A Novel Non-Interactive Multi-party Key Exchange from Homomorphic Encryption

Rakyong Choi and Kwangjo Kim

School of Computing, KAIST
{thepride,kkj}@kaist.ac.kr

Abstract. In this paper, we consider the problem of key exchange among n parties. There are several multi-party key exchange schemes like group key exchange protocols. But, most of them are interactive key exchange protocols with more overhead.
Thus, we give a new generic approach to construct a non-interactive multi-party key exchange protocol without trusted third party. For that, we use the concept of homomorphic encryption scheme and generate a Boolean circuit to generate the ephemeral common key for n parties. We can achieve quantum-resistance with the lattice-based homomorphic encryption scheme from the literature.

Keywords: Non-interactive Key Exchange · Homomorphic Encryption · Group Key Exchange

1 Introduction

Since the seminal work by Diffie and Hellman [11], the need for a key exchange protocol over an insecure channel becomes essential to prevent unauthorized access or accidental disclosure of the information while transmission process between entities over an insecure network. Communicating between two entities on a public network needs to be secure to prevent any attacks to read transmitted messages. Secure transmission means encrypting the message with an encryption key and then sending it from one entity to another. The problem is how to share the key between two entities securely. For that, we use key exchange protocols which identifies each entity to another, create and distribute the key among them securely.

Homomorphic encryption supports any computation on encrypted data without decryption key. After Gentry's paper [15] in 2009, there are a number of research on homomorphic encryption based on ideal lattices, (ring-)LWE problem, and Approximate GCD problem [6, 7, 16, 24]. Homomorphic encryption is applicable to various areas using outsourcing computation like machine learning methods for encrypted data [9, 17, 18] or two-party key exchange protocol [21].

In this paper, we suggest a generic approach to construct a non-interactive multi-party key exchange protocol from rich cryptographic ingredients like homomorphic encryption scheme.

1.1 Outline of the Paper

The rest of this paper is structured as follows. We review the history of group key exchange protocol and homomorphic encryption scheme briefly in Chapter 2. We give the definition of non-interactive key exchange, homomorphic encryption, and homomorphic encryption scheme in Chapter 3. Then, we propose a new methodology to construct a non-interactive multi-party key exchange protocols without trusted third party in Chapter 4 and compare it with previous protocols in Chapter 5. Finally, we give a conclusion with future work in Chapter 6.

2 Previous Work

2.1 Group Key Exchange

A group key exchange (GKE) protocol is a multi-party key exchange protocol in which a shared secret is derived from n parties as a function of the information contributed by each of these. In GKE protocol, every group member has to interact in order to compute the group key and no entity can predetermine the resulting value. GKE protocol does not require the existence of secure channels between its participants since no secure transfer takes place during the processing.

Tree-based GKE is one method to obtain a common session key by some tree structure. For example, in Kim *et al.*'s paper [20], all user is considered as a leaf node of the tree and thus, no parties have higher authority.

In the paradigm of provable security, Bresson *et al.* [8] suggested the first security model for GKE protocols with two major security notions. The first notion is authenticated key exchange (AKE) security which requires the indistinguishability of computed group keys from random keys and the second notion is mutual authentication (MA) security which means that two parties authenticate mutually.

For quantum-resistant multi-party key exchange protocols, Ding *et al.* [12] constructed the lattice-based interactive multi-party GKE protocol and recently, Boneh *et al.* [4] proposed the non-interactive key exchange protocols from isogenies on elliptic curves.

2.2 Homomorphic Encryption

Since Rivest *et al.* [23] questioned whether there exist any encryption schemes that are homomorphic under any group/ring/field operations, which allows to perform arbitrary computation on the input data, it had been remained as an interesting open problem in cryptography for decades.

After Gentry's breakthrough paper [15] in 2009, many attempts are dedicated to make more efficient homomorphic encryption schemes based on LWE, Ring-LWE, and approximate GCD problems [5–7, 14, 16, 24].

For key agreement protocol, Krendelev and Kuzmin [21] recently proposed two-party key exchange protocol based on homomorphic encryption but their protocol misses the security proof and it considers two parties only.

3 Preliminaries

In this chapter, we review the definition of non-interactive key exchange protocol and homomorphic encryption scheme.

Definition 1. (non-interactive key exchange) A key exchange protocol is *non-interactive* when the protocol enables two parties who know each other's public key to agree on a shared common key without requiring any interaction and a multi-party key exchange protocol is *non-interactive* when there is no interaction between n parties.

Definition 2. (homomorphic encryption) A homomorphic encryption scheme $\mathcal{HE}$ is a tuple of PPT algorithms $\mathcal{HE} = (\mathsf{HE.Gen}, \mathsf{HE.Enc}, \mathsf{HE.Eval}, \mathsf{HE.Dec})$ with the following functionality:

$\mathsf{HE.Gen}(n, \alpha)$:
Given the security parameter n and an auxiliary input α, this algorithm outputs a key triple (pk, sk, evk), where pk is the key used for encryption, sk is the key used for decryption and evk is the key used for evaluation.

$\mathsf{HE.Enc}(pk, m)$:
Given a public key pk and a message m, this algorithm outputs a ciphertext c of the message m.

$\mathsf{HE.Eval}(evk, C, c_1, \cdots, c_n)$:
Given an evaluation key evk, a Boolean circuit C, and pairs $\{c_i\}_{i=1}^{n}$ where c_i is either a ciphertext or previous evaluation results, this algorithm produces an evaluation output.

$\mathsf{HE.Dec}(sk, c)$:
Given a secret key sk and a ciphertext or an evaluation output c, this algorithm outputs a message m.

4 Our Approach

In this chapter, we propose how to construct multi-party key exchange protocol with rich ingredients. Our construction can be considered as quantum-resistant protocol if we use lattice-based key exchange protocol and lattice-based homomorphic encryption scheme as underlying cryptographic protocols.

4.1 Security Model

In the following methods, we assume that the server is honest but curious so that the server should not be able to gain any information on the value of that session key during the protocol. Also, we assume that all parties are fully trusted so that no one can reveal the other's ephemeral key.

We assume that the adversary $\mathcal{A}$ can make queries to any instance as the former security modelling of key exchange protocols [1–3]. $\mathcal{A}$ can send messages

to some party, run the protocol to get the appropriate session key, and reveal the some session key but $\mathcal{A}$ cannot corrupt a party for any insider attacks.

To check the security of our multi-party key exchange protocol, we have to prove the following security requirements:

1. **Session key security**
   If uncorrupted parties in the proposed protocol complete matching sessions, they have the same key and the probability that the adversary guesses whether the key is from the protocol or from random is negligible. This can be interpreted as AKE security from Bresson *et al.*'s paper about the security model of GKE protocols [8].
2. **Known key security**
   Even after an adversary $\mathcal{A}$ has acquired one particular session key, other session keys are still secure.
3. **Key privacy**
   In the proposed protocol, the server should not be able to gain any information on the value of the session key, even though the server's help is mandatory to establish a session key between n parties in the protocol.
4. **Resistance to other various attacks**
   The protocol should withstand well-known network attacks such as user impersonation and modification attacks as well as man-in-the-middle attacks.

With these security requirements, we will check the validity of our multi-party key exchange protocol under the security of the given homomorphic encryption scheme.

Compared to Bresson *et al.*'s security requirement, we miss the security notion of mutual authentication since the underlying homomorphic encryption guarantees that the protocol outputs a valid output only if each party behaves honestly.

4.2 Construction with Homomorphic Encryption

In Fig. 1, we give a generic construction of non-interactive multi-party key exchange protocols from homomorphic encryption. As tree-based group key exchange protocol by Kim *et al.* [20], we restrict the parties to be located in the leaf node of the given graph (or a given circuit). Red rectangle box shows the area that remains hidden from outsiders and every party pre-shares the same key sk from $\mathsf{HE.Gen}$ algorithm of homomorphic encryption scheme $\mathcal{HE}$, like password-authenticated key exchange protocols. Note that a circuit C can be public in our protocol.

Under this condition, our protocol runs as follows.

Step 1. Make the Boolean circuit C with n inputs.

Step 2. Each party makes ephemeral session key k_i and encrypt it with public key pk, $c_i = \mathsf{HE.Enc}(pk, k_i)$.

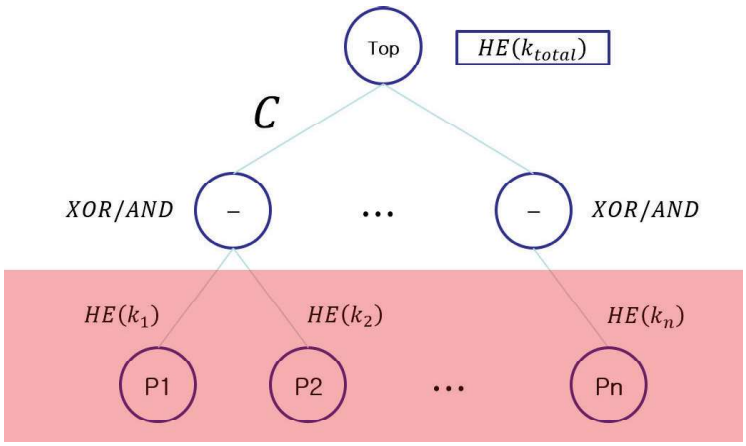

Fig. 1: Multi-party Key Exchange Protocol from Homomorphic Encryption

Step 3. Run evaluation algorithm $c = \mathsf{HE.Eval}(evk, C, c_1, \cdots, c_n)$ given the Boolean circuit C.

Step 4. Each party decrypts the evaluation value c and get the session group key $k = \mathsf{HE.Dec}(sk, c)$.

Theorem 1. *If underlying homomorphic encryption scheme $\mathcal{HE}$ is secure, our non-interactive multi-party key exchange protocol is also secure, i.e., it satisfies session key security, known key security, and key privacy.*

Proof. (sketch)

1. **Session key security**
   Since $\mathcal{HE}$ is secure, each ciphertext and evaluation value are distinguishable from random. Thus, all ciphertext c_i of the ephemeral session key k_i from party i are indistinguishable from random and so does the ciphertext c of the session group key k, evaluation value of all the ciphertext $c_i's$. Hence, our construction guarantees session key security.
2. **Known key security**
   Each session group key does not reveal the ephemeral session key k_i since the evaluation value k does not leak the information of the values in the circuit C. Even more, the party chooses different ephemeral session key k_i for each session. Hence, we cannot guess the previous session group key from one particular session group key. *i.e.*, we can guarantee known key security for our protocol.
3. **Key privacy**
   Since the server doesn't have the information of the pre-shared secret key sk, the server is not possible to know the session group key k but the evaluation value of it. As we stated in the session key security proof, evaluation value is

distinguishable from random when $\mathcal{HE}$ is secure and thus, our construction provides key privacy.

5 Comparison with Other Method

In Table 1, we compare our construction with other previous approaches like Kim *et al.*'s tree-based GKE protocol [20] and lattice-based multi-party key exchange protocols by Ding *et al.* [12].

Since Ding *et al.*'s protocol interacts one to each other, its complexity is $O(n^2)$, where n is the number of group members for group key agreement protocol. Meanwhile, our method can be achieved in $O(n)$ complexity with natural tree structure in the design.

Compared to other two methods, our method is non-interactive and does not need any fully-trusted third party in the protocol. We only need a server which is honest but curious. Also, our method can become a quantum-resistant group key agreement protocol if we adopt lattice-based homomorphic encryption scheme from the literature, like [16], for example.

Table 1: Comparison of group key agreement protocols

Method	Tree-based GKE [20]	DXL12 [12]	Ours
Communication Complexity[a]	$O(n)$	$O(n^2)$	$O(n)$
Non-interactivity[b]	X	X	**O**
Trusted Third Party[c]	O	**X**	**X**
Quantum Resistance[d]	X	**O**	△ [e]

[a] n is the number of group members for group key agreement protocol.
[b] O: protocol is non-interactive, O: protocol is interactive
[c] O: protocol needs the trusted third party, X: protocol doesn't need any trusted third party
[d] O: quantum-resistant, X: vulnerable to quantum computing attacks
[e] △: our design is quantum-resistant if the underlying homomorphic encryption scheme was designed to be quantum-resistant.

6 Concluding Remark

In this paper, we construct a novel method to design non-interactive multi-party key exchange protocol using homomorphic encryption scheme and compare this method with other protocols like tree-based group key exchange by Kim *et al.* [20] and lattice-based multi-party key exchange protocol by Ding *et al.* [12]

Our construction is a kind of group key exchange protocol and shares some properties that tree-based group key exchange and password-authenticated key exchange protocols.

As future work, first among several directions, we will adopt our generic construction to Gentry *et al.*'s well-known lattice-based homomorphic encryption scheme paper in CRYPTO 2013 [16]. Then, we will give the more concrete security proof including forward secrecy, where forward secrecy states that even if a party's long-term key is leaked to the adversary, the adversary is not able to acquire previous session keys, even though the adversary actively interfered, or tried to act as a man-in-the-middle attack. We will check security proof in both classical and quantum adversaries.

We also consider implementation of our quantum-resistant multi-party non-interactive key exchange protocol using some lattice-based libraries with homomorphic encryption tools like HElib and FHEW [13, 19].

Besides that, we leave the followings as challenging issues:

1) When each party has the different secret key. (We may use a multi-key variant of homomorphic encryption scheme [10, 22] instead of vanilla homomorphic encryption.)
2) When dynamic group settings are considered instead of static group setting so that the tree structure becomes updated.

Acknowledgement

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. Bellare, M., Pointcheval, D., and Rogaway, P. Authenticated key exchange secure against dictionary attacks. In *Advances in Cryptology–EUROCRYPT 2000* (2000), Springer, pp. 139–155.
2. Bellare, M., and Rogaway, P. Entity authentication and key distribution. In *Advances in Cryptology–CRYPTO'93* (1993), Springer, pp. 232–249.
3. Bellare, M., and Rogaway, P. Provably secure session key distribution: the three party case. In *Annual ACM symposium on Theory of computing* (1995), ACM, pp. 57–66.
4. Boneh, D., Glass, D., Krashen, D., Lauter, K., Sharif, S., Silverberg, A., Tibouchi, M., and Zhandry, M. Multiparty non-interactive key exchange and more from isogenies on elliptic curves. *arXiv preprint arXiv:1807.03038* (2018).
5. Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT) 6*, 3 (2014), 13.
6. Brakerski, Z., and Vaikuntanathan, V. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Advances in Cryptology–CRYPTO 2011*. Springer, 2011, pp. 505–524.
7. Brakerski, Z., and Vaikuntanathan, V. Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing 43*, 2 (2014), 831–871.

8. Bresson, E., Chevassut, O., Pointcheval, D., and Quisquater, J.-J. Provably authenticated group diffie-hellman key exchange. In *Proceedings of the 8th ACM conference on Computer and Communications Security* (2001), ACM, pp. 255–264.
9. Cheon, J. H., Kim, M., and Lauter, K. Homomorphic computation of edit distance. In *International Conference on Financial Cryptography and Data Security* (2015), Springer, pp. 194–212.
10. Clear, M., and McGoldrick, C. Multi-identity and multi-key leveled fhe from learning with errors. In *Annual Cryptology Conference* (2015), Springer, pp. 630–656.
11. Diffie, W., and Hellman, M. New directions in cryptography. *IEEE transactions on Information Theory 22*, 6 (1976), 644–654.
12. Ding, J., Xie, X., and Lin, X. A simple provably secure key exchange scheme based on the learning with errors problem. *IACR Cryptology ePrint Archive 2012/688* (2012).
13. Ducas, L., and Micciancio, D. Fhew: bootstrapping homomorphic encryption in less than a second. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2015), Springer, pp. 617–640.
14. Fan, J., and Vercauteren, F. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive 2012* (2012), 144.
15. Gentry, C. Fully homomorphic encryption using ideal lattices. In *Annual ACM on Symposium on Theory of Computing* (2009), ACM, pp. 169–178.
16. Gentry, C., Sahai, A., and Waters, B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 75–92.
17. Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning* (2016), pp. 201–210.
18. Graepel, T., Lauter, K., and Naehrig, M. Ml confidential: Machine learning on encrypted data. In *International Conference on Information Security and Cryptology* (2012), Springer, pp. 1–21.
19. Halevi, S., and Shoup, V. Algorithms in helib. In *International Cryptology Conference* (2014), Springer, pp. 554–571.
20. Kim, Y., Perrig, A., and Tsudik, G. Tree-based group key agreement. *ACM Transactions on Information and System Security (TISSEC) 7*, 1 (2004), 60–96.
21. Krendelev, S., and Kuzmin, I. Key exchange algorithm based on homomorphic encryption. In *Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on* (2017), IEEE, pp. 793–795.
22. López-Alt, A., Tromer, E., and Vaikuntanathan, V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing* (2012), ACM, pp. 1219–1234.
23. Rivest, R. L., Adleman, L., and Dertouzos, M. L. On data banks and privacy homomorphisms. In *Foundations of secure computation 4.11* (1978), pp. 169–180.
24. Van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V. Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2010), Springer, pp. 24–43.