KeyRek and $w\mathsf{GKE}$ from the previous sections. Therefore, we omit the concrete description here.

References

1. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In T. Holz and S. Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, Aug. 2016.

2. D. Apon, D. Dachman-Soled, H. Gong, and J. Katz. Constant-round group key exchange from the ring-LWE assumption. In J. Ding and R. Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 189–205. Springer, Heidelberg, 2019.
3. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu, and V. Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993.
4. M. Bellare and P. Rogaway. Provably secure session key distribution: The three party case. In *27th ACM STOC*, pages 57–66. ACM Press, May / June 1995.
5. D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79(4):1233–1285, 2017.
6. J. W. Bos, C. Costello, M. Naehrig, and D. Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 553–570. IEEE Computer Society, 2015.
7. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013.
8. E. Bresson and D. Catalano. Constant round authenticated group key agreement via distributed computation. In F. Bao, R. Deng, and J. Zhou, editors, *PKC 2004*, volume 2947 of *LNCS*, pages 115–129. Springer, Heidelberg, Mar. 2004.
9. E. Bresson, O. Chevassut, and D. Pointcheval. Provably authenticated group Diffie-Hellman key exchange – the dynamic case. In C. Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 290–309. Springer, Heidelberg, Dec. 2001.
10. E. Bresson, O. Chevassut, and D. Pointcheval. Dynamic group Diffie-Hellman key exchange under standard assumptions. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 321–336. Springer, Heidelberg, Apr. / May 2002.
11. E. Bresson, O. Chevassut, D. Pointcheval, and J. Quisquater. Provably authenticated group diffie-hellman key exchange. In M. K. Reiter and P. Samarati, editors, *CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001*, pages 255–264. ACM, 2001.
12. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.
13. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976.
14. J. Ding. New cryptographic constructions using generalized learning with errors problem. Cryptology ePrint Archive, Report 2012/387, 2012. `http://eprint.iacr.org/2012/387`.
15. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal Gaussians. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 40–56. Springer, Heidelberg, Aug. 2013.
16. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.
17. L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 335–352. Springer, Heidelberg, Aug. 2014.
18. S. Guo, P. Kamath, A. Rosen, and K. Sotiraki. Limits on the efficiency of (ring) LWE based non-interactive key exchange. In A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 374–395. Springer, Heidelberg, May 2020.
19. Z. Jin and Y. Zhao. Optimal key consensus in presence of noise. Cryptology ePrint Archive, Report 2017/1058, 2017. `http://eprint.iacr.org/2017/1058`.
20. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In J. H. Cheon and T. Takagi, editors,

ASIACRYPT 2016, Part II, volume 10032 of *LNCS*, pages 682–712. Springer, Heidelberg, Dec. 2016.

21. J. Katz and M. Yung. Scalable protocols for authenticated group key exchange. *Journal of Cryptology*, 20(1):85–113, Jan. 2007.
22. H. Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 546–566. Springer, Heidelberg, Aug. 2005.
23. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.
24. A. Menezes, M. Qu, and S. Vanstone. Some key agreement protocols providing implicit authentication. 01 1995.
25. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009.
26. C. Peikert. Lattice cryptography for the internet. In M. Mosca, editor, *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014*, pages 197–219. Springer, Heidelberg, Oct. 2014.
27. C. Peikert, O. Regev, and N. Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In H. Hatami, P. McKenzie, and V. King, editors, *49th ACM STOC*, pages 461–473. ACM Press, June 2017.
28. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
29. P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, Nov. 1994.
30. J. Zhang, Z. Zhang, J. Ding, M. Snook, and Ö. Dagdelen. Authenticated key exchange from ideal lattices. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 719–751. Springer, Heidelberg, Apr. 2015.

© 2021 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.