Efficient and Evolvable Key Reconciliation Mechanism in Multi-party Networks Based on Automatic Learning Structure

Shuaishuai Zhu[1(✉)], Yiliang Han[1,2], Xiaoyuan Yang[1,2], and Xuguang Wu[2]

[1] College of Cryptography Engineering,
Engineering University of People's Armed Police, Xi'an 710086, China
zhu_sama@126.com

[2] Key Laboratory of Network and Information Security under the People's Armed Police, Xi'an 710086, China

Abstract. Key reconciliation protocols are critical components to deploy secure cryptographic primitives in practical applications. In this paper, we demonstrate on these new requirements and try to explore a new design routine in solving the key reconciliation problem in large scale p2p networks with automatic intelligent end user under the notion of evolvable cryptography. We design a new evolvable key reconciliation mechanism (KRM) based on two tricks for the AI user: the observation of shared beacons to evolve based on a deep auto-encoder, and the exchange of observed features as a hint to reconcile a shared key based on a deep paired decoder. For any passive adversary, the KRM is forward provable secure under the linear decoding hardness assumption. Compared with existing schemes, the performance evaluation showed our KRM is practical and quite efficient in communication and time costs, especially in multi-party scenarios.

Keywords: Evolvable cryptography · Key exchange protocol · Automatic learning · Peer-to-peer network · Security model

1 Introduction

While the Internet is entering into the era of artificial intelligence, the development pace of cryptography seems to be delayed. When we focus on designing post-quantum cryptographic primitives, new pattern of requirements and security threats in AI application scenarios boom. Countless of intellectual devices and AI terminals have access to the Internet to share data, features and models,

Supported by the National Natural Science Foundation of China (No. 61572521, U163 6114), National Key Project of Research and Development Plan (2017YFB0802000), Innovative Research Team Project of Engineering University of APF (KYTD201805), Fundamental Research Project of Engineering University of APF (WJY201910).

S. Yu et al. (Eds.): SPDE 2020, CCIS 1268, pp. 91–99, 2020.
https://doi.org/10.1007/978-981-15-9129-7_7

which require communication based on large scale of secure sessions. How to efficiently and delicately share a session key in a peer-to-peer AI network is totally a new topic. The main applying environment of current KEM schemes and key negotiation protocols are all heavy deployment based on the terminal browsers, more specially, embedded in the TLS handshake protocols [22]. Why don't deploy the traditional KEM based on DH or encryption key exchange? Technically, we sure can do that, but the KEM would become a performance bottleneck and the advantages of AI users, such as evolvable and cheap in computing power while costly in communication channels, is neglected in KEM design. In networks involving large scale of AI users like auto-pilots or smart sensor devices, for the sake of efficiency and global cost of key reconciliation, the current KEM primitives such as the post-quantum candidates of NIST [2] and current standard schemes of ISO/IEC cannot be directly deployed in p2p scenarios with large scale of AI users.

The deployment of traditional key exchange protocol in a vast scale p2p network is awkward and inefficient, because of the high cost of maintaining independent parameters for each key reconciliation, which brings communication inefficiency, inconvenience and security issue in the long run. In practical applications like multi-user p2p networks, communication cost is always much higher than computing cost, so that new KRM construction should occupy lower message exchange cost. Besides, current KEM and KRM solutions including the NIST's post-quantum candidates [2] only support fixed system parameters and configurations in real scenarios, in which the deployment in multi-user p2p networks is clumsy and awkward, and its security cost is expensive to reconcile a session key in a short slice of connection slot. To solve the above obvious drawbacks, here we resort to an evolvable design routine to passively or adoptively generate session keys in p2p networks. Compared with the existing computing reconciliation based KEMs, we apply a generative methodology based on which a share secret key is learned and generated from public observation during the p2p connection. In this section, we try to fundamentally improve these issues by the constructions of evolvable KRM based on the combination of automatically learning encoder and decoder (auto-encoder, noted as Ae for short).

2 Related Works

Key Exchange Mechanisms. We assume that key exchange mechanism is a special instantiation for key encapsulation mechanisms(KEM), which a key component to encapsulate a cryptographic primitive in the practical communication protocols. For a long time, the discrete logistic based Diffie-Hellman key exchange is the standard KEM realization [17]. But in the post-quantum KEM, lattice and LWE based reconciliation [8,9,19] or exchange [3,5–7] take the main role. The post-quantum KEM usually includes the authenticated protocols like [13,23], in which signatures or additional verifying structures are applied, and the direct KEM which is much brief and efficient, such as Ding [8,9], Peikert [19], and Alkim's NewHope [3,4] that built on Ring-LWE assumption. Also,

there are KEM based on standard LWE assumption, which makes the scheme more brief, such as Frodo protocol [5] and Kyber protocol [6]. Schemes in the first category can easily satisfy strong security like IND-CCA and IND-CCA2 in quantum security model, despite there complex steps and heavy bandwidth costs. KEMs in the second and the third categray may only achieve passively secure, unless safe hash functions or FO transformation [11] are applied, such as Alkim's NewHope.

Generative Secure Communication. The possibility of designing cryptography schemes with the automatic learning techniques such as machine learning and deep learning is discussed firstly in [20]. In the research of KEM, early works focused on how to build secure channels to establish session keys using the method of machine learning [14,18,21]. These automatic approaches cannot generate secure KEM protocols, and their secure keys cannot evolve during further communication. Then for a long period, the research process seems quite hard in handling learning details such as the discrete data training problem [15], the computation overload problem [10] and a less practical outcome. But in recent years, with the widely application of AI technologies and the development of supporting hardware, pure AI based secure communication is becoming an attempting pattern in the future. In 2014, the well-known learning model called generative adversarial network (GAN) [12] appeared, and then it is immediately applied to train a map between an arbitrary input and a target output. The optimized map is then naturally be treated as an encryption or an decryption algorithm. Compared with a mathematical concrete algorithm, the map from a GAN is automatically acquired through statistical adjustment during the training phase. In 2016, Google Brain team [1] published their first secure communication model with automatic negotiated encryption scheme whose security is guaranteed by a passive security model in which the adversary is a third party similar passive learner. As in their demonstration, the receiver can decrypt the message (a 16 bits message sampled in a normal distribution) with overwhelming probability, while the adversary cannot avoid approximately 50% of decoding error with overwhelming probability. But in the continues work of [16], the Google brain's model is found insecure under the attack of stronger adversaries.

3 Construction of Key Reconciliation Mechanism

3.1 System Framework

We first start from constructing KRM in the two party scenario in which our approach can easily be demonstrated. Each user in the network configures a auto-learning system Ae, whose initial state is shared by all users when entering the network. Ae can automatically observe and learn the connection of the recent beacon users which are also leveled users in the network, see Fig. 1. Features f is a transformed representation of the input G_i, and with a complete sample set input, an Ae statistically satisfies $acc(Ae) = P\{G_i = G_{i+1}\} \to 1$.

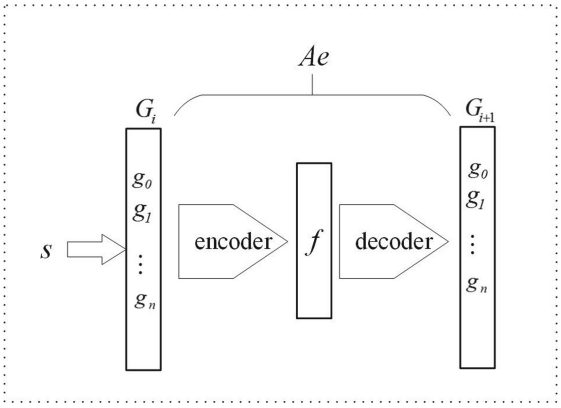

Fig. 1. Auto-encoder structure

3.2 Basic Model: Key Reconciliation Mechanism Within Two Parties

In a peer-to-peer network, there are three types of entities: requesters, responders and beacons. Their computing and storage resources may be vary, but their place in executing p2p protocol is leveled, and their roles may switch in different tasks. KRM within two parties takes the following four steps:

System Init. A set of global parameters are generated, including an secret initial state S_0 of a network auto-encoder $(Ae(n = 2^k, d = 2^{k-\alpha}, G)$ with accuracy threshold $\eta)$, a secure parameter k, a global identity set $S = \{\cdots, ID_A, \cdots, ID_B \cdots\}$, a collection of beacons $s \in S$, a collision resistent hash function $h(\cdot)$, and a global time tick $i \geq 0$. In each time tick, a beacon scans its connections and sets a vector $g = \{g_0, g_1, \cdots, g_n\}$, $g_i \in \{0, 1\}$, as a current broadcasting beacon sample in S. When $i < 1$, each user observes sample set $s \in S$ and trains Ae_{ID} until $\eta_{ID} \geq \eta$. When $i \geq 1$, the latest state of Ae_{ID} is kept as a secret to evolve new keys in each time tick.

State Evolve. The requestor Alice with ID_A first observes the current beacons s to sample the state of the current network, and obtain a sample set I as the input of Ae. For randomly picked $G_i \in I$, if the accuracy $a \geq \eta$ in decoding G_i, output a feature f and a decoding result G_{i+1} for G_i. Alice runs $\alpha \leftarrow Eval(G'_{i-1}, G'_i)$ to obtains a valid reconciliation threshold. Then he continues to compute $r \leftarrow Rec(G'_{i-1}, G'_i, \alpha)$. Finally, for any user with whom Alice wants to negotiate a key, he sends f to the receiver with ID_B, and computes $k_i \leftarrow h(f||r||ID_A||ID_B)$.

Key Gen. This is a probabilistic procedure. On received the feature f, Bob train his own Ae through the observation of s, and generate a G'_i applying f in Ae. Then Bob runs $\alpha \leftarrow Eval(G'_{i-1}, G'_i)$, and if α exists, he continues to compute $r \leftarrow Rec(G'_{i-1}, G'_i, \alpha)$. Finally, $k_i \leftarrow h(f||r||ID_A||ID_B)$. If α is invalid, then Bob reject f, and jumps to next time tick $i + 1$.

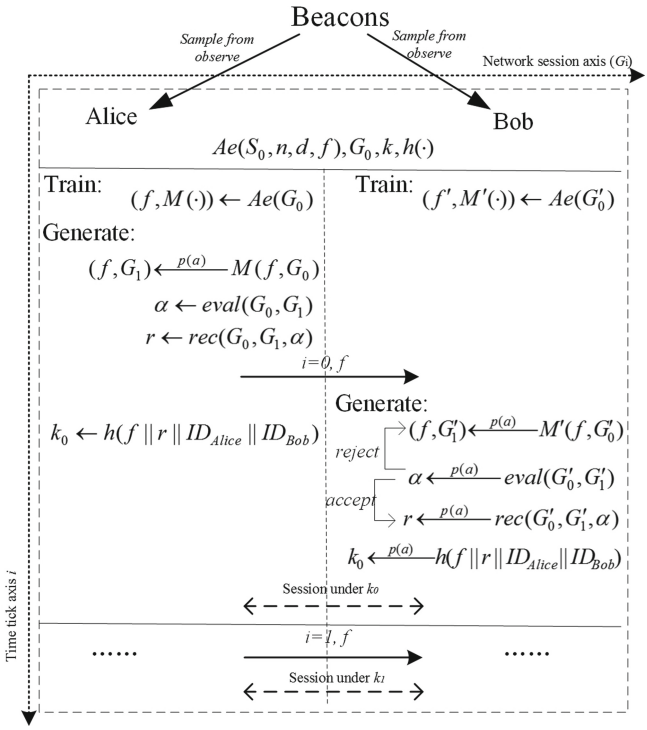

Fig. 2. Basic model with two parties

State Est. Once k_i successfully reconciliated, Alice and Bob establish a connection and update g_{Alice} and g_{Bob} as new beacons from time tick $i+1$ with probability p (Fig. 2).

3.3 Improved Model: Key Reconciliation Mechanism Within Two Parties

Although the basic model is brief enough with its one-pass message, but its security relies only on the decoder of the responder which may bring expected attack during its instantiation. Besides, a complete passive reconciliation can increase failure probability of the KEM procedure. So we less efficient but more secure variant.

KRM(ID_A, ID_B, S) within two nodes ID_A and ID_B follows five steps:

System Init. A set of global parameters are generated, including an initial state S_0 of two types of network auto-encoders (Ae_{ext} and Ae_{cpr} with accuracy threshold η and feature dimension n, $n > \alpha$ and $d = k/\alpha$ respectively), a secure parameter k, a global identity set $S = \{\cdots, ID_A, \cdots, ID_B \cdots\}$, a collection of beacons $s \in S$, and a collision resistent hash function $h(\cdot)$.

State Evolve. The requestor Alice with ID_A first observes the current beacons s to sample the state of the current network, and obtain a sample set I as the training input of Ae_{ext} and Ae_{cpr}. For randomly picked $G_i \in I$, if the accuracy $a \geq \eta$ in decoding G_i, Ae_{ext} and Ae_{cpr} output features f and f_c and a decoding result G_{i+1} for G_i. Alice runs $\alpha \leftarrow eval(G'_{i-1}, G'_i)$ to obtains a valid reconciliation threshold. Then he continues to compute $r \leftarrow Rec(G'_{i-1}, G'_i, \alpha)$. Finally, for any user with whom Alice wants to negotiate a key, he sends f to the receiver with ID_B.

Key Gen1. This is a probabilistic procedure. On received the feature f, Bob train his own Ae_{ext} through the observation of s, and generate a G'_i applying f in Ae_{ext}. Then Bob runs $\alpha \leftarrow eval(G'_{i-1}, G'_i)$, and if α exists, he continues to compute $r \leftarrow Rec(G'_{i-1}, G'_i, \alpha)$. If α is invalid, then Bob reject f, and jumps to next time tick $i+1$, else Bob sets $n = k/\alpha$ for Ae_{cpr}. By decoding G'_i in Ae_{cpr}, Bob obtains f_c as the compressed feature of G'_i. Finally, Bob computes $k_i \leftarrow h(f_c||r||ID_A||ID_B)$, and sends the fresh G'_i back to Alice.

Key Gen2. On received an G'_i, Alice generates f_c in Ae_{cpr}, and computes $k_i \leftarrow h(f_c||r||ID_A||ID_B)$.

State Est. Once k_i successfully reconciliated, Alice and Bob establish a connection and update g_{Alice} and g_{Bob} as new beacons from time tick $i+1$ (Fig. 3).

3.4 Key Reconciliation Mechanism Within Multi-parties

KRM within multi-parties scenario is basically a multi-replica of two parties with one essential problem to handle: extra update of keys for user's dynamic connectivity. With the evolvement of the p2p network, old connections might be disconnected, and new connections might be established according to an average transition probability p. After genesis of the network, it assumed to contain at least $|S|+2$ nodes including a unique beacon to allocate parameters for dynamic nodes. We extract four types of events: *system init*, *key evolve*, *join*, and *drop*. System init is a global event to initialize parameters and prepare local encoders & decoders by observation. The rest operations are used to update keys for connection transition. We apply the improved KRM model to demonstrate the four events of the multi-parties scenario.

System Init. For a p2p network involving at least $|S|+2$ different nodes where contains an unique initial beacon ID_0 staying online, all global parameters are generated, including an initial state S_0 of two types of network auto-encoders (Ae_{ext} and Ae_{cpr} with accuracy threshold η and feature dimension n, $n > \alpha$ and $d = k/\alpha$ respectively), a secure parameter k, a global identity set $S = \{\cdots, ID_A, \cdots, ID_B \cdots\}$, a collection of beacons $s \in S$, and a collision resistent hash function $h(\cdot)$. Global parameters are allocated by ID_0.

Key Evolve. For every time tick i, any two users $ID_A \in S$ and $ID_B \in S$ who make connection transition in the network make an observation of S and execute $k_i \leftarrow$KEM(ID_A, ID_B, S) with each other. On reconciliation success, k_i established, or else the process retry in the next time tick for the same nodes. Finally, after enough time ticks, independent keys are generated between any two users with overwhelming probability.

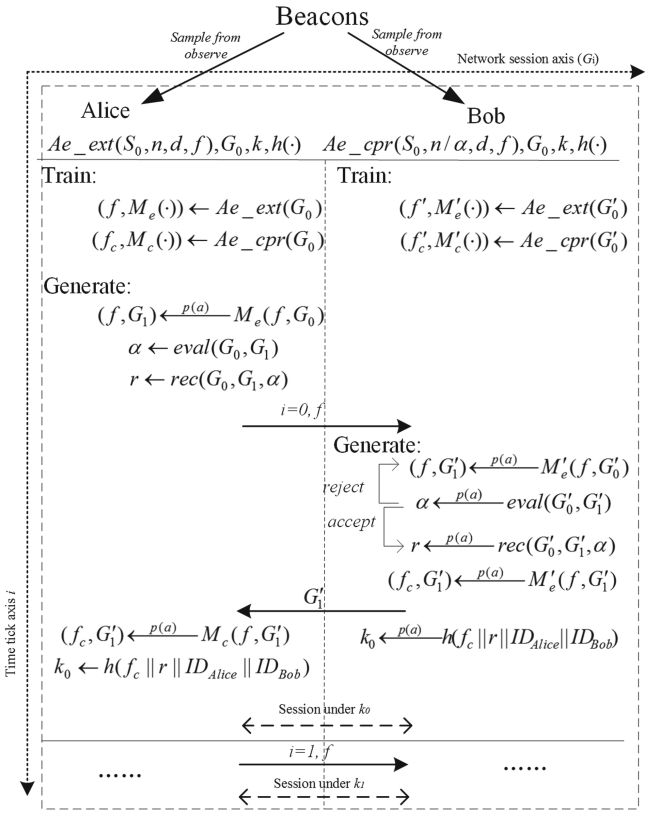

Fig. 3. Improved model with two parties

Join. This is a probabilistic procedure. For new user ID_C joining the network with Ae_{ext} and Ae_{cpr}, ID_0 allocates an S_0,G_0 and the current f for ID_C. Then taking G_0 as the expected state of S, ID_C randomly generates s_i with state transition probability of p. Ae_{ext} and Ae_{cpr} update their state by training on s_i. Taking f as the input of decoder of Ae_ext, ID_C generates G'_{i+1}. Taking G'_{i+1} as the input of encoder of Ae_cpr and generates f_c. Finally, ID_C reconciliates with G_0 and G'_{i+1}, computes $k_i \leftarrow h(f_c||r||ID_C||ID_0)$, and sends the fresh G'_i back to ID_0.

Drop. During each time tick i, for any user ID_A's connection state transits, its key stops evolving. The dependency of Ae_{ext} and Ae_{cpr} toward G_i drops after time tick i. Users in a p2p network is free to join and exit, and the transiting probability may vary according to network task, local resources and routine modification. Here we only considered an ideal case of constant state transition probability to comply with the previous correctness base. But if the training results is independent or weak dependent with partial state change of connection graph, the KRM within multi-parties can also be applicable.

4 Conclusions and Future Works

In the late Internet ecology, the AI technologies carry through nearly all the major applications. The trend of automatic design and analysis of cryptographic primitives for specific communication patterns in the era of AI is inevitable. Current works on the spot have already showed their vitality in designing secure communication protocols and analyzing traditional encryption algorithms. Following this interesting direction, we explored the possibility of designing one of the most important cryptographic mechanisms, the KRM in the specific P2P communication scenario.

In this paper, we designed a generative approach to automatically generate the KRM instances for P2P communication networks without the heavy load of frequently key exchange. Instead, the peers in the network only need to randomly observe the surrounding beacons to negotiate shared features. Then each peers generate their own session keys with these features. So far as we know, it is the first generative model to negotiate shared keys, and its advantages in efficiency and briefness naturally required in the P2P communication with vast amount of peers. But in our approach, there are still many unsolved problems, including the unstable success rate in generating shared keys, hardness in extending the width of a satisfying auto-encoder, and lack of standard evaluation in key evolvement. In our current experiment, the length of practical keys only reaches 64∼128 bits, which obviously cannot satisfy a long-term secure communication.

In our future work, two directions need to be explored. On security aspect, the state of referred beacons need to be improved to generate random and stable input samples for target peers. Then the architecture and parameters of the generative model should be optimized to obtain wide and stable outputs. On the efficiency aspect, a practical modification of the decoding component is required to polish the randomness of the key evolvement.

References

1. Abadi, M., Andersen, D.G.: Learning to protect communications with adversarial neural cryptography. arXiv preprint arXiv:1610.06918 (2016)
2. Alagic, G., et al.: Status report on the first round of the NIST post-quantum cryptography standardization process. US Department of Commerce, National Institute of Standards and Technology (2019)
3. Alkim, E., et al.: Newhope-algorithm specifications and supporting documentation. Second Round NIST PQC Project Submission Document (2019)
4. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange-a new hope. In: 25th {USENIX} Security Symposium ({USENIX} Security 2016), pp. 327–343 (2016)
5. Bos, J., Costello, C., Ducas, L., et al.: Frodo: take off the ring! practical, quantum-secure key exchange from LWE. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1006–1018 (2016)
6. Bos, J., et al.: 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)

7. 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: 2015 IEEE Symposium on Security and Privacy, pp. 553–570. IEEE (2015)
8. Ding, J.: New cryptographic constructions using generalized learning with errors problem. IACR Cryptology ePrint Archive, 2012:387 (2012)
9. Ding, J., Takagi, T., Gao, X., Wang, Y.: Ding key exchange. Technical report, National Institute of Standards and Technology (2017)
10. Dudzik, M., Drapik, S., Prusak, J.: Approximation of overloads for a selected tram traction substation using artificial neural networks. Technical Transactions (2016)
11. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_34
12. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
13. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10993, pp. 96–125. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96878-0_4
14. Klimov, A., Mityagin, A., Shamir, A.: Analysis of neural cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 288–298. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36178-2_18
15. Kusner, M.J., Hernández-Lobato, J.M.: GANS for sequences of discrete elements with the Gumbel-Softmax distribution. arXiv preprint arXiv:1611.04051 (2016)
16. Zhou, L., Chen, J., Zhang, Y., Su, C., James, M.A.: Security analysis and new models on the intelligent symmetric key encryption. Comput. Secur. **25**, 14–24 (2019)
17. Maurer, U.M., Wolf, S.: The Diffie-Hellman protocol. Des. Codes Crypt. **19**(2–3), 147–171 (2000)
18. Mislovaty, R., Klein, E., Kanter, I., Kinzel, W.: Security of neural cryptography. In: Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2004, pp. 219–221. IEEE (2004)
19. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 197–219. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_12
20. Rivest, R.L.: Cryptography and machine learning. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT 1991. LNCS, vol. 739, pp. 427–439. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57332-1_36
21. Ruttor, A.: Neural synchronization and cryptography. arXiv preprint arXiv:0711.2411 (2007)
22. Smith III, T.J., Rai, V.R., Collins, B.M.: Creating and utilizing black keys for the transport layer security (TLS) handshake protocol and method therefor. US Patent App. 15/738,567, 5 July 2018
23. Zhang, J., Zhang, Z., Ding, J., Snook, M., Dagdelen, Ö.: Authenticated key exchange from ideal lattices. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 719–751. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_24