Lattice-Based 0-RTT Key Exchange Protocol with Perfect Forward Secrecy (Postprint)

Authors: Zhao Zongqu, Ma Shaoti, Tang Yongli, Ye Qing

Date: 2020-09-28T00:00:00+00:00

Abstract

0-RTT (Zero Round-Trip Time) key exchange protocols enable clients to send encrypted-protected payloads and the first key exchange protocol message with zero round-trip time, featuring advantages such as non-interactivity and offline capability. To reduce the round-trip time of key exchange, this paper proposes a lattice-based 0-RTT key exchange protocol based on the puncturable encryption paradigm. First, a puncturable forward-secret key encapsulation scheme is constructed utilizing a one-time signature algorithm and a hierarchical identity-based key encapsulation mechanism, which is then employed to design the 0-RTT key exchange protocol. The protocol requires only unidirectional authentication from the client to the server and can effectively resist quantum attacks and replay attacks. Compared with similar protocols, the proposed protocol achieves puncturable full forward security, reduces the number of communication rounds, and improves communication efficiency.

Full Text

Preamble

Vol. 38 No. 3
Application Research of Computers
ChinaXiv Cooperative Journal

Zero Round-Trip Time Key Exchange Protocol with Full Forward Secrecy on Lattice

Zhao Zongqu, Ma Shaoti, Tang Yongli, Ye Qing†
(College of Computer Science & Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, China)

Abstract: 0-RTT key exchange protocols enable clients to send encrypted payloads and the first key exchange protocol message with zero round-trip time,

offering advantages such as non-interactivity and offline operation. To reduce the round-trip time of key exchange, this paper proposes a lattice-based 0-RTT key exchange protocol built upon the concept of puncturable encryption. First, we construct a puncturable forward-secret key encapsulation scheme using a one-time signature algorithm and a hierarchical identity-based key encapsulation mechanism, then design a 0-RTT key exchange protocol using this puncturable forward-secret key encapsulation scheme. The protocol requires only unidirectional authentication of the server by the client and effectively resists both quantum and replay attacks. Compared with similar protocols, the proposed protocol achieves puncturable full forward secrecy, reduces the number of communication rounds, and improves communication efficiency.

Keywords: lattice; key exchange; zero round-trip time (0-RTT); forward secrecy

0 Introduction

Authenticated key exchange (AKE) protocols are prerequisites for secure communication between entities, allowing communication parties to establish a shared high-entropy session key over a public channel, which is then used for message encryption, authentication, and integrity verification. In 1976, Diffie and Hellman [**?**] designed the first key exchange protocol based on the discrete logarithm problem, known as the "DH protocol." This groundbreaking work accelerated development in the field of key exchange protocols. Since the DH protocol is only passively secure and cannot resist active attacks, posing risks of session key leakage, many key exchange protocols based on the DH assumption \cite{2~4} have been proposed to defend against active attacks by exchanging partial key elements or incorporating identity information.

Classic AKE protocols such as TLS require exchanging substantial protocol information to negotiate a shared session key before transmitting the first effective data message, resulting in considerable latency overhead. Latency is typically measured in round-trip time (RTT), where N round-trip message exchanges are required before sending the first actual data, i.e., N-RTT. The Transport Layer Security (TLS) protocol provides authenticated key exchange, enabling two remote parties to establish a shared session key over an insecure channel. The TLS 1.2 [**?**] protocol requires two round-trip times (2-RTT) to complete the handshake before sending requests; TLS 1.3 [**?**] is faster and more secure than TLS 1.2, requiring only one round-trip time (1-RTT) for the TLS handshake, and zero round-trip time if the client has previously connected to the server.

High-performance AKE protocols such as HMQV [**?**] also require sending at least two messages (i.e., 1-RTT) when negotiating session keys. Reducing the latency overhead of key exchange protocols to zero round-trip time (0-RTT) while maintaining strict security guarantees has become a major design goal in both academia and industry. From a practical perspective, Google' s QUIC

protocol [?] not only reduces latency overhead to zero round-trip time but has also been implemented in Google Chrome and Opera web browsers, and was proposed to the IETF as an IETF standard by Google in 2015.

In 2017, Günther, Hale, Jager, and Lauer [?] [?] proposed a 0-RTT key exchange protocol with full forward secrecy based on puncturable encryption, constructed upon the forward-secure public-key encryption work of Canetti, Halevi, and Katz [?] and the forward-secret puncturable public-key encryption work of Green et al. [?]. The GHJL17 [?] scheme achieves a forward-secret one-pass key exchange protocol by constructing a puncturable forward-secret key encapsulation mechanism using one-time signature techniques and the hierarchical identity-based key encapsulation mechanism of Blazy et al. [?], making forward-secret 0-RTT key exchange protocols possible with full forward secrecy and resistance to replay attacks. In 2018, Derler and Jager et al. [?] constructed a 0-RTT key exchange protocol using Bloom Filter Encryption (BFE), improving computational efficiency and limiting key growth to a tolerable extent by tolerating a non-negligible correctness error.

With the development of quantum computing theory, AKE protocols [?] based on traditional integer factorization and discrete logarithm problems cannot resist quantum attacks. In the post-quantum era, these difficult problems can be solved efficiently by quantum algorithms in polynomial time. However, public-key cryptographic schemes based on lattice problems have no known polynomial-time efficient solving algorithms under quantum theory, and lattice operations involve matrix-vector multiplication, offering parallel computation and high efficiency.

Lattice-based cryptography has received widespread attention. In 2009, Katz et al. [?] designed a lattice-based encryption scheme secure against chosen-ciphertext attacks (CCA) and constructed an approximate smooth projection hash (ASPH) function, proposing the first lattice-based two-party password-authenticated key exchange (2PAKE) protocol. In 2011, Ding et al. [?] proposed an efficient lattice-based PAKE protocol by combining the encryption scheme and ASPH function from Katz et al. [?] within the Groce-Katz framework [?], and proved its security in the standard model. In 2017, Zhang et al. [?] combined a split public-key encryption scheme with the three-message framework of Katz et al. [?] to propose a lattice-based PAKE protocol requiring only two communication rounds, improving communication efficiency, though the scheme suffers from increased computational overhead due to split public-key encryption. In 2019, Li Zichen et al. [?] designed a post-quantum authenticated key exchange protocol based on the ring learning with errors problem, provably secure in the standard eCK model and achieving weak full forward secrecy. In these lattice-based AKE protocols, negotiating session keys requires 2 or 3 communication rounds, incurring greater communication overhead.

This paper constructs a novel 0-RTT key exchange protocol based on the design philosophy of the GHJL17 scheme [?]. The main contributions are: (1) We design a puncturable forward-secret key encapsulation mechanism that im-

plements puncturable encryption and key update functionality, endowing our protocol with full forward secrecy; (2) We construct our protocol based on this puncturable forward-secret key encapsulation mechanism, reducing communication rounds and lowering communication overhead. The proposed protocol features full forward secrecy, resistance to quantum and replay attacks, and achieves zero-round communication, offering higher communication efficiency.

---

1 Background Knowledge

Definition 1 (0-RTT Key Exchange). Taking Diffie-Hellman key exchange as an example, a user first obtains shared information from the server through previous key exchanges, then selects an exponent x to generate a key, and sends encrypted data and the key exchange message to the server. Upon receipt, the server selects y to generate a key and sends encrypted data and the key exchange message back to the user. In this communication, both parties use the derived value as the session key. When encrypted information and key exchange information are sent simultaneously during key negotiation, this is called 0-RTT key exchange.

Definition 2 (Puncturable Encryption). In a puncturable forward-secret key encapsulation scheme, assume a server possesses a long-term private key sk. When receiving a ciphertext message c that encapsulates a session key, the server decrypts c using sk and derives a new private key sk'. The new private key is punctured at the time of c and can be used to decrypt all ciphertexts except c, after which the server deletes the old key.

1.1 Lattice Preliminaries

[Technical content about lattices with preserved mathematical notation]

1.2 Security Model

In the research of provable security for AKE protocols, Bellare and Rogaway [**?**] made pioneering contributions. Their work [**?**] provided the first formal treatment of entity authentication and authenticated key distribution suitable for distributed environments, discussed mutual authentication and authenticated key exchange in symmetric and two-party settings, and presented definitions, protocols, and proofs for each scenario under the assumption of a pseudorandom function.

This paper adopts the protocol security analysis model from the GHJL17 scheme [**?**]. Let $\mathcal{I}$ denote the set of identities modeling clients (C) and servers (S) in the system, each associated with a public/private key pair (pk_u, sk_u). The public key component pk_u is generated once and fixed, while the private key sk_u can be modified by the relevant participant' s session over time. Additionally, each identity u maintains a local current time variable τ_u initialized to 1. In the

security model, adversary $\mathcal{A}$ interacts with multiple identity sessions and runs the forward-secret one-pass key exchange protocol. Session π_u^i represents the i-th session of identity u and is associated with the following internal state variables:

- π_u^i.role $\in \{\text{client}, \text{server}\}$: indicates the session' s role.
- π_u^i.owner $= u$: indicates the session owner (e.g., u owns π_u^i).
- π_u^i.pid $\in \mathcal{I} \cup \{\perp\}$: indicates the intended communication partner and is set exactly once. For example, π_u^i.pid $=\perp$ indicates the client is not authenticated. Initially, π_u^i.pid can also be set to $\perp$ to indicate the client identity needs to be learned in the protocol.
- π_u^i.trans $\in \{0,1\}^* \cup \{\perp\}$: records the single sent and received message.
- π_u^i.time $\in \mathbb{N}$: records the time interval when processing sent and received messages.
- π_u^i.key $\in \{0,1\}^* \cup \{\perp\}$: is the session key derived in the session, referenced by specific session state variables.
- π_u^i.keystate $\in \{\text{fresh}, \text{revealed}\}$: indicates whether the session key has been compromised, initially set to fresh.

Definition 7 (Matching Sessions). Two sessions π_u^i and π_v^j are matching if they satisfy the following conditions: - π_u^i.trans $= \pi_v^j$.trans: the two sessions share the same transmission. - π_u^i.time $= \pi_v^j$.time: the two sessions run in the same time interval. - π_u^i.role $=$ client $\wedge$ π_v^j.role $=$ server: the two sessions have opposite roles. - π_u^i.pid $= v \wedge \pi_v^j$.pid $= u$: the server session is from the client' s expected matching partner, or the client session belongs to the server' s expected partner, or the server considers its partner unauthenticated.

Assume adversary $\mathcal{A}$ controls the network and is responsible for transmitting messages, allowing arbitrary modification, deletion, or reordering of messages. It can interact with the key exchange protocol and sessions through the following queries:

- NewSession$(u, \text{role}, \text{pid}, m)$: Initializes a new session for identity $u \in \mathcal{I}$ (where the server will be role $=$ server and pid $=\perp$ indicates an unauthenticated client partner). If role $=$ server, returns m; when role $=$ client, returns $\perp$, otherwise returns m.
- Reveal(π_u^i): If the session key is derived, reveals the session key of the specific session. If π_u^i.key $\neq\perp$, returns the key, otherwise returns $\perp$.
- Corrupt(u): Corrupts the long-term state of identity $u \in \mathcal{I}$. This query can be made at most once per identity u, and no further queries are allowed for session u after corruption. Let corr_u denote the time when corruption occurs, initially set to ∞.
- Tick(u): Forwards the state of identity u one time step. Records the new time as $u.\tau \leftarrow u.\tau + 1$.
- Test(π_u^i): Allows the adversary to challenge the derived session key and is queried only once. This oracle randomly selects a secret bit $b \in \{0,1\}$ in the security game. If $b = 0$, returns the real session key π_u^i.key; other-

wise returns a randomly selected key from the protocol-specific probability distribution. Sets test $\leftarrow \pi_u^i$.

---

2 Lattice-Based 0-RTT Key Exchange Protocol with Forward Secrecy

To construct our protocol, we first implement a one-time signature scheme (OTSIG) based on the SIS problem on lattices in the standard model using the MP12 trapdoor [**?**]. Next, we design a hierarchical identity-based key encapsulation mechanism (HIBKEM) in the standard model on lattices. Based on these two building blocks, we construct a puncturable forward-secret key encapsulation mechanism (PFSKEM). Finally, we design a lattice-based 0-RTT key exchange protocol with forward secrecy using the puncturable forward-secret key encapsulation mechanism. The concrete construction is as follows.

2.1 One-Time Signature Scheme

A one-time signature scheme OTSIG consists of three probabilistic polynomial-time algorithms (OTSIG.KGen, OTSIG.Sign, OTSIG.Vfy).

- **OTSIG.KGen**(1^n): The algorithm inputs a security parameter n, selects a matrix A from distribution D, and chooses a vector v. It outputs a public key pk and a secret key sk.
- **OTSIG.Sign**(sk, μ): The algorithm uses the trapdoor R of matrix A (containing the trapdoor for an extended matrix) to sample vector v from the distribution. Here μ is the i-th bit of the message, treated as an integer. The signature is generated using the SampleR algorithm.
- **OTSIG.Vfy**(pk, μ, σ): The verification algorithm accepts if the verification equation holds, otherwise rejects.

2.2 Hierarchical Identity-Based Key Encapsulation

A hierarchical identity-based key encapsulation mechanism HIBKEM consists of four probabilistic polynomial-time algorithms (HIBKEM.KGen, HIBKEM.Del, HIBKEM.Encap, HIBKEM.Decap).

- **HIBKEM.KGen**$(1^n, 1^d)$: Inputs a security parameter n and maximum hierarchical depth d, generates a uniformly random matrix A and trapdoor matrix R for A, and selects n-dimensional uniformly random vectors. It outputs a master public key MPK and master secret key MSK, explicitly defining the identity space ID and key space K.
- **HIBKEM.Del**$(MPK, SK|_{id}, id)$: Inputs the master public key MPK, parent user identity id, and child user identity. Computes the invertible matrix and uses the dual Regev algorithm to encrypt key k: first selects the receiver user identity, then selects error terms, and outputs the trapdoor matrix $SK|_{id}$.

- **HIBKEM.Encap**(MPK, id): Inputs the master public key MPK and user identity id, runs the encapsulation algorithm to generate a uniformly random vector, then computes the encryption ciphertext and outputs the ciphertext CT and key K.
- **HIBKEM.Decap**$(MPK, SK|_{id}, CT)$: Inputs the master public key MPK, user secret key matrix $SK|_{id}$, and ciphertext CT, where the user identity has hierarchical depth d. Let the Gaussian parameter be σ. The user performs the same operations as the encapsulation algorithm to obtain CT', runs the preimage sampling algorithm, and outputs 1 if the verification succeeds, otherwise outputs 0.

HIBKEM Correctness: The correctness of HIBKEM decryption is characterized by Theorem 1.

Theorem 1. HIBKEM decryption is correct. For any $id \in ID$, where ID is the identity space, we have:

$$\Pr[\text{Decap}(MPK, SK|_{id}, \text{Encap}(MPK, id)) = K] = 1 - \text{negl}(n)$$

Proof. The output of the HIBKEM decryption algorithm is $K' = \lfloor (s^T(u+e)) \rceil$, where e is the error term with absolute value less than $q/5$. Given the parameter settings of Theorem 1, the theorem holds.

HIBKEM Security: In the selective-ID game, the advantage of adversary $\mathcal{A}$ is defined as:

$$\text{Adv}^{\text{IND-sID-CPA}}_{\text{HIBKEM},\mathcal{A}}(n) = |\Pr[\text{Exp}^{\text{IND-sID-CPA}}_{\text{HIBKEM},\mathcal{A}}(n) = 1] - 1/2|$$

The following describes the selective-ID CPA security experiment for hierarchical identity-based key encapsulation between challenger C and adversary $\mathcal{A}$:

a) $\mathcal{A}$ inputs its target identity id^* to be challenged.
b) C generates $(MPK, MSK) \leftarrow \text{HIBKEM.KGen}(1^n, 1^d)$.
c) $\mathcal{A}$ can query HIBKEM.Del$(MPK, SK|_{id}, id)$. The challenger sends the private key for requested identity id. The only restriction is that $\mathcal{A}$ cannot query the oracle for the private key of id^* or its ancestors.

If for all probabilistic polynomial-time adversaries $\mathcal{A}$, the advantage function $\text{Adv}^{\text{IND-sID-CPA}}_{\text{HIBKEM},\mathcal{A}}(n)$ is a negligible function in security parameter n, then the hierarchical identity-based key encapsulation mechanism HIBKEM is selective-ID CPA secure (IND-sID-CPA).

2.3 Puncturable Forward-Secret Key Encapsulation

A puncturable forward-secret key encapsulation mechanism PFSKEM consists of five probabilistic polynomial-time algorithms (PFSKEM.KGen, PFSKEM.Encap, PFSKEM.PnctCxt, PFSKEM.Decap, PFSKEM.PnctInt).

- **PFSKEM.KGen**(1^n): The algorithm inputs a security parameter n and generates public key PK and secret key SK.
- **PFSKEM.Encap**(PK, τ): The algorithm inputs the public key and time interval τ, generates a random key K, and computes the ciphertext CT.
- **PFSKEM.PnctCxt**(SK, CT, τ): Parses SK as $SK|_{id}$, let T denote the HIBKEM tree. Computes the punctured key and outputs the new secret key SK'.
- **PFSKEM.Decap**(SK, CT, τ): The decryption algorithm parses SK as $SK|_{id}$. a) If CT contains a label owned by $SK|_{id}$, output K; b) If CT contains a label that is an ancestor of a node in $SK|_{id}$, compute and output K; c) Otherwise output $\perp$.
- **PFSKEM.PnctInt**(SK, τ): The next time interval key update algorithm computes the new secret key for the next time interval $\tau + 1$, where T is the HIBKEM tree. Outputs the secret key for the next time interval.

PFSKEM Security: The security of PFSKEM is defined through the IND-sT-CCA game. The advantage of adversary $\mathcal{A}$ is:

$$\mathrm{Adv}^{\text{IND-sT-CCA}}_{\text{PFSKEM},\mathcal{A}}(n) = |\Pr[\mathrm{Exp}^{\text{IND-sT-CCA}}_{\text{PFSKEM},\mathcal{A}}(n) = 1] - 1/2|$$

If this advantage is negligible for all PPT adversaries, the scheme is secure.

---

3 Performance Analysis

This section compares our protocol with the PAKE protocol proposed by Katz et al. [**?**] and the PAKE protocol proposed by Zhang et al. [**?**] from both security and efficiency perspectives. All these protocols are constructed from lattice hard problems. The performance of the three protocols is shown in Table 1 .

In terms of security, our protocol introduces a puncturable encryption mechanism, providing full forward secrecy. Compared with protocols [**?**][**?**], our protocol can resist replay attacks and offers higher security. In terms of efficiency, the protocol is constructed from one-time signature techniques and hierarchical identity-based key encapsulation, requiring only 1 communication round to complete key agreement. As shown in Table 1, compared with protocols [**?**][**?**], our protocol has smaller communication overhead, mainly consisting of the ciphertext from hierarchical identity-based key encapsulation and its signature, which improves communication efficiency.

Table 1 Performance comparison of three schemes

Scheme	Type	Rounds	Forward Secrecy	Communication Overhead
Katz et al.	2-party	3	No	$2m + 4l - 2$
Zhang et al.	2-party	2	No	$m + 1$
Our protocol	2-party	1	Yes	$m + 3n$

Protocol [**?**] is an LWE-based 2PAKE protocol requiring 3 communication rounds. The communication cost, determined by ciphertext, projection key, and message authentication code, is $2m + 4l - 2$, which is $m + 4l - 3$ more than our protocol. Additionally, our protocol provides full forward secrecy through puncturable encryption. Based on the analysis and Table 1, our protocol not only has stronger forward secrecy but also higher communication efficiency.

Protocol [**?**] is a lattice-based 2PAKE protocol requiring 2 communication rounds. The communication cost mainly depends on ciphertext and projection key sizes. Compared with protocol [**?**], our protocol has communication overhead of only $m + 3n$, with increased server and user computational overhead. Meanwhile, our protocol requires only 1 communication round and provides stronger forward secrecy. The analysis and Table 1 show that our protocol has lower communication efficiency but higher security.

The above analysis demonstrates that our constructed key exchange protocol achieves full forward secrecy through key puncturing update strategies, can resist replay attacks and quantum attacks, and realizes 0-RTT communication by sending only one message containing encrypted protected payloads and a key exchange protocol message to the server without requiring a server response, greatly reducing communication overhead. Therefore, our protocol is feasible.

---

4 Conclusion

This paper proposes a 0-RTT key exchange protocol where the one-time signature technique is based on the SIS hard problem on lattices and the hierarchical identity-based key encapsulation mechanism is based on the LWE hard problem on lattices, which is significant in the post-quantum era. The designed 0-RTT key exchange protocol requires only server authentication, sending the session key and transmitted message together to the server, thereby achieving 0-RTT communication. The protocol can effectively resist quantum and replay attacks, improving application security, and we provide a rigorous security proof in the standard model. While the puncturing operation for key update may consume considerable time due to security requirements and deployment parameters, this can be optimized by performing a small amount of effective computation and deleting partial private keys in the binary tree to reduce time consumption.

References

[1] Diffie W, Hellman M. New directions in cryptography [J]. IEEE Trans on Information Theory, 1976, 22 (6): 644-654.

[2] Shor, Peter W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer [J]. SIAM Journal on Computing, 1997, 26 (5): 1484-1509.

[3] Hu Zhao, Zhu Yuesheng, Ma Limin. An improved Kerberos protocol based on Diffie-Hellman-DSA key exchange [C]// IEEE International Conference on Networks. Piscataway, NJ: IEEE Press, 2012: 400-404.

[4] Hu Xuexian, Liu Wenfen, Zhang Jianhui. An Efficient ID-Based Authenticated Key Exchange Protocol [C]// Wase International Conference on Information Engineering. Piscataway, NJ: IEEE Press, 2009: 229-233.

[5] Dierks T, Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.2, RFC 5246 [EB/OL]. [2008-08]. https://www.rfc-editor.org/info/rfc5246.

[6] Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3 [EB/OL]. [2016-10]. https://tools.ietf.org/html/draft-ietf-tls-tls13-18.

[7] Krawczyk H. HMQV: A high-performance secure Diffie-Hellman protocol [C]// Advances in Cryptology -CRYPTO 2005. CRYPTO 2005. Lecture Notes in Computer Science. Berlin: Springer, 2005: 546-566.

[8] QUIC, a multiplexed stream transport over UDP [EB/OL]. https://www.chromium.org/quic.

[9] Günther F, Hale B, Jager T, et al. 0-RTT Key Exchange with Full Forward Secrecy [C]// International Conference on the Theory & Applications of Cryptographic Techniques. Berlin: Springer, 2017: 519-548.

[10] Canetti R, Halevi S, Katz J. A Forward-Secure Public-Key Encryption Scheme [M]. Advances in Cryptology —EUROCRYPT 2003. Berlin: Springer, 2003: 255-271.

[11] Green M D, Miers I. Forward Secure Asynchronous Messaging from Puncturable Encryption [C]// IEEE Symposium on Security & Privacy. Piscataway, NJ: IEEE Press, 2015: 305-320.

[12] Blazy O, Kiltz E, Pan J. (Hierarchical) Identity-Based Encryption from Affine Message Authentication [M]// Advances in Cryptology -CRYPTO 2014. Berlin: Springer, 2014: 408-425.

[13] Derler D, Jager T, Slamanig D, et al. Bloom Filter Encryption and Applications to Efficient Forward-Secret 0-RTT Key Exchange [C]// International Conference on the Theory & Applications of Cryptographic Techniques. Berlin: Springer, 2018: 425-455.

[14] 魏福山, 马建峰, 李光松, 等. 标准模型下高效的三方口令认证密钥交换协议 [J]. 软件学报, 2016, 27 (9): 2389-2399. (Wei Fushan, Ma Jianfeng, Li Guangsong, et al. Efficient Three-Party Password-Based Authenticated Key Exchange Protocol in the Standard Model [J]. Journal of Software, 2016, 27 (9): 2389-2399.)

[15] Katz J, Vaikuntanathan V. Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices [C]// Advances in Cryptology-ASIACRYPT 2009, the 15th International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2009: 636-652.

[16] Ding Yi, Fan Lei. Efficient password-based authenticated key exchange from lattices [C]// Seventh International Conference on Computational Intelligence and Security. Piscataway, NJ: IEEE Press, 2012: 934-938.

[17] Groce A, Katz J. A New Framework for Efficient Password-Based Authenticated Key Exchange [C]// Proceedings of the 17th ACM Conference on Computer and Communications Security. New York: ACM Press, 2010: 516-525.

[18] Zhang Jiang, Yu Yu. Two-Round PAKE from Approximate SPH and Instantiations from Lattices [C]// International Conference on the Theory and Application of Cryptology and International Security. Berlin: Springer, 2017: 37-67.

[19] 李子臣, 谢婷, 张卷美, 等. 基于 RLWE 的后量子认证密钥交换协议 [J]. 计算机研究与发展, 2019, 56 (12): 2694-2701. (Li Zichen, Xie Ting, Zhang Juanmei, et al. Post Quantum Authenticated Key Exchange Protocol Based on Ring Learning with Errors Problem [J]. Journal of Computer Research and Development, 2019, 56 (12): 2694-2701.)

[20] Bellare M, Rogaway P. Entity Authentication and Key Distribution [C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 1993: 232-249.

[21] Micciancio D, Peikert C. Trapdoors for lattices: simpler, tighter, faster, smaller [C]. Advances in Cryptology –EUROCRYPT 2012. EUROCRYPT 2012. Lecture Notes in Computer Science. Berlin: Springer, 2012: 700 –718.

Note: Figure translations are in progress. See original paper for figures.