Constant-round Dynamic Group Key Exchange from RLWE Assumption

Rakyong Choi, Dongyeon Hong, and Kwangjo Kim

Korea Advanced Institute of Science and Technology, Daejeon, South Korea
{thepride, decenthong93, kkj}@kaist.ac.kr

Abstract. In this paper, we propose a novel lattice-based group key exchange protocol with dynamic membership. Our protocol is constructed by generalizing Dutta-Barua protocol to RLWE setting, inspired by Apon *et al.*'s recent paper in PQCrypto 2019.
We describe our (static) group key exchange protocol from Apon *et al.*'s paper by modifying its third round and computation step. Then, we present both authenticated and dynamic group key exchange protocol with Join and Leave algorithms. The number of rounds for authenticated group key exchange remains the same as unauthenticated one.
Our protocol also supports the scalable property so that the number of rounds does not change depending on the number of group participants. By assuming the hardness of RLWE assumption and unforgeability of digital signatures, we give a full security proof for (un-)authenticated (dynamic) group key exchange protocols.

Keywords: Dynamic group key exchange · authenticated key exchange · RLWE · constant-round group key exchange

1 Introduction

An authenticated key exchange (AKE) protocol is needed over an insecure channel, to prevent any attacks in the presence of active adversaries, to read transmitted messages during a secure communication between two parties over the network. As network topology becomes more complex, we require a secure communication between multiple parties instead of two parties. A group key exchange (GKE) protocol is a cryptographic primitive that establishes a common group secret key in which a shared secret is derived from group members There have been many works on GKE protocols [2, 5, 9–13, 18, 22–24, 30–32, 34].

On the other hand, as quantum computer becomes realistic, National Institute of Standards and Technology (NIST) has been selecting standard post-quantum cryptographic algorithms like key exchange, encryption, and signature schemes. Unfortunately, group (authenticated) key exchange protocol is out-of-scope in this competition. Beyond NIST post-quantum algorithm standardization, there are a few work on post-quantum GKE protocols. Ding *et al.* [17] constructed the first lattice-based GKE protocol and Yang *et al.* [34] and Apon *et al.* [2] suggested constant-round lattice-based GKE protocols, respectively.

But, to the best of our knowledge, there exists no post-quantum dynamic GKE without trusted authority that involves in the process of generating common secret key, in the literature.

1.1 Our Contributions

In this paper, we give a constant-round dynamic GKE protocol based on hardness of RLWE assumption [26] where a party can join or leave the group. We extend two-round Dutta-Barua protocol [18] into RLWE setting.

Given a group $\mathbb{G}$ of prime order q and a generator $g \in \mathbb{G}$, we briefly describe Burmester-Desmedt and Dutta-Barua protocols as below:

1. **(Round 1)** Each party P_i chooses "uniform" value $r_i \in \mathbb{Z}_q$ and broadcasts $z_i = g^{r_i}$ to all other parties.
2. **(Round 2)** Each party P_i broadcasts $X_i = (z_{i+1}/z_{i-1})^{r_i}$ to all other parties.
3. **(Key Computation)**
   – Burmester-Desmedt protocol:
   $b_i = {z_{i-1}}^{Nr_i} \cdot {X_i}^{N-1} \cdot {X_{i+1}}^{N-2} \cdots X_{i+N-2}$.
   – Dutta-Barua protocol:
   Each party P_i calculate $Y_{i+1} = X_{i+1}{z_{i+1}}^{r_i}$ and $Y_{i+j} = X_{i+j}Y_{i+(j-1)}$ for $j = 2$ to $N-1$, then $b_i = \prod_{j=0}^{N-1} Y_{i+j}$.

Since Dutta-Barua protocol is a modification from Burmester-Desmedt protocol [12,13,22] used in Apon *et al.*'s recent work [2], our unauthenticated GKE protocol in static setting is somewhat similar to Apon *et al.*'s protocol.

We apply this relationship into Apon *et al.*'s protocol. Given a ring R_q and a ring element $a \leftarrow R_q$, we sketch our unauthenticated GKE protocol compared to Apon *et al.*'s as below:

1. **(Round 1)** Each party P_i chooses 'small' secret value $s_i \in R_q$ and 'small' noise $e_i \in R_q$ and broadcasts $z_i = as_i + e_i$ to all other parties.
2. **(Round 2)** Each party P_i chooses another 'small' noise $e'_i \in R_q$ and broadcasts $X_i = (z_{i+1} - z_{i-1})\, s_i + e'_i$ to all other parties.
3. **(Key Computation)**
   – Apon *et al.*'s protocol: $b_i = Nz_{i-1}s_i + (N-1)X_i + (N-2)X_{i+1} + \cdots + X_{i+N-2}$.
   – Our protocol: Each party P_i calculate $Y_i = X_i + z_{i-1}s_i$ and $Y_{i+j} = X_{i+j} + Y_{i+(j-1)}$ for $j = 1$ to $N-1$, then $b_i = \sum_{j=0}^{N-1} Y_{i+j}$.

Hence, we follow security analysis of Apon *et al.*'s protocol with slight modification in the presence of the passive adversary. We adopt "unpredictability-based" security analysis (*i.e.*, given the transcript, it is infeasible to determine the real session key) instead of "indistinguishability-based" one (*i.e.*, given the transcript, the real session key should be indistinguishable from random) to apply the characteristic of bounded Rényi divergence.

But instead of applying Katz-Yung compiler [22] for authenticated GKE with active adversary, we adopt the security model of Bresson *et al.* [9] to give a full security analysis of the dynamic case. Hence, our authenticated GKE protocol also achieves forward secrecy, almost fully symmetric and being constant-round but we do not require one more round to achieve AKE security, compared to Apon *et al.*'s protocol.

1.2 Outline of the Paper

The rest of this paper is organized as follows. First, we review the previous work on lattice-based key exchange protocols, constant-round group key exchange, and security models of group key exchange in Chapter 2. We define the basic terms and security model for our protocol in Chapters 3 and 4. Then, we give a design and security analysis of our (authenticated) GKE protocol in Chapters 5 and 6, respectively. In Chapter 7, We compare our protocol with the previous lattice-based GKE protocols and finally, we give a conclusion and future work in Chapter 8.

2 Previous Work

2.1 Constant-round Group Key Exchange

Burmester and Desmedt [12] proposed the first constant-round GKE protocol. In [12], the indices of users are organized logically in a ring structure and the session key is generated by a cyclic function with the contributions of all users. Just and Vaudenay [**?**] proposed an authenticated GKE protocol by combining the idea from [12] and a public key signature scheme. Compared with the protocol of [12], this protocol is more efficient with respect to communication bandwidth while requires four-round to generate the session key.

Katz and Yung [22] brought forward a scalable compiler that converts any unauthenticated GKA protocol into an authenticated key exchange (AKE) security protocol by adding one round to the original protocol. An authenticated GKA protocol is proposed via utilizing the compiler to the protocol of [12], while each user is required to perform additional signing and verification operations.

Dutta and Barua [**?**,18] proposed an two-round authenticated GKE protocol, which is constructed by combining a variant of [12] and a signature scheme modified from [22]. Besides, this protocol supports dynamic membership updating. Compared to the [22] protocol, this protocol is more efficient in communication rounds and computation overhead.

For dynamic GKE, Kim *et al.* [**?**] brought forward a two-round authenticated GKE protocol for the ad-hoc network, in which no trustee is involved. In the protocol [**?**], the XOR operation is introduced into the generation of session key to reduce the computational cost of each group member. Besides, this protocol supports dynamic membership updating, in which the computation and communication overhead of group members rely on the amount of joining/leaving members rather than relying on the cardinality of group members.

Dutta and Barua [?, 18] also proposed a dynamic extension. In the joining algorithm, the original members are considered to be a member with a pre-calculated value, which is generated in the previous session. The new session key is calculated with the input of the pre-calculated value and the contributions of the joining members. In the leaving algorithm, the remaining members cooperate to update the membership and the contributions of the new joining members. The new session key is calculated with the input of the pre-calculated values in the previous session and the updated contributions. Compared to the protocol [?], the dynamic version of [?, 18] is proven secure under the standard model. Besides, each group member in this protocol is capable of detecting the presence of the malicious insiders without recognizing who behave improperly.

2.2 Security Model of Group Key Exchange

Bresson *et al.* [11] suggested a first formal security model called BCPQ model for authenticated GKE protocols in static setting. In the paper, they defined AKE security and Mutual authentication (MA) security. AKE security guarantees that the active adversary who does not participate in the session cannot distinguish the common secret key from a random number. Active adversary can control underlying communication channel by eavesdropping and modifying messages. MA security ensures that only legitimate participants can compute identical session group secret key. After that, Katz and Yung [22] revised this model to compile unauthenticated GKE protocol into authenticated GKE protocol. They proved the security of Burmester-Desmedt protocol [12] in the presence of a passive adversary who can only eavesdrop messages and make a compiler from GKE to authenticated GKE with an active adversary. After that, Katz and Shin [21] proposed another compiler which can transform an implicitly secure authenticated GKE into a secure authenticated GKE resistant to insider attacks, in the universally-composable (UC) model.

For dynamic setting, Bresson *et al.* [9,10] suggested two formal security models for authenticated GKE protocols depending on the power of corruption and the presence of MA security. Compared to weak corruption model, with strong corruption model, the adversary $\mathcal{A}$ is capable of revealing the long-term key as well as the short-term ephemeral secrets of the protocol instance. Moreover, the security notion of forward secrecy is also defined in this security model.

2.3 Lattice-based Key Exchange

By modifying Diffie-Hellman key exchange protocol [15] into RLWE setting, Ding *et al.* [17] suggested the first lattice-based key exchange protocol in 2012. Following this research, numerous work [1–3, 6–8, 14, 16, 19, 27–29, 34, 35] studied on constructing key exchange protocols based on lattice but most of them are focusing on two-party key exchange.

For lattice-based GKE protocol, Ding *et al.* [15] suggested the natural extension to GKE protocol based on their key exchange protocol using the GKE compiler by Bresson *et al.* [9]. After that, Yang *et al.* [34] proposed the first provably-secure (authenticated) GKE protocol based on the hardness of LWE/RLWE assumption and security property of secure sketch in the random oracle model. For secure sketch, trusted authority is necessary and this protocol is not contributory.

Recently, Apon *et al.* [2] proposed the first constant-round authenticated GKE protocol based on the hardness of RLWE assumption, without trusted third party. This protocol uses Katz-Yung compiler for authentication and it is also contributory since they adopt the protocol in [12].

3 Preliminaries

3.1 Notation

Let $\mathbb{Z}$ be the set of integers and $[N] = \{0, 1, 2, \cdots N-1\}$. For a set A, $x_i \leftarrow A$ denotes a uniformly random sampling of $x_i \in A$. Let $\chi(E)$ stand for a probability of a set E of events occurs under a distribution χ. We set $\mathrm{Supp}(\chi) = \{\epsilon : \chi(\epsilon) \neq 0\}$ and let $\bar{E}$ be the complement of an event set E. Let $f(a, b)$ be a function f on a and b. We say a function f is negligible when $f = O(n^{-c})$ for all $c > 0$.

Given a polynomial p, $(p)_j$ denotes the j-th coefficient of p. We use $\log(x)$ and $\exp(x)$ to denote $\log_2(x)$ and e^x, respectively. We denote P_i and $P[0, 1, \cdots, k] = \{P_0, P_1, \cdots, P_k\}$ for i-th party of a protocol and an array of parties, respectively.

3.2 Ring Learning with Errors

Informally, the decisional Ring Learning with Errors (RLWE) problem [26] is that given m independent samples in $R_q \times R_q$ which is defined below, distinguish each sample is either a noisy product with a secret element s of R_q or uniformly random element of R_q. More precisely, RLWE problem is defined as follows: given a tuple (R, q, χ, l) where $R = \mathbb{Z}[x]/(f(x))$ is a polynomial ring for an irreducible polynomial $f(x)$, q is a positive integer modulus defining a quotient ring $R_q = R/qR$, $\chi = (\chi_s; \chi_e)$ is a pair of noise distributions over R_q, and l is the number of samples given to the adversary, distinguish each sample is either (1) $(a, as + e) \in R_q \times R_q$ for some uniform element $a \leftarrow R_q$, secret key $s \leftarrow \chi_s$ and error $e \leftarrow \chi_e$ or (2) uniformly sampled from $R_q \times R_q$.

We let $\mathsf{Adv}^{RLWE}_{n,q,\chi_s,\chi_e,l}(\mathcal{B})$ denote the advantage of algorithm $\mathcal{B}$ in distinguishing these two cases, and defining $\mathsf{Adv}^{RLWE}_{n,q,\chi_s,\chi_e,l}(t)$ to be the maximum advantage of any algorithm running in time t. If $\chi = \chi_s = \chi_e$, we write $\mathsf{Adv}^{RLWE}_{n,q,\chi,l}$ for simplicity.

3.3 Rényi Divergence

For two discrete probability distributions P and Q with $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$, their Rényi divergence is defined as

$$\mathrm{RD}_2(P||Q) = \sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^2}{Q(x)}.$$

Rényi divergence measures closeness of two probability distributions and it is widely used in cryptographic research [4, 25, 26, 33]. We introduce some important results related to Rényi divergence that can be used in our protocol.

Proposition 1. *[4] For discrete distributions P and Q with $Supp(P) \subseteq Supp(Q)$, let $E \subseteq Supp(Q)$ be an arbitrary event. We have*

$$Q(E) \geq P(E)^2 / RD_2(P \| Q)$$

.

Roughly, the proposition says that if $RD_2(P\|Q)$ is bounded by some polynomial, then any event set E that occurs with negligible probability $Q(E)$ under Q also occurs with negligible probability $P(E)$ under P.

Lemma 1. *[4] Let $m, q, \lambda \in \mathbb{Z}$ and fix a bound $\beta_{\mathsf{Rényi}}$ and σ with $\beta_{\mathsf{Rényi}} < \sigma < q$. Let $e \in \mathbb{Z}$ satisfying $|e| \leq \beta_{\mathsf{Rényi}}$. Then*

$$RD_2((e + D_{\mathbb{Z},\sigma})^m || D^m_{\mathbb{Z},\sigma}) \leq \exp(2\pi m(\beta_{\mathsf{Rényi}}/\sigma)^2)$$

where χ^m means that we sample m times independently from the distribution χ. Moreover, if we take $\sigma = \Omega(\beta_{\mathsf{Rényi}}\sqrt{m/\log\lambda})$ with security parameter λ, we can deduce $RD_2((e + D_{\mathbb{Z},\sigma})^m || D^m_{\mathbb{Z},\sigma}) \leq poly(\lambda)$.

3.4 Generic Key Reconciliation Algorithm

The concept of key reconciliation was first introduced by Ding *et al.* [17] to handle error between two approximately agreed ring elements in their lattice-based key exchange protocol. Then, it has been used in several works on lattice-based two-party key exchange protocol [1, 6, 8, 27, 35].

From Apon *et al.*'s paper [2], we describe a generic key reconciliation algorithm which is performed between two-party in one-round.

A key reconciliation $\mathsf{KeyRec} = (\mathsf{recMsg}, \mathsf{recKey})$ allows two parties to derive the same key from approximately agreed ring elements. One of two participants runs the first algorithm recMsg taking the security parameter λ and a ring element $b \in R_q$ and outputs rec and a key $k \in \{0,1\}^\lambda$. The other participant runs recKey taking rec and a ring element $b' \in R_q$ and outputs a key value $k' \in \{0,1\}^\lambda$.

We say a key exchange protocol works correctly when two participants have the same key (i.e. $k = k'$). To hold this equality, b and b' have to be sufficiently close. Especially, if $b - b'$ are bounded by some value β_{Rec} and two participants run KeyRec algorithm, then they share the same key except with negligible probability.

Security is defined by the indistinguishability between a key k, result of key exchange, and uniformly random value. Formally, an attacker $\mathcal{A}$ is computationally infeasible to distinguish two distribution,

$$\{(\mathsf{rec}, k) : b \leftarrow R_q; (\mathsf{rec}, k) \leftarrow \mathsf{recMsg}(1^\lambda, b)\}_{\lambda \in \mathbb{N}},$$

$$\{(\mathsf{rec}, k') : b \leftarrow R_q; (\mathsf{rec}, k) \leftarrow \mathsf{recMsg}(1^\lambda, b); k' \leftarrow \{0,1\}^\lambda\}_{\lambda \in \mathbb{N}}$$

For a fixed value of λ, we denote the advantage of adversary $\mathcal{A}$ in distinguishing these two distributions by $\mathsf{Adv}_{\mathsf{KeyRec}}(\mathcal{A})$, and the maximum advantage of any such adversary running in time t by $\mathsf{Adv}_{\mathsf{KeyRec}}(t)$.

4 Security Model

We describe the adversary model of Bresson *et al.* [9]. This model is suitable in our protocol since it covers authenticated GKE with dynamic setting.

Let $\mathcal{P} = P[0, 1, \cdots, N-1]$ be a set of N parties. Any subset of $\mathcal{P}$ wishes to establish a session key. We identify the execution of protocols for (authenticated) GKE or Join/Leave for inclusion/exclusion of a party or a set of parties as different sessions. We assume adversary never participates as a party in the protocol.

This adversary model allows concurrent execution of the protocol. The interaction between the adversary $\mathcal{A}$ and the protocol participants happens via oracle queries only. We denote a set of session identity and partner identity as sid_P^i and pid_P^i, respectively. For a party $(U_j, i_j) \in S$, we set $\mathsf{sid}_{U_j}^{i_j} = S = \{(U_0, i_0), \cdots, (U_{l-1}, i_{l-1})\}$ and $\mathsf{pid}_{U_j}^{i_j} = U[0, 1, \cdots, l-1]$ when $U[0, 1, \cdots, l-1]$ wish to agree a common secret key.

We assume that the adversary has full control over all communications in the network. All information that the adversary gets is written in a transcript since a transcript consists of all public information flowing across the network. The following oracles model adversary's interaction with the protocol participants:

- $\mathsf{Send}(U, i, m)$: This oracle models an active attack where the adversary has full control on the communication. The output is the reply by (U, i) upon the receipt of message m. The adversary can initiate the protocol with partners $U[0, 1, \cdots, l-1]$ where $l \leq N$, by invoking $\mathsf{Send}(U, i, U[0, 1, \cdots, l-1])$.
- $\mathsf{Execute}(S)$: This oracle models passive attacks where the attacker eavesdrops on honest execution of the protocol and outputs the transcript of the execution. A transcript consists of all messages exchanged.
- $\mathsf{Join}(S, S_1)$: This oracle models the addition of a set of party instances S_1 in the group S, where all parties in S or S_1 are in $\mathcal{P}$. For S, Execute oracle has already been queried. The output is the transcript generated by the honest execution of algorithm Join. If $\mathsf{Execute}(S)$ is not preprocessed, the adversary gets no output.
- $\mathsf{Leave}(S, S_2)$: This oracle models the removal of a set of party instances $S_2 \subseteq S$ from the group S where all parties are in $\mathcal{P}$. Similaar to $\mathsf{Join}(S, S_1)$, if $\mathsf{Execute}(S)$ is not preprocessed, the adversary gets no output. Otherwise, algorithm Leave is invoked. The adversary obtains the transcript from the honest execution of algorithm Leave.
- $\mathsf{Reveal}(U, i)$: This oracle models the misuse of the session keys, *i.e* known session key attack. This query outputs session key sk_U^i.

– $\mathsf{Corrupt}(U)$: This oracle models (perfect) forward secrecy. It outputs the long-term secret key of player U. The adversary model that we adopt is a weak-corruption model where ephemeral keys or internal states of protocol participants are not corrupted.
– $\mathsf{Test}(U,\, i)$: We can query this oracle only once during the adversary's execution. A bit $b \in \{0,1\}$ is chosen uniformly at random. The adversary gets sk if $b = 1$ and a random session key sk' if $b = 0$. This oracle checks the adversary's ability to distinguish a real session key from random.

An adversary, which can access Execute, Join, Leave, Reveal, Corrupt and Test oracles, is considered as "passive" while an "active" adversary has full access to above-mentioned oracles including Send oracle. (For static case, Join or Leave queries doesn't need to be considered.)

The adversary can ask Send, Execute, Join, Leave, Reveal and Corrupt queries several times, but Test query is asked only once for a fresh instance. We say that an instance (U, i) is *fresh* if none of the following occurs:

(1) the adversary queried $\mathsf{Reveal}(U, i)$ or $\mathsf{Reveal}(U', j)$ with $U' \in \mathsf{pid}_U^i$,
(2) the adversary queried $\mathsf{Corrupt}(U')$ (with $U' \in \mathsf{pid}_U^i$) before a query of the form $\mathsf{Send}(U, i, \star)$ or $\mathsf{Send}(U', j, \star)$ where $U' \in \mathsf{pid}_U^i$.

Adversary outputs a guess b'. Adversary wins the game if $b = b'$ where b is chosen bit from Test oracle.

Let Succ denote the event that the adversary $\mathcal{A}$ wins the game for a protocol XP. We define $\mathsf{Adv}_{\mathcal{A},\mathsf{XP}} := |2 \cdot \Pr[\mathsf{Succ}] - 1|$ to be the advantage of the adversary $\mathcal{A}$ in attacking the protocol XP.

The protocol XP provides *secure unauthenticated/authenticated GKE* (KE/AKE) security if there is no polynomial time passive/active adversary with non-negligible advantage, respectively.

Let t be the running time for adversary and q_E, q_J, q_L, q_S be the number of queries to Execute, Join, Leave, Send oracles, respectively. $\mathsf{Adv}_{\mathsf{XP}}^{\mathsf{KE}}(t, q_E)$ is the maximum advantage of any passive adversary attacking protocol XP and $\mathsf{Adv}_{\mathsf{XP}}^{\mathsf{AKE}}(t, q_E, q_S)$ and $\mathsf{Adv}_{\mathsf{XP}}^{\mathsf{AKE}}(t, q_E, q_J, q_L, q_S)$ are the maximum advantage of any active adversary attacking protocol XP.

5 Dynamic (Authenticated) Group Key Exchange

In this section, we describe our (authenticated) GKE protocol with static and dynamic membership.

As we mentioned earlier, for the basic static setting, we follow the very similar procedure as Apon *et al.*'s scheme. We run $\mathsf{KeyRec} = (\mathsf{recMsg}, \mathsf{recKey})$ as a subroutine. We also consider two security parameters for security analysis, λ and ρ. λ is used for security proof and ρ is used for correctness check.

Algorithm 1: $\mathsf{STUG}(P\,[0,1,\cdots,N-1]\,,a,\mathcal{H},\sigma_1,\sigma_2)$

(Round 1) For each party P_i for $i=0$ to $N-1$, do the following in parallel.

1. Computes $z_i = as_i + e_i$ where $s_i, e_i \leftarrow \chi_{\sigma_1}$;
2. Broadcasts z_i;

(Round 2) For $i=0$ to $N-1$, do the following in parallel.

1. If $i=0$, party P_0 samples $e'_0 \leftarrow \chi_{\sigma_2}$ and otherwise, party P_i samples $e'_i \leftarrow \chi_{\sigma_1}$;
2. Each party P_i broadcasts $X_i = (z_{i+1} - z_{i-1})\,s_i + e'_i$;

(Round 3) For party P_{N-1} only.

1. Samples $e''_{N-1} \leftarrow \chi_{\sigma_1}$ and computes $Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e''_{N-1}$;
2. For $j=1$ to $N-1$, computes $Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}$;
3. Calculates $b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}$;
4. Runs $\mathsf{recMsg}()$ to output $(\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1})$;
5. Broadcasts rec and gets session key as $\mathsf{sk}_{N-1} = \mathcal{H}(k_{N-1})$;

(Key Computation) For party P_i $(i \neq N-1)$.

1. Computes $Y_{i,i} = X_i + z_{i-1}s_i$;
2. For $j=1$ to $N-1$, computes $Y_{i,i+j} = X_{i+j} + Y_{i,i+(j-1)}$;
3. $b_i = \sum_{j=0}^{N-1} Y_{i,i+j}$;
4. Runs $\mathsf{recKey}()$ to output $k_i = \mathsf{recKey}\,(b_i, \mathsf{rec})$ and gets session key as $\mathsf{sk}_i = \mathcal{H}(k_i)$;

5.1 Unauthenticated Group Key Exchange

In the static setting, given $R_q = \mathbb{Z}_q\,[x]\,/(x^n+1)$ and $a \leftarrow R_q$, all parties calculate the partial numbers X_i and $Y_{i,j}$ and agree on "close" values $b_0 \approx b_1 \approx \cdots \approx b_{N-1}$ after the second round. Then, party P_{N-1} runs recMsg algorithm from KeyRec to allow all parties to get a common value $k = k_0 = k_1 = \cdots = k_{N-1}$.

Since we only show that k is difficult to compute for a passive adversary in the security proof, we hash k using random oracle $\mathcal{H}$ to get the session group secret key sk, which is indistinguishable from random. More detail description of unauthenticated GKE is given in Algorithm 1.

5.2 Authenticated Group Key Exchange

To authenticate the unauthenticated one in Section 5.1, we use a digital signature scheme $\mathsf{DSig} = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ where $\mathcal{K}$ is the key generation algorithm with output (sk_i, pk_i) for each party, $\mathcal{S}$ outputs a signature δ_i for a message m_i, and $\mathcal{V}$ outputs whether the input signature is valid or not.

Following Dutta-Barua protocol [18], at the start of the session, P_i doesn't need to know the entire session identity set $\mathsf{sid}_{P_i}^{d_i}$. As protocol proceeds, we build this set from partial session identity set $\mathsf{psid}_{P_i}^{d_i}$. Initially, $\mathsf{psid}_{P_i}^{d_i} = \{(P_i, d_i)\}$ and after completing the procedure, it becomes the full session identity set $\mathsf{sid}_{P_i}^{d_i}$. We assume that all parties know its partner identity $\mathsf{pid}_{P_i}^{d_i}$. More detail description of authenticated GKE is given in Algorithm 2.

Algorithm 2: $\mathsf{STAG}(P[0,1,\cdots,N-1], a, \mathcal{H}, \mathcal{S}, \sigma_1, \sigma_2)$

(Round 1) For each party P_i for $i = 0$ to $N-1$, do the following in parallel.

1. Sets partial session-identity $\mathsf{psid}_{P_i}^{d_i} = \{P_i, d_i\}$;
2. Computes $z_i = as_i + e_i$ where $s_i, e_i \leftarrow \chi_{\sigma_1}$;
3. Sets $m_i = P_i \mid 1 \mid z_i$ and $\delta_i = \mathcal{S}(m_i)$;
4. Broadcasts $m_i \mid \delta_i$;

(Round 2) For each party P_i for $i = 0$ to $N-1$, do the following in parallel.

1. Verifies δ_{i-1} of m_{i-1} and δ_{i+1} of m_{i+1} and proceeds only if both signatures are valid (Otherwise, aborts);
2. If $i = 0$, party P_0 samples $e_0' \leftarrow \chi_{\sigma_2}$ and otherwise, party P_i samples $e_i' \leftarrow \chi_{\sigma_1}$;
3. Computes $X_i = (z_{i+1} - z_{i-1})\, s_i + e_i'$;
4. Sets $m_i' = P_i \mid 2 \mid X_i \mid d_i$ and $\delta_i' = \mathcal{S}(m_i')$ and broadcasts $m_i' \mid \delta_i'$;

(Round 3) For party P_{N-1} only.

1. Verifies all δ_j' of m_j' where $j \neq N-1$ and proceeds only if both signatures are valid (Otherwise, aborts);
2. Extracts d_j from m_j' and sets $\mathsf{psid}_{P_{N-1}}^{d_{N-1}} = \mathsf{psid}_{P_{N-1}}^{d_{N-1}} \bigcup \{(P_j, d_j)\}$;
3. Samples $e_{N-1}'' \leftarrow \chi_{\sigma_1}$ and computes $Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e_{N-1}''$;
4. For $j = 1$ to $N-1$, computes $Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}$;
5. Calculates $b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}$;
6. Runs $\mathsf{recMsg}(\cdot)$ to output $(\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1})$;
7. Broadcasts rec and gets session key as $\mathsf{sk}_{N-1} = \mathcal{H}(k_{N-1})$;

(Key Computation) For party P_i $(i \neq N-1)$.

1. Verifies all δ_j' of m_j' where $j \neq i$ and proceeds only if both signatures are valid (Otherwise, aborts);
2. Extracts d_j from m_j' and sets $\mathsf{psid}_{P_i}^{d_i} = \mathsf{psid}_{P_i}^{d_i} \bigcup \{(P_j, d_j)\}$;
3. Computes $Y_{i,i} = X_i + z_{i-1}s_i$;
4. For $j = 1$ to $N-1$, computes $Y_{i,i+j} = X_{i+j} + Y_{i,i+(j-1)}$;
5. $b_i = \sum_{j=0}^{N-1} Y_{i,i+j}$;
6. Runs $\mathsf{recKey}()$ to output $k_i = \mathsf{recKey}(b_i, \mathsf{rec})$ and gets session key as $\mathsf{sk}_i = \mathcal{H}(k_i)$;

5.3 Dynamic Group Key Exchange

Join Algorithm In the dynamic setting, we require another hash function $\mathcal{H}_1$ that outputs a value from the distribution χ_{σ_1}. This function is required since we cannot apply the original common secret sk as a secret key of $U_1 = P_1$ due to its type difference. Instead, we apply $\mathcal{H}_1(\mathsf{sk})$ as a secret key of $U_1 = P_1$.

If we assume that there are M parties in the set $P[N, N+1, \cdots, N+M-1]$ who wish to join the group $P[0, 1, \cdots, N-1]$ who already shared the common secret key sk, we make a new ring that consists of three parties P_0, P_1, P_{N-1} from $P[0, 1, \cdots, N-1]$ and all parties from the set $P[N, N+1, \cdots, N+M-1]$. P_1 chooses the original session key sk as his ephemeral key $\overline{s}_1$.

For authenticated version $\mathsf{A.Join}$ algorithm, we consider partial session-identity as STAG algorithm but we assume that $\mathsf{psid}^{d_i}_{P_i} = \mathsf{psid}^{d_i}_{P_i} \bigcup \{\{(P_j, d_j) \mid j = 1 \text{ to } N-2\}$ if $P_i (i = 0, 1, \text{ or } N \le i \le N+M-1)$ verifies $\overline{\delta}'_1$ of $\overline{m}'_1$. We assume this since the ephemeral keys $\overline{s}_1$ and $\overline{z}_1$ are from the session key sk among the group $P\,[0, 1, \cdots, N-1]$.

Signature generation and verification happen by switching STUG algorithm into STAG algorithm, also these operations happen in Round 2 of $\mathsf{A.Join}$ algorithm when $\overline{z}_1$, $\overline{z}_3$, and $\overline{X}_i$ are delivered to group $P\,[2, \cdots, N-2]$.

By modifying the concept of $\mathsf{psid}^{d_i}_{P_i}$ slightly, we can achieve a common session identity $\mathsf{sid}^{d_i}_{P_i} = \{(P_j, d_j) \mid j \in [N+M]\}$ for parties in $P[0, 1, \cdots, N+M-1]$ while Dutta-Barua only provides a common session identity $\mathsf{sid}^{d_i}_{U_i} = \{(U_j, d_j) \mid j \in [\overline{N}]\}$ for parties in $U[0, 1, \cdots, \overline{N}-1]$ where $\overline{N} = M+3$.

Algorithm 3: $\mathsf{U.Join}(P\,[0, 1, \cdots, N-1]\,, P[N, N+1, \cdots, N+M-1])$

(Round 1) Rearrange the order with a new array of $\overline{N} = M+3$ parties

1. $U_0 = P_0, U_1 = P_1, U_2 = P_{N-1}, \overline{s}_0 = s_0, \overline{s}_1 = \mathcal{H}_1(\mathsf{sk}), \overline{s}_2 = s_{N-1}$ and for $1 \le i \le \overline{N}-3, U_{i+3} = P_{N-1+i}$;
2. Let $U[0, 1, \cdots, \overline{N}-1]$ be a new ring that we run in **(Round 2)**;

(Round 2) Run STUG algorithm.

1. Group $U[0, 1, \cdots, \overline{N}-1]$ runs STUG;
2. U_i calculates $\overline{z}_i$ during the 1st round of STUG and broadcasts it;
3. U_0 and U_2 during 1st round of STUG additionally sends $\overline{z}_1$ and $\overline{z}_3$ to all parties in $P\,[2, \cdots, N-2]$;
4. U_i calculates $\overline{X}_i$ during the 2nd round of STUG sends $\overline{X}_i$ to all parties in $P\,[0, \cdots, N+M-1]$;
5. After the 3rd round of STUG, $U_{\overline{N}-1}$ sends $\overline{\mathsf{rec}}$ to all parties in $P\,[0, \cdots, N+M-1]$;

(Key Computation) For party P_i $(2 \le i \le N-2)$.

1. Computes $\overline{Y}_{i,2} = \overline{X}_2 + \overline{z}_2\overline{s}_1 = \overline{X}_2 + \overline{z}_2 \cdot \mathcal{H}_1(\mathsf{sk})$;
2. For $j = 1$ to $\overline{N}-2$, computes $\overline{Y}_{i,2+j} = \overline{X}_{2+j} + \overline{Y}_{i,2+(j-1)}$;
3. $b'_i = \sum_{j=0}^{\overline{N}-1} \overline{Y}_{i,j}$;
4. Runs $\mathsf{recKey}(\cdot,\cdot)$ to output $\overline{k}_i = \mathsf{recKey}\left(\overline{b}_i, \overline{\mathsf{rec}}\right)$ and gets session key as $\overline{\mathsf{sk}}_i = \mathcal{H}(\overline{k}_i)$;

Leave Algorithm Let the set of parties $P_{l_1}, P_{l_2}, \cdots P_{l_M}$ want to leave the group $P\left[0, 1, \cdots, N-1\right]$. Then, the new group becomes $P' = P\left[0, \cdots, l_1 - L\right] \cup P\left[l_1 + R, \cdots, l_2 - L\right] \cup \cdots \cup P\left[l_M + R, \cdots, N-1\right]$. Instead of $l_i - 1$ and $l_i + 1$, we use $l_i - L$ and $l_i + R$ since there might be consecutive parties who want to leave the group $P\left[0, 1, \cdots, N-1\right]$. *e.g.*, if $P_l, P_{l-1}, P_{l-2}, \cdots, P_{l-(j-1)}$ are consecutive parties who want to leave, then $P_{l-L} = P_{l-j}$.

After making a new group P', we simply relabel orders to make a new array $U\left[0, 1, \cdots, N-M-1\right]$ of the parties in the protocol and run $\mathsf{U.Leave}$ algorithm for $U\left[0, 1, \cdots, N-M-1\right]$ based on the remaining parties and run STUG algorithm. For authenticated version $\mathsf{A.Leave}$, we simply apply STAG algorithm instead of STUG algorithm.

Our dynamic unauthenticated and authenticated GKE protocols DRUG and DRAG consist of three algorithms, (STUG, $\mathsf{U.Join}$, $\mathsf{U.Leave}$) and (STAG, $\mathsf{A.Join}$, $\mathsf{A.Leave}$) as a subroutine, respectively.

6 Security Analysis

In this section, we check the correctness of our protocol and give a full security proof using the security model by Bresson *et al.* [9]. Our proof techniques is based on Apon *et al.*'s protocol [2] and Dutta-Barua protocol [18].

In this section, we check the correctness of our protocol and give a full security proof using the security model by Bresson *et al.* [9]. Our proof techniques is based on Apon *et al.*'s protocol [2] and Dutta-Barua protocol [18].

6.1 Correctness Proof

In Theorem 1, we give a condition that our GKE is correct. Most part of our correctness proof follow Apon *et al.*'s correctness proof but there are some modification on error bound.

Note that correctness of GKE protocol is all parties agree on the same secret key. Lemmas 2 and 3 and its proofs are from Apon *et al.*'s paper [2].

Lemma 2. *[2] Given s_i for all i defined in the group key exchange protocol, fix $c = \sqrt{\frac{2\rho}{\pi \log(e)}}$ and let bound_ρ be the event that for all $i \in [N]$ and all coordinate $j \in [n]$, $|(s_i)_j|, |(e_i)_j|, |(e'_i)_j|, |(e''_{N-1})_j| \leq c\sigma_1$ except $|(e'_0)_j| \leq c\sigma_2$. Then*

$$\Pr[\mathsf{bound}_\rho] \geq 1 - 2^{\rho}.$$

Proof. Since the complementary error function $erfc(x) = \frac{2}{\pi}\int_x^\infty \exp(-t^2)dt \leq \exp(-x^2)$, we get

$$\begin{aligned}\Pr[v \leftarrow D_{\mathbb{Z}_q,\sigma}; |v| \geq c\sigma + 1] &\leq 2 \sum_{x=\lfloor c\sigma+1 \rceil}^{\infty} D_{\mathbb{Z}_q,\sigma}(x) \\ &\leq \frac{2}{\sigma}\int_{c\sigma}^{\infty} \exp(\frac{-\pi x^2}{\sigma^2})dx \\ &= \frac{2}{\pi}\int_{\frac{\sqrt{\pi}}{\sigma}(c\sigma)}^{\infty} \exp(-t^2)dt \leq \exp(-c^2\pi).\end{aligned}$$

Then we have $3nN$ samplings from $D_{\mathbb{Z}_q,\sigma_1}$ and n samplings from $D_{\mathbb{Z}_q,\sigma_2}$ in our protocol. Under the assumption that $3nN + n \leq \exp(c^2\pi/2)$, we have

$$\begin{aligned}\Pr[\mathsf{bound}_\rho] &= (1 - \Pr[v \leftarrow D_{\mathbb{Z}_q,\sigma_1}; |v| \geq c\sigma_1 + 1])^{3nN} \\ &\quad \cdot (1 - \Pr[e_0' \leftarrow D_{\mathbb{Z}_q,\sigma_2}; |v| \geq c\sigma_2 + 1])^{n} \\ &\geq 1 - (3nN + n) \cdot \exp(-c^2\pi) \geq 1 - \exp(c^2\pi/2) \\ &\geq 1 - 2^{-\rho}.\end{aligned}$$

Lemma 3. *[2] Given bound_ρ defined in Lemma 2, let $\mathsf{product}_{s_i,\, e_j}$ be the event that for all v-th coordinate, $|(s_i \cdot e_j)_v| \leq \sqrt{n}\rho^{3/2}\sigma_1^2$. Then*

$$\Pr[\mathit{product}_{s_i \cdot e_j} \mid \mathit{bound}_\rho] \geq 1 - n \cdot 2 \cdot 2^{-2\rho}$$

Proof. Note that for $l \in [n]$, $(s_i)_l$ denotes the l-th coefficient of s_i and we can express $s_i = \sum_{l=0}^{n-1}(s_i)_l X^l$. Since we take $X^n + 1$ as modulus of R, $(s_i e_j)_l = \sum_{k=0}^{n-1}(s_i)_k(e_j)^*_{l-k}X^l$ where $(e_j)^*_{l-k}$ is $(e_j)_{l-k}$ if $l-k \geq 0$ and $-(e_j)_{l-k}$ otherwise. Thus, under bound_ρ, specifically $|(s_i)_l|,\ |(e_j)_l| \leq c\sigma_1$ where $c = \sqrt{\frac{2\rho}{\pi \cdot \log(e)}}$, by Hoeffding's inequality [20], we can get

$$\begin{aligned}&\Pr[|(s_i e_j)_l| \geq \gamma \mid \mathsf{bound}_\rho] \\ &\qquad = \Pr\left[\left|\sum_{k=0}^{n-1}(s_i)_k(e_j)_{l-k}\right| \geq \gamma \mid \mathsf{bound}_\rho\right] \\ &\qquad \leq 2 \cdot \exp\left(\frac{-2\gamma^2}{n(2c^2\sigma_1^2)^2}\right).\end{aligned}$$

(Note that $(s_i)_k(e_j)_{l-k}$ is an independent random variable with mean 0 in interval $[-c^2\sigma_1^2,\ c^2\sigma_1^2]$.) If we take $\gamma = \sqrt{n}\rho^{3/2}\sigma_1^2$, then we get

$$\Pr[|(s_i e_j)_l| \geq \gamma \mid \mathsf{bound}_\rho] \leq 2 \cdot \exp(\frac{-\rho^3}{2c^4}) \leq 2^{-2\rho+1}$$

Thus, after union all bound, we have

$$\begin{aligned}\Pr[\mathsf{product}_{s_i,\, e_j} \mid \mathsf{bound}_\rho] &= \Pr[\forall l,\ |(s_i e_j)_l| \leq \sqrt{n}\rho^{3/2}\sigma_1^2] \\ &\geq 1 - n \cdot 2 \cdot 2^{-2\rho}.\end{aligned}$$

Theorem 1. *For a fixed ρ, and assume that*

$$(N-1)N/2 \cdot \sqrt{n}\rho^{3/2}\sigma_1^2 + (N(N+1)/2 + N)\,\sigma_1 + (N-2)\sigma_2 \le \beta_{\mathsf{Rec}}.$$

Then all participants in a group have the same key except with probability at most $2^{-\rho+1}$.

Proof. As mentioned in Section 3.4, we will show that all parties have the same secret key except with negligible probability. To hold this, we claim that if for all $i \in [N]$ and $j \in [n]$ j-th coefficient of $|b_{N-1} - b_i| \le \beta_{\mathsf{Rec}}$, then $k_i = k_{N-1}$. After some tedious computation, we have

$$\begin{aligned} b_{N-1} - b_i = Ne''_{N-1} &+ \sum_{j=0}^{N-1}(N-j)(e'_{N-1+j} - e'_{i+j}) \\ &+ \sum_{j=0}^{N-2}(N-1-j)\{(e_{N+j}s_{N-1+j} - e_{N-1+j}s_{N+j}) \\ &- (e_{i+j+1}s_{i+j} - e_{i+j}s_{i+j+1})\}. \end{aligned}$$

Now observe how many terms are in $b_{N-1} - b_i$. There are at most $(N-1)N/2$ terms in form of $s_i \cdot e_j$, at most $N(N+1)/2$ terms in form of e'_k sampled from χ_{σ_1}, at most $N-2$ terms of e'_0 sampled from χ_{σ_2}, and N terms of e''_{N-1}. Sum of these at most terms is less than Apon et al.'s terms.

Let $\mathsf{product}_{\mathsf{ALL}}$ be the event that for all terms in form of $s_i \cdot e_j$, each coefficient of this form is bounded by $\sqrt{n}\rho^{3/2}\sigma_1^2$. Under an assumptiont that $2n(N-1)N/2 \le 2^\rho$, by Lemma 3 we can get

$$\Pr[\overline{\mathsf{product}_{\mathsf{ALL}}} \,|\, \mathsf{bound}_\rho] \le \frac{(N-1)N}{2} \cdot n \cdot 2^{-2\rho+1} \le 2^{-\rho}$$

Denote fail by the event that at least one of parties does not agree on the same key. Given a condition that $(N-1)N/2 \cdot \sqrt{n}\rho^{3/2}\sigma_1^2 + (N(N+1)/2+N)\sigma_1 + (N-2)\sigma_2 \le \beta_{\mathsf{Rec}}$, by Lemma 2 and the above inequality we have

$$\begin{aligned} \Pr[\mathsf{fail}] &= \Pr[\mathsf{fail} \,|\, \mathsf{bound}_\rho] \cdot \Pr[\mathsf{bound}_\rho] \\ &\quad + \Pr[\mathsf{fail} \,|\, \overline{\mathsf{bound}_\rho}] \cdot \Pr[\overline{\mathsf{bound}_\rho}] \\ &\le \Pr[\overline{\mathsf{product}_{\mathsf{ALL}}} \,|\, \mathsf{bound}_\rho] \cdot 1 + 1 \cdot \Pr[\overline{\mathsf{bound}_\rho}] \\ &\le 2 \cdot 2^{-\rho}. \end{aligned}$$

Therefore, all parties agree on the same secret key except with probability $2 \cdot 2^{-\rho}$.

From the result of Theorem 1, the number of error terms in our protocol is smaller than Apon *et al.*'s protocol. Then, the probability $\Pr_{\mathsf{STUG}}[\mathsf{AbortKey}]$ of the event $\mathsf{AbortKey}$ that error between b_i's exceeds β_{Rec} in our protocol is smaller than the probability $\Pr_{\mathsf{Apon}}[\mathsf{AbortKey}]$ in Apon *et al.*'s protocol. Thus, our protocol has higher probability to have the common secret key between protocol participants.

6.2 Security Proof

We write Theorems 2, 3 and 4 to show that our dynamic key exchange protocol $\mathsf{DRUG} = (\mathsf{STUG}, \mathsf{U.Join}, \mathsf{U.Leave})$ (or $\mathsf{DRAG} = (\mathsf{STAG}, \mathsf{A.Join}, \mathsf{A.Leave})$) is secure in the random oracle model based on hardness of RLWE assumption. We prove all theorems in this section.

Theorem 2. *For unauthenticated GKE protocol STUG, $2N\sqrt{n}\lambda^{3/2}\sigma_1^2 + (N-1)\sigma_1 \le \beta_{\mathsf{Rényi}}$ and $\sigma_2 = \Omega\left(\beta_{\mathsf{Rényi}}\sqrt{n/\log\lambda}\right)$. Then, we have the following:*

$$\mathsf{Adv}^{\mathsf{KE}}_{\mathsf{STUG}}(t, q_E) \le 2^{-\lambda+1} + \sqrt{\mathsf{Adv}_{\mathsf{Exp\text{-}1}} \cdot \frac{\exp\left(2\pi n(\beta_{\mathsf{Rényi}}/\sigma_2)^2\right)}{1-2^{-\lambda+1}}}$$

where $\mathsf{Adv}_{\mathsf{Exp\text{-}1}} = N \cdot \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},3}(t_1) + \mathsf{Adv}_{\mathsf{KeyRec}}(t_2) + \frac{q_E}{2^\lambda}$, $t_1 = t + \mathcal{O}(N \cdot t_{ring})$, and $t_2 = t + \mathcal{O}(N \cdot t_{ring})$ such that t_{ring} is the maximum time required to make operations in R_q.

Proof. Let $\mathcal{A}$ be an adversary that breaks the protocol STUG. From this, we construct an adversary $\mathcal{B}$ that solves RLWE problem with non-negligible advantage. Since we do not have any long-term secret key in our protocol STUG, $\mathsf{Corrupt}$ can be ignored and the protocol achieves the forward secrecy.

Let Query be the event that k_{N-1} is among the adversary $\mathcal{A}$'s random oracle queries and $\Pr_i[\mathsf{Query}]$ be the probability of Query in Experiment i.

Then, by a sequence of experiments, we show that an efficient adversary who queries the random oracle in Ideal experiment with at most negligible probability can query the random oracle in Exp_0 experiment. For Ideal experiment, the input k_{N-1} is chosen uniformly random while k_{N-1} is chosen by the honest execution of STUG in $\mathsf{Exp}_0\mathsf{Exp}_1$ experiment.

Experiment 0. This is the original experiment that is equal to the procedure of STUG.

$$\mathsf{Exp}_0 := \left\{ \begin{array}{ll} a \leftarrow R_q; & \\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i \text{ for } i \in [N]; & \\ e'_0 \leftarrow \chi_{\sigma_2}; e'_i \leftarrow \chi_{\sigma_1} \text{ for } 1 \le i \le N-1; & \\ X_i = (z_{i+1} - z_{i-1})s_i + e'_i \text{ for } i \in [N]; & \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; & \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e''_{N-1}; & \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; & \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; & \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); & : (\mathsf{T}, \mathsf{sk}) \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); & \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) & \end{array} \right\}$$

$$\text{Since } \Pr[\mathcal{A} \text{ wins}] = \frac{1}{2} + \mathsf{Adv}^{\mathsf{KE}}_{\mathsf{STUG}}(t, q_E) = \Pr_0[\mathsf{Query}] + \Pr_0[\overline{\mathsf{Query}}] \cdot \frac{1}{2},$$

$$\mathsf{Adv}^{\mathsf{KE}}_{\mathsf{STUG}}(t, q_E) \le \Pr_0[\mathsf{Query}].$$

Experiment 1. We replace X_0 into $X_0' = -\sum_{i=1}^{N-1} X_i + e_0'$. The rest are same as the previous experiment.

$$\mathsf{Exp}_1 := \left\{ \begin{array}{l} a \leftarrow R_q; \\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i; \text{ for } i \in [N]; \\ e_0' \leftarrow \chi_{\sigma_2}; e_i' \leftarrow \chi_{\sigma_1} \text{ for } 1 \le i \le N-1; \\ X_0' = -\sum_{i=1}^{N-1} X_i + e_0'; \\ X_i = (z_{i+1} - z_{i-1})s_i + e_i' \text{ for } 1 \le i \le N-1; \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e''_{N-1}; \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; \qquad\qquad : (\mathsf{T}, \mathsf{sk}) \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) \end{array} \right\}$$

Lemma 4. *Given two distributions of X_0 and X_0', if we have $2N\sqrt{n}\lambda^{3/2}\sigma_1^2 + (N-1)\sigma_1 \le \beta_{\mathsf{Rényi}}$, then*

$$Pr_0\left[\mathit{Query}\right] \le 2^{-\lambda+1} + \sqrt{Pr_1\left[\mathit{Query}\right] \cdot \frac{\exp\left(2\pi n(\beta_{\mathsf{Rényi}}/\sigma_2)^2\right)}{1-2^{-\lambda+1}}}$$

using the property of Rényi divergence.

Proof. Note that we may define the random variables X_0, X_0' in both experiments Exp_1 and Dist_1. We define Error and main as

$$\mathsf{Error} = \sum_{i=0}^{N-1}(s_ie_{i+1} - s_ie_{i-1}) + \sum_{i=1}^{N-1} e_i' \text{ and}$$
$$\mathsf{main} = z_1s_0 - z_{N-1}s_0 - \mathsf{Error},$$

respectively. Then,

$$X_0 = \mathsf{main} + \mathsf{Error} + e_0' \text{ and } X_0' = \mathsf{main} + e_0'$$

where $e_0' \leftarrow \sigma_2$. We check whether Rényi divergence between two distributions of X_0 and X_0' is small using Lemma 1. Let $\mathsf{bound}_{\mathsf{Error}}$ be the event that for all participants j, $\mathsf{Error}_j \le \beta_{\mathsf{Rényi}}$. Then,

$$|\mathsf{Error}_j| = \left| \left(\sum_{i=0}^{N-1}(s_ie_{i+1} - s_ie_{i-1}) + \sum_{i=1}^{N-1} e_i' \right)_j \right|.$$

Set $c = \sqrt{\frac{2\lambda}{\pi \log e}}$ and let bound be the event that $|(e'_0)_j| \leq c\sigma_2$, $|(s_i)_j|, |(e_i)_j|$, $\left|(e''_{N-1})_j\right| \leq c\sigma_1$, and $|(e'_i)_j| \leq c\sigma_1$ for all $i > 0$ and j.

From Lemmas 2 and 3, we have $\Pr[\mathsf{bound}] \geq 1 - 2^{-\lambda}$ and $\Pr[|(s_i e_j)_v| \leq \sqrt{n}\lambda^{3/2}\sigma_1^2 \mid \mathsf{bound}] \geq 1 - 2^{-2\lambda+1}$. With a union bound, we have

$$\begin{aligned}\Pr[\forall j : |\mathsf{Error}_j| \leq 2N\sqrt{n}\lambda^{3/2}\sigma_1^2 + (N-1)\sigma_1 \mid \mathsf{bound}]\\ \geq 1 - 4N \cdot n \cdot 2^{-2\lambda}.\end{aligned}$$

If we assume $4Nn \leq 2^{\lambda}$, we derive that $\Pr[\mathsf{bound}_{\mathsf{Error}}] \geq 1 - 2^{-\lambda+1}$.

We have $\mathrm{RD}_2(\mathsf{Error} + \chi_{\sigma_2} \| \chi_{\sigma_2}) \leq \exp(2\pi n(\beta_{\mathsf{Rényi}}/\sigma)^2)$ from Lemma 1. Thus,

$$\begin{aligned}\Pr_0[\mathsf{Query}] &\leq \Pr_0[\mathsf{Query} \mid \mathsf{bound}_{\mathsf{Error}}] + \Pr_0\left[\overline{\mathsf{bound}_{\mathsf{Error}}}\right]\\ &\leq \Pr_0[\mathsf{Query} \mid \mathsf{bound}_{\mathsf{Error}}] + 2^{-\lambda+1}\\ &\leq \sqrt{\Pr_1[\mathsf{Query} \mid \mathsf{bound}_{\mathsf{Error}}] \cdot \exp\left(2\pi n(\beta_{\mathsf{Rényi}}/\sigma_2)^2\right)}\\ &\quad + 2^{-\lambda+1}\\ &\leq \sqrt{\Pr_1[\mathsf{Query}] \cdot \frac{\exp\left(2\pi n(\beta_{\mathsf{Rényi}}/\sigma_2)^2\right)}{\Pr_1[\mathsf{bound}_{\mathsf{Error}}]}} + 2^{-\lambda+1}\\ &\leq \sqrt{\Pr_1[\mathsf{Query}] \cdot \frac{\exp\left(2\pi n(\beta_{\mathsf{Rényi}}/\sigma_2)^2\right)}{1 - 2^{-\lambda+1}}} + 2^{-\lambda+1}\end{aligned}$$

From second to third inequality, we use the property that Rényi divergence is bounded.

For the rest of the proof, we will show that

$$\Pr_1[\mathsf{Query}] \leq N \cdot \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},3}(t_1) + \mathsf{Adv}_{\mathsf{KeyRec}}(t_2) + \frac{q_E}{2^{\lambda}}.$$

Experiment 2. We replace z_0 into uniform element in R_q. The rest are same as the previous experiment.

$$\mathsf{Exp}_2 := \left\{\begin{array}{l} a, z_0 \leftarrow R_q;\\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i \text{ for } 1 \leq i \leq N-1;\\ e'_0 \leftarrow \chi_{\sigma_2}; e'_i \leftarrow \chi_{\sigma_1} \text{ for } 1 \leq i \leq N-1;\\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0;\\ X_i = (z_{i+1} - z_{i-1})s_i + e'_i \text{ for } 1 \leq i \leq N-1;\\ e''_{N-1} \leftarrow \chi_{\sigma_1};\\ Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e''_{N-1};\\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)};\\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j};\\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1});\\ \mathsf{sk} = \mathcal{H}(k_{N-1});\\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) \end{array} : (\mathsf{T}, \mathsf{sk})\right\}$$

Between Experiment 1 and Experiment 2, we replace one RLWE instance into random. Hence, $|\Pr_2[\mathsf{Query}] - \Pr_1[\mathsf{Query}]| \le \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},1}(t_1)$

where $t_1 = t + \mathcal{O}(N \cdot t_{ring})$ and t_{ring} is the time required to perform operations in R_q. Since $\mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},1}(t_1) \le \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},2}(t_1) \le \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},3}(t_1)$, we have $|\Pr_2[\mathsf{Query}] - \Pr_1[\mathsf{Query}]| \le \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},3}(t_1)$.

Experiment 3. We replace z_0 into $z_2 - r_1$ and X_1 into $r_1 s_1 + e'_1$ where $r_1 \leftarrow R_q$. The rest are same as the previous experiment.

$$\mathsf{Exp}_3 := \left\{ \begin{array}{ll} a, r_1 \leftarrow R_q; & \\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i \text{ for } 1 \le i \le N-1; & \\ z_0 = z_2 - r_1; & \\ e'_0 \leftarrow \chi_{\sigma_2}; e'_i \leftarrow \chi_{\sigma_1} \text{ for } 1 \le i \le N-1; & \\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0; & \\ X_1 = r_1 s_1 + e'_1; & \\ X_i = (z_{i+1} - z_{i-1}) s_i + e'_i \text{ for } 2 \le i \le N-1; & \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; & \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2} s_{N-1} + e''_{N-1}; & : (\mathsf{T}, \mathsf{sk}) \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; & \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; & \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); & \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); & \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) & \end{array} \right\}$$

Since both z_0 and $z_2 - r_1$ are uniform, $\Pr_3[\mathsf{Query}] = \Pr_2[\mathsf{Query}]$.

Experiment 4. We replace z_1, X_1 into uniform element in R_q. The rest are same as the previous experiment.

$$\mathsf{Exp}_4 := \left\{ \begin{array}{ll} a, r_1, z_1 \leftarrow R_q; & \\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i \text{ for } 2 \le i \le N-1; & \\ z_0 = z_2 - r_1; & \\ e'_0 \leftarrow \chi_{\sigma_2}; e'_i \leftarrow \chi_{\sigma_1} \text{ for } 2 \le i \le N-1; & \\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0;\ X_1 \leftarrow R_q; & \\ X_i = (z_{i+1} - z_{i-1}) s_i + e'_i \text{ for } 2 \le i \le N-1; & \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; & \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2} s_{N-1} + e''_{N-1}; & \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; & : (\mathsf{T}, \mathsf{sk}) \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; & \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); & \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); & \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) & \end{array} \right\}$$

Between Experiment 3 and Experiment 4, we replace two RLWE instances into random. Hence, $|\Pr_4[\mathsf{Query}] - \Pr_3[\mathsf{Query}]| \leq \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},2}(t_1)$ and thus,

$$|\Pr_4[\mathsf{Query}] - \Pr_3[\mathsf{Query}]| \leq \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},3}(t_1)$$

t_1 is the time to solve RLWE problem which is the sum of t and some minor overhead $\mathcal{O}(t_{ring})$ for simulation.

Experiment 5. We replace z_0 into uniform element in R_q. The rest are same as the previous experiment.

$$\mathsf{Exp}_5 := \left\{ \begin{array}{l} a, z_0, z_1 \leftarrow R_q; \\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i \text{ for } 2 \leq i \leq N-1; \\ e'_0 \leftarrow \chi_{\sigma_2}; e'_i \leftarrow \chi_{\sigma_1} \text{ for } 2 \leq i \leq N-1; \\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0;\ X_1 \leftarrow R_q; \\ X_i = (z_{i+1} - z_{i-1})s_i + e'_i \text{ for } 2 \leq i \leq N-1; \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e''_{N-1}; \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) \end{array} : (\mathsf{T}, \mathsf{sk}) \right\}$$

Since both z_0 and $z_2 - r_1$ are uniform, $\Pr_5[\mathsf{Query}] = \Pr_4[\mathsf{Query}]$. Similarly, we can design distribution of $(\mathsf{T}, \mathsf{sk})$ in Experiment $3j, 3j+1, 3j+2$ as below:

Experiment $3j$. We replace z_{j-1} into $z_{j+1} - r_i$ and X_i into $r_j s_j + e'_i$ where $r_j \leftarrow R_q$. The rest are same as the previous experiment.

$$\mathsf{Exp}_{3j} := \left\{ \begin{array}{l} a, r_j \leftarrow R_q; \\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i \text{ for } j \leq i \leq N-1; \\ z_0, \cdots, z_{j-2} \leftarrow R_q; z_{j-1} = z_{j+1} - r_j; \\ e'_0 \leftarrow \chi_{\sigma_2}; e'_i \leftarrow \chi_{\sigma_1} \text{ for } j+1 \leq i \leq N-1; \\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0; \\ X_1, \cdots, X_{j-1} \leftarrow R_q; X_j = r_j s_j + e'_j; \\ X_i = (z_{i+1} - z_{i-1})s_i + e'_i \text{ for } j+1 \leq i \leq N-1; \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e''_{N-1}; \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) \end{array} : (\mathsf{T}, \mathsf{sk}) \right\}$$

Experiment $3j+1$**.** We replace z_j, X_j into uniform element in R_q. The rest are same as the previous experiment.

$$\mathsf{Exp}_{3j+1} := \left\{ \begin{array}{l} a, r_j \leftarrow R_q; \\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i \text{ for } j+1 \leq i \leq N-1; \\ z_0, \cdots, z_{j-2}, z_j \leftarrow R_q; z_{j-1} = z_{j+1} - r_j; \\ e'_0 \leftarrow \chi_{\sigma_2}; e'_i \leftarrow \chi_{\sigma_1} \text{ for } j+1 \leq i \leq N-1; \\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0; \\ X_1, \cdots, X_j \leftarrow R_q; \\ X_i = (z_{i+1} - z_{i-1})s_i + e'_i \text{ for } j+1 \leq i \leq N-1; \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e''_{N-1}; \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) \end{array} : (\mathsf{T}, \mathsf{sk}) \right\}$$

Experiment $3j+2$**.** We replace z_{j-1} into uniform element in R_q. The rest are same as the previous experiment.

$$\mathsf{Exp}_{3j+2} := \left\{ \begin{array}{l} a \leftarrow R_q; \\ s_i, e_i \leftarrow \chi_{\sigma_1}; z_i = as_i + e_i \text{ for } j+1 \leq i \leq N-1; \\ z_0, \cdots, z_j \leftarrow R_q; \\ e'_0 \leftarrow \chi_{\sigma_2}; e'_i \leftarrow \chi_{\sigma_1} \text{ for } j+1 \leq i \leq N-1; \\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0; \\ X_1, \cdots, X_j \leftarrow R_q; X_j = r_j s_j + e'_j; \\ X_i = (z_{i+1} - z_{i-1})s_i + e'_i \text{ for } j+1 \leq i \leq N-1; \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2}s_{N-1} + e''_{N-1}; \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) \end{array} : (\mathsf{T}, \mathsf{sk}) \right\}$$

With similar argument of Experiment 3, 4 and 5, we have

$$\begin{aligned} &\Pr_{3i}[\mathsf{Query}] = \Pr_{3i-1}[\mathsf{Query}] \\ &|\Pr_{3i+1}[\mathsf{Query}] - \Pr_{3i}[\mathsf{Query}]| \leq \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},3}(t_1) \\ &\Pr_{3i+2}[\mathsf{Query}] = \Pr_{3i+1}[\mathsf{Query}] \end{aligned}$$

Experiment $3N-3$**.** We set $z_{N-2} = r_2, X_{N-1} = r_1 s_{N-1} + e'_{N-1}, z_0 = r_1 + r_2$ where $r_1, r_2 \leftarrow R_q$. The rest are same as the previous experiment.

$$\mathsf{Exp}_{3N-3} := \left\{ \begin{array}{ll} a, r_1, r_2 \leftarrow R_q; & \\ s_{N-1}, e_{N-1} \leftarrow \chi_{\sigma_1}; & \\ z_0 = r_1 + r_2; z_i \leftarrow R_q \text{ for } 1 \le i \le N-3; z_{N-2} = r_2; & \\ z_{N-1} = a s_{N-1} + e_{N-1}; & \\ e'_0 \leftarrow \chi_{\sigma_2}; e'_{N-1} \leftarrow \chi_{\sigma_1}; & : (\mathsf{T}, \mathsf{sk}) \\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0; & \\ X_i \leftarrow R_q \text{ for } 1 \le i \le N-2; & \\ X_{N-1} = r_1 s_{N-1} + e'_{N-1}; & \\ e''_{N-1} \leftarrow \chi_{\sigma_1}; & \\ Y_{N-1,N-1} = X_{N-1} + z_{N-2} s_{N-1} + e''_{N-1}; & \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; & \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; & \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); & \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); & \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) & \end{array} \right\}$$

Since r_1, r_2 are uniform, so does $z_0 = r_1 + r_2$. For both Experiment $3N-4$ and $3N-3$, z_{N-2} and z_0 are uniform. Then, we have $\Pr_{3N-3}[\mathsf{Query}] = \Pr_{3N-4}[\mathsf{Query}]$.

Experiment $3N-2$**.** We replace $z_{N-1}, X_{N-1}, z_{N-2} s_{N-1} + e''_{N-1}$ into uniform element in R_q. The rest are same as the previous experiment.

$$\mathsf{Exp}_{3N-2} := \left\{ \begin{array}{ll} a, r_3 \leftarrow R_q; & \\ z_i \leftarrow R_q \text{ for } i \in [N]; & \\ e'_0 \leftarrow \chi_{\sigma_2}; & \\ X'_0 = -\sum_{i=1}^{N-1} X_i + e'_0; & \\ X_i \leftarrow R_q \text{ for } 1 \le i \le N-1; & : (\mathsf{T}, \mathsf{sk}) \\ Y_{N-1,N-1} = X_{N-1} + r_3; & \\ Y_{N-1,(N-1)+j} = X_{(N-1)+j} + Y_{N-1,(N-1)+(j-1)}; & \\ b_{N-1} = \sum_{j=0}^{N-1} Y_{N-1,(N-1)+j}; & \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); & \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); & \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) & \end{array} \right\}$$

Between Experiment $3N-3$ and Experiment $3N-2$, we replace three RLWE instances into random. Hence,

$$|\Pr_{3N-2}[\mathsf{Query}] - \Pr_{3N-3}[\mathsf{Query}]| \le \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},3}(t_1)$$

Experiment $3N-1$**.** We replace $Y_{N-1,N-1}, Y_{N-1,(N-1)+j}$, b_{N-1} into uniform element in R_q. The rest are same as the previous experiment.

$$\mathsf{Exp}_{3N-1} := \left\{ \begin{array}{ll} a \leftarrow R_q; & \\ z_i \leftarrow R_q \text{ for } i \in [N]\,; & \\ e_0' \leftarrow \chi_{\sigma_2}; & \\ X_0' = -\sum_{i=1}^{N-1} X_i + e_0'; & \\ X_i \leftarrow R_q \text{ for } 1 \le i \le N-1; & : (\mathsf{T}, \mathsf{sk}) \\ Y_{N-1,(N-1)+j} \leftarrow R_q \text{ for } j \in [N]\,; & \\ b_{N-1} \leftarrow R_q; & \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); & \\ \mathsf{sk} = \mathcal{H}(k_{N-1}); & \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) & \end{array} \right\}$$

For both Experiment $3N-2$ and Experiment $3N-1$, $Y_{N-1,N-1}, Y_{N-1,(N-1)+j}$, and b_{N-1} are all uniform since r_3 is uniform in Experiment $3N-2$. Then, we have $\Pr_{3N-1}[\mathsf{Query}] = \Pr_{3N-2}[\mathsf{Query}]$.

Experiment $3N$**.** We replace k_{N-1} into uniform element k_{N-1}' in $\{0,1\}^\lambda$. The rest are same as the previous experiment.

$$\mathsf{Exp}_{3N} := \left\{ \begin{array}{ll} a \leftarrow R_q; & \\ z_i \leftarrow R_q \text{ for } i \in [N]\,; & \\ e_0' \leftarrow \chi_{\sigma_2}; & \\ X_0' = -\sum_{i=1}^{N-1} X_i + e_0'; & \\ X_i \leftarrow R_q \text{ for } 1 \le i \le N-1; & : (\mathsf{T}, \mathsf{sk}) \\ Y_{N-1,(N-1)+j} \leftarrow R_q \text{ for } j \in [N]\,; & \\ b_{N-1} \leftarrow R_q; & \\ (\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1}); & \\ k_{N-1}' \leftarrow \{0,1\}^\lambda; \mathsf{sk} = \mathcal{H}(k_{N-1}'); & \\ \mathsf{T} = (z_0, z_1, \cdots, z_{N-1}, X_0, X_1, \cdots, X_{N-1}, \mathsf{rec}) & \end{array} \right\}$$

Between Experiment $3N-1$ and Experiment $3N$, we replace k_{N-1} from $\mathsf{recMsg}(b_{N-1})$ into random. Hence,

$$|\Pr_{3N}[\mathsf{Query}] - \Pr_{3N-1}[\mathsf{Query}]| \le \mathsf{Adv}_{\mathsf{KeyRec}}(t_2)$$

t_2 is the time to break KeyRec algorithm which is the sum of t and some minor overhead $\mathcal{O}(t_{ring})$ for simulation.

Since adversary attacking STUG makes at most q_E queries to the random oracle, we have $\Pr_1[\mathsf{Query}] = \frac{q_E}{2^\lambda}$, which is negligible in λ.

From Experiment 1 to Experiment $3N$, we have

$$\Pr_1[\mathsf{Query}] \le N \cdot \mathsf{Adv}^{RLWE}_{n,q,\chi_{\sigma_1},3}(t_1) + \mathsf{Adv}_{\mathsf{KeyRec}}(t_2) + \frac{q_E}{2^\lambda}.$$

as expected. With the Lemma 4 and $\mathsf{Adv}^{\mathsf{KE}}_{\mathsf{STUG}}(t, q_E) \le \Pr_0[\mathsf{Query}]$, we derive the result of the theorem.

Theorem 3. *The authenticated GKE protocol STAG described in Section 5.2 is secure against active adversary under RLWE assumption, achieves forward secrecy and satisfies the following:*

$$\mathsf{Adv}^{\mathsf{AKE}}_{\mathsf{STAG}}(t, q_E, q_S) \leq \mathsf{Adv}^{\mathsf{KE}}_{\mathsf{STUG}}(t', q_E + \frac{q_S}{2}) + |\mathcal{P}|\mathsf{Adv}_{\mathsf{DSig}}(t')$$

where $t' \leq t + (|\mathcal{P}|q_E + q_S)t_{\mathsf{STAG}}$ when t_{STAG} is the time required for execution of STAG by any one of the protocol participants.

Proof. From an adversary $\mathcal{A}'$ which attacks STAG, we construct an adversary $\mathcal{A}$ who attacks STUG. We divide the event Succ that $\mathcal{A}'$ wins the security game defined in Section 4 into the one that $\mathcal{A}'$ can forge a signature and the one that $\mathcal{A}'$ cannot forge a signature.

For the former case, we claim that the probability of event Forge that the adversary can forge a signature is bounded by $|\mathcal{P}|\mathsf{Adv}_{\mathsf{DSig}}(t')$ where $|\mathcal{P}|$ is the number of participants. This is obvious since we have $|\mathcal{P}|$ protocol participants who generates their own signature. For the latter case, we claim that we can answer Execute and Send queries from STAG using Execute queries from STUG. Then, after $\mathcal{A}'$ ‘makes the query Reveal or Test, we derive the result of the theorem.

Theorem 4. *The dynamic authenticated GKE protocol DRAG described in Section 5.3 is secure against active adversary under RLWE assumption, achieves forward secrecy and satisfies the following:*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{AKE}}_{\mathsf{DRAG}}(t, q_E, q_J, q_L, q_S) \leq\ & \mathsf{Adv}^{\mathsf{KE}}_{\mathsf{STUG}}(t', q_E + \frac{q_J + q_L + q_S}{2}) \\ & + |\mathcal{P}|\mathsf{Adv}_{\mathsf{DSig}}(t')\end{aligned}$$

where $t' \leq t + (|\mathcal{P}|q_E + q_J + q_L + q_S)t_{\mathsf{DRAG}}$ when t_{DRAG} is the time required for execution of DRAG by any one of the protocol participants.

Proof. Similar to Theorem 4, we separate the winning event into two cases as the one with forging a signature and the other without forging. Then, we design how to answer Execute, Join, Leave and Send queries from DRAG using Execute queries from STUG.

7 Comparison with Other Protocols

In Table 1, we compare our construction with other lattice-based GKE protocols [2,17,34]. For computation complexity, we ignore ring addition/deletion, or scalar multiplication with smaller computing power. We consider the following:

Samp	total number of Gaussian samplings
R.Mult	total number of ring multiplication computed
Sign	total number of signatures generated
Verify	total number of verification

Table 1: Comparison with other lattice-based (authenticated) GKE protocols

Method	Ding *et al.* [17]	Yang *et al.* [34]	Apon *et al.*'s [2]	Ours
Trusted Authority[a]	**X**	O	**X**	**X**
Scalability[b]	X	**O**	**O**	**O**
Communication Round for GKE (AGKE)[c]	N	**2**	3(4)	**3(3)**
Computation Complexity[c] (Samp, R.Mult, Sign, Verify)	$(N^2, N^2-N, \cdot, \cdot)$	$(2N, 2N+2, \cdot, \cdot)$	$(3N+1, 2N+1, 2N, 2N)$	$(\boldsymbol{3N{+}1},\ \boldsymbol{2N{+}1},\ \boldsymbol{2N},\ \boldsymbol{N{+}2})$
Dynamic Setting[d]	X	X	X	**O**

[a] O: protocol needs trusted authority to run the procedure, X: protocol does not need trusted authority
[b] O: protocol is scalable, *i.e.*, protocol is constant-round regardless of the number of protocol participants, X: protocol is not scalable
[c] N is the number of protocol participants on GKE protocol.
[d] O: protocol supports dynamic membership changes like Join or Leave, X: protocol does not support them

From Table 1, Ding *et al.*'s protocol requires $N-1$ rounds to have N approximately agreed ring elements and one round to obtain session secret key by key reconciliation. For each party, it has N Gaussian samplings (one secret sampling and $N-1$ error samplings) and $N-1$ ring multiplications. Yang *et al.*'s protocol provides the minimum communication rounds but Yang *et al.*'s protocol has trusted authority so that it contains more security issues such as a single point of failure. Moreover, this protocol does one more computation for secure sketch, which requires huge computing power. Both Ding *et al.*'s and Yang *et al.*'s protocols do not specify digital signature scheme in the paper.

For Apon *et al.*'s protocol and our protocol, both provides scalability without trusted authority. Our protocol remains 3 round for authenticated GKE while Apon *et al.*'s protocol needs one more round from Katz-Yung compiler. The number of Gaussian sampling and ring multiplications are $3N+1$ and $2N+1$, respectively, for both protocols. But, we expect smaller number of signature verification step since we only check the signatures from the neighbourhood.

8 Conclusion and Future Work

In this paper, we construct a novel method to design a quantum-resistant dynamic (authenticated) GKE protocol by extending Dutta-Barua protocol to RLWE setting. Then, we compare our protocol with other lattice-based GKE protocols. Assuming the hardness of RLWE assumption and underlying digital signature scheme, we provide a concrete security analysis of our protocol against active adversary in the random oracle model.

As future work, we will check the vulnerability against key reuse attacks by applying the practical key reconciliation algorithm used in other lattice-based key exchange protocols. Then, we will implement the protocol based on our parameter selection. Then, we plan to check the security in the quantum-accessible random oracle model.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00555, Towards Provable-secure Multi-party Authenticated Key Exchange Protocol based on Lattices in a Quantum World)

References

1. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange-a new hope. In: USENIX Security Symposium. pp. 327–343 (2016)
2. Apon, D., Dachman-Soled, D., Gong, H., Katz, J.: Constant-round group key exchange from the ring-LWE assumption. In: International Conference on Post-Quantum Cryptography. pp. 189–205. Springer (2019)
3. Baan, H., Bhattacharya, S., Fluhrer, S.R., Garcia-Morchon, O., Laarhoven, T., Rietman, R., Saarinen, M.J.O., Tolhuizen, L., Zhang, Z.: Round5: Compact and fast post-quantum public-key encryption. pp. 83–102 (2019)
4. Bogdanov, A., Guo, S., Masny, D., Richelson, S., Rosen, A.: On the hardness of learning with rounding over small modulus. In: Theory of Cryptography Conference. pp. 209–224. Springer (2016)
5. Boneh, D., Glass, D., Krashen, D., Lauter, K., Sharif, S., Silverberg, A., Tibouchi, M., Zhandry, M.: Multiparty non-interactive key exchange and more from isogenies on elliptic curves. arXiv preprint arXiv:1807.03038 (2018)
6. Bos, J., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In: ACM SIGSAC Conference on Computer and Communications Security. pp. 1006–1018. ACM (2016)
7. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS–Kyber: a CCA-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367. IEEE (2018)
8. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: IEEE Symposium on Security and Privacy. pp. 553–570. IEEE (2015)
9. Bresson, E., Chevassut, O., Pointcheval, D.: Provably authenticated group Diffie-Hellman key exchange – the dynamic case. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 290–309. Springer (2001)
10. Bresson, E., Chevassut, O., Pointcheval, D.: Dynamic group Diffie-Hellman key exchange under standard assumptions. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 321–336. Springer (2002)
11. Bresson, E., Chevassut, O., Pointcheval, D., Quisquater, J.J.: Provably authenticated group Diffie-Hellman key exchange. In: Proceedings of the 8th ACM conference on Computer and Communications Security. pp. 255–264. ACM (2001)

12. Burmester, M., Desmedt, Y.: A secure and efficient conference key distribution system. In: Workshop on the Theory and Application of of Cryptographic Techniques. pp. 275–286. Springer (1994)
13. Burmester, M., Desmedt, Y.: A secure and scalable group key exchange system. Information Processing Letters **94**(3), 137–143 (2005)
14. Cheon, J.H., Kim, D., Lee, J., Song, Y.: Lizard: Cut off the tail! a practical post-quantum public-key encryption from LWE and LWR. In: International Conference on Security and Cryptography for Networks. pp. 160–177. Springer (2018)
15. Diffie, W., Hellman, M.: New directions in cryptography. IEEE transactions on Information Theory **22**(6), 644–654 (1976)
16. Ding, J., Alsayigh, S., Lancrenon, J., Saraswathy, R., Snook, M.: Provably secure password authenticated key exchange based on RLWE for the post-quantum world. In: Cryptographers' Track at the RSA Conference. pp. 183–204. Springer (2017)
17. Ding, J., Xie, X., Lin, X.: A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptology ePrint Archive 2012/688 (2012)
18. Dutta, R., Barua, R.: Constant round dynamic group key agreement. In: International Conference on Information Security. pp. 74–88. Springer (2005)
19. Gao, X., Ding, J., Liu, J., Li, L.: Post-quantum secure remote password protocol from rlwe problem. In: International Conference on Information Security and Cryptology. pp. 99–116. Springer (2017)
20. Hoeffding, W.: Probability inequalities for sums of bounded random variables. In: The Collected Works of Wassily Hoeffding, pp. 409–426. Springer (1994)
21. Katz, J., Shin, J.S.: Modeling insider attacks on group key-exchange protocols. In: Proceedings of the 12th ACM conference on Computer and communications security. pp. 180–189. ACM (2005)
22. Katz, J., Yung, M.: Scalable protocols for authenticated group key exchange. In: Annual International Cryptology Conference. pp. 110–125. Springer (2003)
23. Kim, Y., Perrig, A., Tsudik, G.: Simple and fault-tolerant key agreement for dynamic collaborative groups. In: Proceedings of the 7th ACM conference on Computer and communications security. pp. 235–244. ACM (2000)
24. Kim, Y., Perrig, A., Tsudik, G.: Tree-based group key agreement. ACM Transactions on Information and System Security (TISSEC) **7**(1), 60–96 (2004)
25. Langlois, A., Stehlé, D., Steinfeld, R.: GGHLite: More efficient multilinear maps from ideal lattices. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 239–256. Springer (2014)
26. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 1–23. Springer (2010)
27. Peikert, C.: Lattice cryptography for the internet. In: International Workshop on Post-Quantum Cryptography. pp. 197–219. Springer (2014)
28. Saarinen, M.J.O.: HILA5: On reliability, reconciliation, and error correction for Ring-LWE encryption. In: International Conference on Selected Areas in Cryptography. pp. 192–212. Springer (2017)
29. Seo, M., Kim, S., Lee, D.H., Park, J.H.: Emblem:(r) lwe-based key encapsulation with a new multi-bit encoding method. International Journal of Information Security pp. 1–17 (2019)
30. Steiner, M., Tsudik, G., Waidner, M.: Diffie-Hellman key distribution extended to group communication. In: Proceedings of the 3rd ACM conference on Computer and communications security. pp. 31–37. ACM (1996)

31. Steiner, M., Tsudik, G., Waidner, M.: CLIQUES: A new approach to group key agreement. In: Distributed Computing Systems, 1998. Proceedings. 18th International Conference on. pp. 380–387. IEEE (1998)
32. Steiner, M., Tsudik, G., Waidner, M.: Key agreement in dynamic peer groups. IEEE Transactions on Parallel and Distributed Systems **11**(8), 769–780 (2000)
33. Van Erven, T., Harremos, P.: Rényi divergence and Kullback-Leibler divergence. IEEE Transactions on Information Theory **60**(7), 3797–3820 (2014)
34. Yang, X., Ma, W., Zhang, C.: Group authenticated key exchange schemes via learning with errors. Security and Communication Networks **8**(17), 3142–3156 (2015)
35. Zhang, J., Zhang, Z., Ding, J., Snook, M., Dagdelen, Ö.: Authenticated key exchange from ideal lattices. In: Advances in Cryptology–EUROCRYPT 2015. pp. 719–751 (2015)