Developing Secure Messaging Software using Post-Quantum Cryptography

Tat-Thang Nguyen
University of Transport and Communications, Vietnam
mrthanglc@gmail.com

Nhu-Quynh Luc
Academy of Cryptography Techniques, Vietnam
quynhln@actvn.edu.vn (corresponding author)

Toan Thanh Dao
University of Transport and Communications, Vietnam
daotoan@utc.edu.vn

Received: 21 October 2023 | Revised: 11 November 2023 | Accepted: 13 November 2023

 DOI: https://doi.org/10.48084/etasr.6549

ABSTRACT

In this paper, a technique to develop a secure messaging service utilizing a new post-quantum cryptosystem, termed CryptoMess, is proposed. Commutative Supersingular Isogeny Diffie-Hellman (CSIDH) is utilized to secure key exchange paired with the AES algorithm to protect message content in communication. At the same time, the Rainbow post-quantum digital signature technology is incorporated to assure the integrity and authenticity of communications between the sender and the recipient. As a consequence, the messaging program is able to exchange messages between users, assuring safety, security, integrity, and authenticity. The performance of the program has a transmitting rate of approximately 0.26 s and a receiving rate of approximately 0.22 s. The message signing time is approximately 0.027 s, the message verification speed is approximately 0.22 s, and the key exchange time is approximately 0.0017s.

Keywords-CSIDH; AES; Rainbow; UOV; post quantum

I. INTRODUCTION

The problem of safeguarding data when the latter is communicated through public channels is particularly crucial for practical applications. Protecting privacy is an issue when utilizing various media platforms [1]. Currently, there is a wide range of large corporations, such as Facebook and Google, which have developed messaging software. However, when initially accessed, most software does not contain features to secure the content of the message. Numerous firms like Viber, Wickr, and Telegram have enhanced their services to fortify their clients privacy [1, 2]. To protect personal data, most firms have selected security solutions including AES, RSA, ECDH, and ECDSA, accessible in cryptographic libraries such as OpenSSL, and OpenVPN [3, 4]. Authors in [5, 6] brought in several Rainbow schematic implementations based on UOV and AVX2 usage. Rainbow post-quantum signing has the advantage of providing very small signatures of only a few hundred bits (528 bits) which are much shorter than other post-quantum signature schemes [7-10]. Among the Post-quantum cryptographic algorithms published by NIST, the Diffie-Hellman key exchange protocol using isogeny mapping on super singularity elliptic curves (SIDH: Super-singular Isogeny Diffie–Hellman) and Rainbow post-quantum digital signature scheme emerge as potential candidates to ensure the secure communication of quantum computer systems [5, 7-9, 11-13]. For these applications, authors in [14] aimed to apply AES symmetric key cryptographic solutions to protect personal data. Authors in [12, 13, 15, 16] applied the key exchange solution CSIDH (Commutative Super-singular Isogeny Diffie-Hellman) to verify that the communication channel for the designed application is sheltered against a variety of existing threats.

In this paper, Rainbow post-quantum digital signature scheme is utilized in the application to secure the integrity and authenticity of the user's message contents. The specifics of the successful integration of cryptographic solutions into cryptographic modules and the findings gained in this research are detailed.

II. RELATED WORK

A. IP Encapsulation utilizing AES Encryption to Safeguard Message Data

Nowadays, the Internet Protocol Version 4 (IPv4) is most utilized [17]. Typically, the IPSec protocol has been intended to guarantee secured encrypted data in transfer [18]. Cryptographic algorithms such as AES, are widely used to

encrypt data before encapsulating them for transmission over a public channel [3]. Authors in [19] employed IPv4 as a communication protocol. The application's encapsulated data for transmission is a TCP/IP protocol that employs a client/server communication paradigm, in which a user or device (client) is provided a message by another computer (server) to perform a service (such as delivering a web page) through the network. The packet is deconstructed and the AES cipher algorithm is used to encrypt the data. The data are then repackaged and sent over public communication channels using the existing network architecture. The employed AES encryption technique assures that the design adheres to the NIST encryption requirements.

B. The Key Exchange Protocol CSIDH Guarantees a Safe Channel

The idea of isogeny on super-singular elliptic curves is extensively employed in cryptography, such as in public key encryption techniques based on SIKE (Super-singular Isogeny Key Encapsulation) isogeny and the CSIDH key exchange protocol. SIKE is a public key cryptographic method based on isogeny that works by substituting pseudo-random stages in seed schemes. CSIDH is used to create a secret key between two parties across an insecure communication channel. It is comparable to the Diffie-Hellman key exchange but is based on the stages of the seed scheme and is meant to withstand an assault by examining the code of an opponent using a quantum computer. CSIDH claims one of the shortest key sizes. After compression, CSIDH employs a 2688-bit public key at 128-bit quantum security key. CSIDH further separates itself from comparable systems like NTRU (Number Theory Research Unit) and Ring-LWE by enabling complete forward secrecy, a trait that prevents long-term keys from being compromised and affecting the security of the old session messages. These qualities make CSIDH a good contender to replace the extensively used Diffie-Hellman (DH) key exchange protocol and the widely used DH over the elliptic curve (ECDH) key exchange protocol in present Internet communication.

C. Rainbow Post-Quantum Digital Signature Scheme for Data Integrity and Authenticity

The scientific basis of this scheme is based on the theory of multivariable mathematical functions built on finite fields, but it has promoted the advantage of improving the efficiency of the execution speed for the processes of key generation, digital signing, and signature validation [7]. With the desire to be able to use the Rainbow digital signing scheme on hardware platforms and low-resource devices, Petzoldt and his colleagues have come up with other variants such as CZ-Rainbow (Circumzenithal Rainbow) and Compressed-Rainbow which have optimized the key pair size for the Rainbow digital signing scheme [5]. In this paper, we opted to install the post-quantum digital signature system to digitally sign the message data before transferring it to the transmission channel. The purpose of this proccess is to secure the integrity and validity of the user's message data while the program is functioning.

D. Architecture of CSIDH Key Exchange Protocol in Software

In this work, we employed the key exchange CSIDH to assure the key distribution procedure to compute the shared session key in order to ensure the safe transmission of users A and B. Initialization parameters for the calculation process include characteristics $p = 4\prod_{i=1}^{n}\ell_i - 1$ where ℓ_i represents odd primes permanently installed. The original curve used is a montgomery elliptic curve $E_{a,b} : by^2 = x^3 + ax^2 + x$, with $b(a^2 - 4) \neq 0$. These settings, are based on the NIST published standard for post-quantum cryptography in the 3rd round, have been chosen [7, 11].

- For user A: The secret key is k_A (the value of k_A is a random integer in the interval $\left(0, \ell_A^{e_A}\right]$). User A's public key is $PKA\left(\phi_A\left(E_{a_0}\right), \phi_A(P_B), \phi_A(Q_B)\right)$. The corresponding is $\left(a_{e_A}, P'_B, Q'_B\right)$ at the last 2-variant calculation.
- For user B: User B's secret key is k_B (the value of k_B is a random integer in the interval $\left(0, \ell_B^{e_B}\right]$). The public key is $PKB\left(\phi_B\left(E_{a_0}\right), \phi_B(P_A), \phi_B(Q_A)\right)$. The corresponding is $\left(a_{e_B}, P'_A, Q'_A\right)$ at the last 3-variant calculation.

The process to calculate the session key shared between user A and user B is as follows. For user A, his shared session key is SK_A, and for user B is SK_B. These SK_A and SK_B values correspond to the invariant value j of the final curve with coefficients a_{e_A}, $j(E_{a_{e_A}})$, a_{e_B} and $j(E_{a_{e_B}})$ calculated according to [12, 13, 15, 16].

E. Design of the Rainbow Digital Signing Scheme of the Software

The process of key generation, digital signing, and authentication of the Rainbow digital signature scheme is implemented as follows:

- Generate key: Public key, $pk = S \circ F \circ T$, Secret key $sk = (InvS, c_S, F, InvT, c_T)$. The validity of the key is checked if *pk* and *sk* are orthogonal through the expression: $pk \; x \; sk^* = 0 \; mod \; q$.
- Digital signature: First, the application will perform a hash calculation according to the hash function $H(d) : \{0,1\}^* \rightarrow \mathrm{F}^m$ to calculate the hash value $h = H(d)$. Next, the application calculates the inverse values according to the formula $x = S^{-1}(h), y = F^{-1}(x), z = T^{-1}(y)$. Finally, the application generates a digital signature for the message as $z \in \mathrm{F}^n$.
- Selecting a set of parameters to ensure the safety of the Rainbow schema: The input parameters to the Rainbow signing scheme are important to certify a secure digital

signing process. According to [7, 20] for each type of Rainbow's domain parameter as input, a corresponding quantum gate number is required to break the security of the digital signing scheme. With the type 1 parameter, type of the post-quantum digital signature scheme Rainbow, the domain parameter is $(GF(16), 36, 32, 32)$, 162 quantum gates are needed for the cryptanalyst to attack the digital signature scheme.

III. RESULTS AND DISCUSSION

A. Build CryptoMess Software using CSIDH, Rainbow, and AES

For the secure messaging module using CSIDH and AES-256 (called CryptoMess), in this study, when designing and building the CryptoMess module, we used Socket encapsulation as announced and TCP/IP protocol to program the transmission and reception for the application on the internet environment. Figure 1 shows the operation details of the CryptoMess module.

- Server module of CryptoMess (Figure 1(a)): The server's task only includes receiving packets from the sender, the stream part, and sending them to the correct receiver address. In addition, we also added a function to check the login account for the Server.
- Client module of CryptoMess (Figure 1(b)): Authors in [15] designed and built the Client module including the main components: login module, chat and messaging module, CSIDH key generation, CSIDH key distribution and packet encryption using AES-256. Once the key has been successfully generated, key exchange distribution according to the CSIDH schema is performed. This process is done automatically in the software. After the key generation and key exchange process, the application will have a public key and a private key used for the AES algorithm to encrypt and decrypt the message, where the user will alternately change the roles of sender and receiver.

The role of the message sender will proceed as follows. The message content in a clear form will be signed by the software, and then padding is added to the message. After completing the above two processes, the message will be encrypted with the AES-256 algorithm, encapsulated, and sent to the Server. Considering the role of the message recipient, after receiving the packet containing the message content in encrypted form, the software will conduct "peeling" to the packet to get the message content in encrypted form and then decrypt it. This process will be reversed for the sender. The encrypted message will remove the padding, and then verify the message. Successful message verification will show plaintext on the recipient's screen.

1) Security Assessment for the CSIDH Key Exchange Module of CryptoMess

The security of the CSIDH key exchange scheme depends on the problem of ensuring quantum security. For CSIDH, the used curve parameter is the super singularity curve on the field F_{p^2}. The base points of users A and B are different. With the private and public keys, the ECDH scheme uses scalar point multiplication, and the CSIDH scheme uses isomorphism, the image of the curve, the base pixel, and the j-invariant value. The design of the ECDH scheme depends on the complexity of solving a discrete logarithm problem on the curve. Meanwhile, the design of CSIDH depends on the complexity of solving the isomorphism problem when knowing the curve image and two public base points.

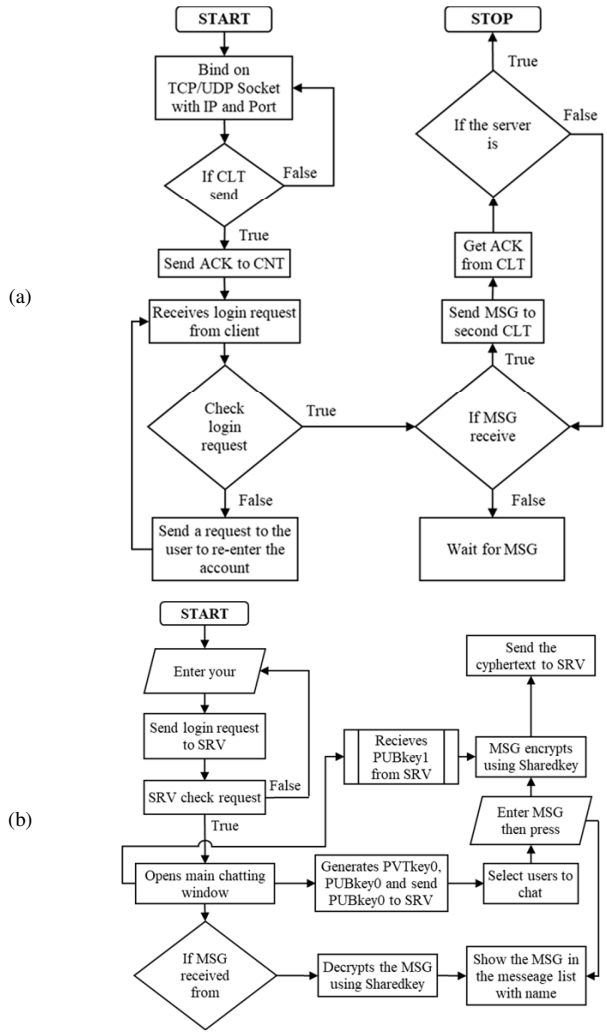

Fig. 1. Operation flow design of CryptoMess: (a) Server, (b) client.

In [15, 16], the SIDH protocol has a security based on the problem: "Given E, the image of the curve $E' = \phi(E)$ and two public base points find the isogeny ϕ". Authors in [21] came up with a method for a successful attack on the SIDH protocol based on Kani's Glue and split theory. The authors pointed out two weaknesses of the attacked SIDH protocol: (1) The public key contains the image of the base points, (2) the isogenies used for secret keys are of a fixed degree.

CSIDH protocol is a post-quantum key exchange protocol proposed in 2018, using isogeny mapping on a supersingular elliptic curve [13]. The important difference between CSIDH

and SIDH lies in its properties: There is no isogeny of a secret key of fixed degree. The security of the CSIDH protocol depends on the problem: "Given two Montgomery curves E_A and E_B, find the isogeny ϕ, such that $E_B = \phi_B . E_A$" [13]. Through these analyses, authors in [22] state that a specific attack method on the CSIDH protocol described in [21] has not been found and the CSIDH protocol has not yet been broken.

2) Theoretical Security Assessment for the Rainbow Post-Quantum Digital Signing Scheme

To reduce the size of the public key while ensuring message security, authors in [11] proposed a variant called CZ-Rainbow. Unlike Standard Rainbow, CZ-Rainbow does not use a cyclic matrix, but uses a PRNG pseudo-random number generator to generate a part of the public key, called the public key *seed* [5, 6]. Then, two linear mappings are randomly selected and a PRNG is used to generate a public key and compute the central mapping. From this, it is possible to generate a random number of key pairs with the length of the key seed used in the key generation process. So, instead of storing the entire public key, the emulator only needs to store the seed and the generated part of the public key. In this way, the size of the public key of the CZ-Rainbow scheme can be reduced by up to 70% and the key generation time is only slightly slower than that of Standard Rainbow. According to [8, 9, 20, 23, 24], the size of the public key and the size of the digital signature are proportional to the size of the input parameters but still ensure the properties of short digital signatures. Compressed Rainbow has effectively optimized the key size and still establishes the security of the digital signature for the message. The computational complexity of the Rainbow digital signature scheme is guaranteed for efficient and secure execution by several points: Key generation $O(n^2)$, digital signing time $O(n^3)$, and signature validation $O(n^3)$ with a public key size of n^3. This shows that with the first domain parameters selected as type 3 the application of the post-quantum digital signing Rainbow scheme, based on the UOV unbalanced digital signature scheme, is secure before several current attacks such as attacks on immutable subspaces, and attacks in between.

3) Working Principle of CryptoMess

After the user successfully creates an account, he will proceed to log in. Before starting to message each other, Users A and B, will conduct a handshake to exchange keys, generated by the key generation algorithm which uses the isomorphism theory on the super-singular elliptic curve - CSIDH. After this process is done successfully, the users may start messaging.

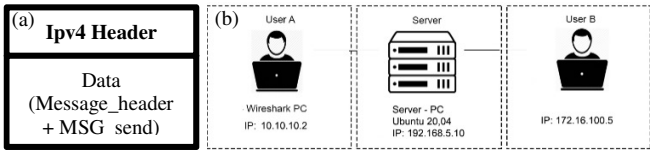

Fig. 2. (a) Extraction and encapsulation of encrypted DATA in IPv4, (b) Configuration of a security system for the CryptoMess encrypted messaging application between users A and B.

Figure 2(a) is a detail of the process of extracting and encapsulating data to encode the IPv4's DATA frame when sent to the channel and decode them when received. Message_header contains data of len(MSG_Send) and Header_Length, MSG_send contain data of Request Send, username, encrypted MSG, and signature. For User A, after entering the message content and pressing send, the software will encrypt the message with the AES - 256 block code, digitally sign the message, and send it to the Server, then it will resend that message in encrypted form to User B. When User B receives the message, the software will verify the signature, then decode and display the software interface. The above process is reversed when User B sends it back to User A.

B. CryptoMess Result Evaluation when integrating CSIDH and Rainbow

In this study, we created, wrote, and performed the CryptoMess software module on a computer with the following configuration: Intel(R) Core i5-4200U, CPU @ 1.60 GHz, up to 2.30 GHz, RAM: 8.00 GB. We utilized Wireshark (IP address: 10.10.10.2/24) installed on User A to capture the package and prove the message content is transferred to the server and back from the server to the receiver in encrypted form. Figure 2(b) depicts the secure messaging operation test model. This type of model includes two users, user A (with IP address: 10.10.10.2/24) and user B (with IP address: 172.16.100.5/24), as well as the server (with IP address: 192.168.5.10/24). The distinction is that this program sends plaintext, and the team also utilizes Wireshark to collect incoming and outgoing packets from this software to test it. The goal in continuing to use such software is to enhance credibility. Users A and B used it to install the program CryptoMess in order to communicate. Wireshark was installed on User A's PC to intercept and capture packets to verify the content of the message delivered to the Server and vice versa in encrypted form during transmission using CryptoMess. Figure 3 depicts the outcome of the CryptoMess software operation at User A, User B, and Server, with a secure transmission channel (according to CSIDH key distribution protocol), message data encryption (using AES 256-bit cryptography), and data integrity and authentication using Rainbow digital signing protocol. Figure 3(a) portrays the program's interface and how to send User A's encrypted message to User B. Figure 3(b) illustrates the interface and how a message is sent when User B sends to User A. Figure 3(c) describes in detail the behavior of the software when User A sends a message to User B and Figure 3(d) thoroughly describes how the software works when User A's message reaches User B's software.

To test the transmission time of CryptoMess, we utilized message strings of varying length samples to determine the software's transmission time. We selected the times at which the program reads the sender's message, encrypts AES-256, signs Rainbow, wraps the packet, and transmits it. When the program gets the packet from the Sever, it verifies the Rainbow signature, decrypts AES-256, and shows it on the receiver's screen, the designated receiving time starts.

(a)

(b)

(c)

(d)

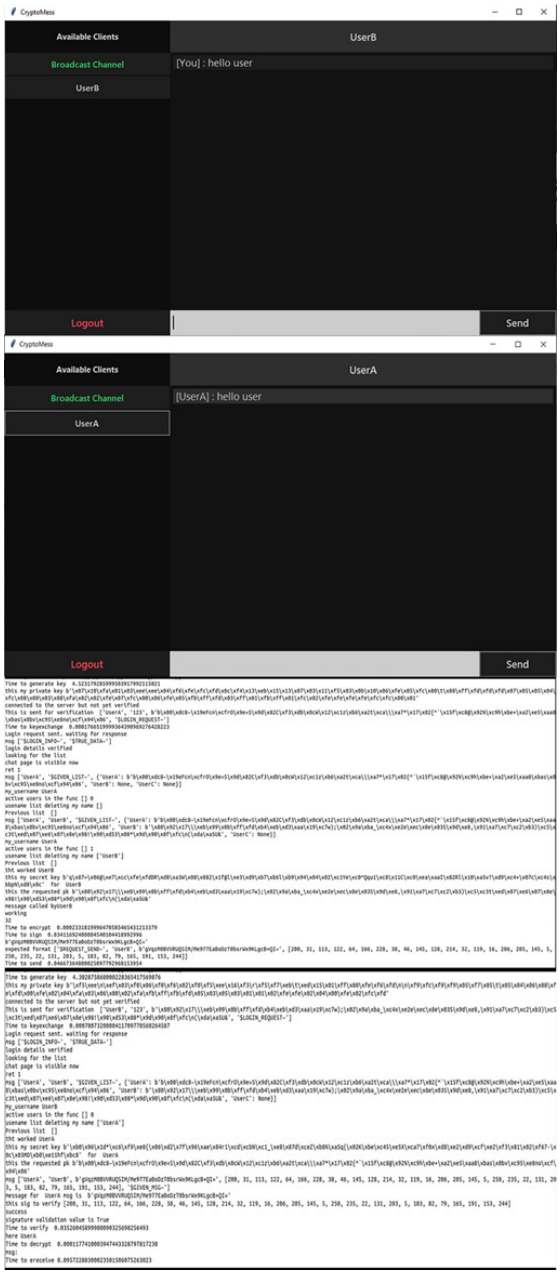

Fig. 3. (a) and (b) CryptoMess behavior when Users A and B text. (c) and (d) Details of the activity on the computers of Users A and B when texting.

When employing the Rainbow quantum post-signing system for digital signature and authentication of message data, the Rainbow digital signing time takes around 0.43363 s and the Rainbow signature validation time takes about 0.22198 s. This demonstrates that the CryptoMess software is fairly quick and fits the criteria for practical application. The results can be seen in Table I. It can be observed that the CryptoMess program has pretty high encryption and decryption speed, as well as a relatively short transmission time, less than RSA. Furthermore, CryptoMess permits communications with text lengths of 2048 characters and higher, whilst the RSA is still in the process of verifying the author group's receipt. The program displays an error if you observe a character length of 2048. As a consequence, CSIDH key creation takes about 9 s, CSIDH key exchange takes approximately 0.0016 s, Rainbow signature takes approximately 0.27 s, and signature validation takes approximately 0.22 s. CryptoMess software operates reliably, improves runtime, assures safety, and fits the real needs of today's applications.

Post-quantum cryptography can be used securely for quantum computers. In Table II [10, 25] we can see a comparison with Rainbow post-quantum cryptosystems on the same machine. We analyzed and evaluated the CryptoMess software source code of the design team using the Fortify Static Code Analyzer toolkit (Version 22.1.0.0166). The test results from the toolkit show that the source code of the CryptoMess software is safe, reliable, and suitable for the real needs of today's applications.

TABLE I. CRYPTOMESS SOFTWARE EXECUTION TIME

Function	Sample length (bit)	CryptoMess (s)
Message sent (s)	256	~0.002371
	512	~0.004221
	2048	~0.00548
Message received (s)	256	~0.002473
	512	~0.002512
	2048	~0.013495
Encrypt (s)	256	~0.001658
	512	~0.001922
	2048	~0.002016
Decrypt (s)	256	~0.000174
	512	~0.000246
	2048	~0.000287

CryptoMess key creation and distribution calculation time				
User	**CSIDH key generator (s)**	**CSIDH key exchange (s)**	**Rainbow Digital Signing (s)**	**Rainbow Signature Validation (s)**
User A	~8.52317	~0.00161	~0.02491	~0.23804
User B	~8.30287	~0.00176	~0.02852	~0.21526

TABLE II. THE RESULTS OTHER POST-QUANTUM DIGITAL SIGNATURE SCHEMES [10, 25]

Algorithm	**Signature (bytes)**	**Sign** (ms)	**Verify** (ms)
Dilithium2	1184	0.050	0.036
qTESLA-P-I	14390	1.055	0.312
Picnic-L1-FS	33	3.429	2.584

IV. CONCLUSION

In this paper, we have implemented the integration of new post-quantum cryptographic solutions, i.e. the Rainbow digital signing scheme, selected for use on quantum computing systems by the NIST Standards Institute, CSIDH key exchange, and AES-256 symmetric key cryptography to build a secure messaging application. CryptoMess offers a secure communication method, the message data were encrypted throughout the application's communication process, ensuring the data's integrity and authenticity. As a result, the messaging application was able to transport messages between users while protecting their safety, secrecy, and validity. The performance of the program has a transmitting rate of approximately 0.26 s, and receiving speed of approximately 0.22 s. The message authentication speed is typically 0.22 s, and the message

signing time is approximately 0.027 s. The essential exchange time spans approximately 0.0017 s.

ACKNOWLEDGMENT

The authors acknowledge the Academy of Cryptography Techniques and the Minister of Education and Training (MOET) for supporting this work under grant number B2022-GHA-10.

REFERENCES

[1] R. Bhat, N. R. Sunitha, and S. S. Iyengar, "A probabilistic public key encryption switching scheme for secure cloud storage," *International Journal of Information Technology*, vol. 15, no. 2, pp. 675–690, Feb. 2023, https://doi.org/10.1007/s41870-022-01084-8.

[2] U. Iftikhar, K. Asrar, M. Waqas, and S. A. Ali, "Evaluating the Performance Parameters of Cryptographic Algorithms for IOT-based Devices," *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7867–7874, Dec. 2021, https://doi.org/10.48084/etasr.4263.

[3] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," Internet Engineering Task Force, RFC 8446, Dec. 2018, https://doi.org/10.17487/RFC8446.

[4] M. F. Hyder, S. Tooba, and Waseemullah, "Performance Evaluation of RSA-based Secure Cloud Storage Protocol using OpenStack," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7321–7325, Aug. 2021, https://doi.org/10.48084/etasr.4220.

[5] K.-A. Shim, S. Lee, and N. Koo, "Efficient Implementations of Rainbow and UOV using AVX2," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2022, no. 1, pp. 245–269, 2022, https://doi.org/10.46586/tches.v2022.i1.245-269.

[6] V.-H. Le, N.-Q. Luc, T. T. Dao, and Q.-T. Do, "Building an Application that reads Secure Information Stored on the Chip of the Citizen Identity Card in Vietnam," *Engineering, Technology & Applied Science Research*, vol. 13, no. 1, pp. 10100–10107, Feb. 2023, https://doi.org/10.48084/etasr.5531.

[7] G. Alagic *et al.*, "Status report on the third round of the NIST Post-Quantum Cryptography Standardization process," National Institute of Standards and Technology (U.S.), Gaithersburg, MD, USA, NIST IR 8413, Sep. 2022, https://doi.org/10.6028/NIST.IR.8413-upd1.

[8] A. Dalle Zotte, A. Concollato, G. Secci, M. Cullere, and G. Parisi, "Rainbow trout (Oncorhynchus mykiss) farmed at two different temperatures: Post rigor mortis changes in function of the stunning method," *Czech Journal of Animal Science*, vol. 65, no. 9, pp. 354–364, Sep. 2020, https://doi.org/10.17221/144/2020-CJAS.

[9] K. M. Carlson *et al.*, "Global rainbow distribution under current and future climates," *Global Environmental Change*, vol. 77, Nov. 2022, Art. no. 102604, https://doi.org/10.1016/j.gloenvcha.2022.102604.

[10] N. Q. Luc, T. T. Nguyen, D. H. Quach, T. T. Dao, and N. T. Pham, "Building Applications and Developing Digital Signature Devices based on the Falcon Post-Quantum Digital Signature Scheme," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10401–10406, Apr. 2023, https://doi.org/10.48084/etasr.5674.

[11] V. Soukharev and B. Hess, "PQDH: A Quantum-Safe Replacement for Diffie-Hellman based on SIDH." 2019, [Online]. Available: https://eprint.iacr.org/2019/730.

[12] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes, "CSIDH: An Efficient Post-Quantum Commutative Group Action," in *Advances in Cryptology – ASIACRYPT 2018*, 2018, pp. 395–427, https://doi.org/10.1007/978-3-030-03332-3_15.

[13] X. Bonnetain and A. Schrottenloher, "Quantum Security Analysis of CSIDH," in *Advances in Cryptology – EUROCRYPT 2020*, 2020, pp. 493–522, https://doi.org/10.1007/978-3-030-45724-2_17.

[14] I. K. Nti, E. Gymfi, and O. Nyarko, "Implementation of Advanced Encryption Standard Algorithm with Key Length of 256 Bits for Preventing Data Loss in an Organization," *International Journal of Advancements in Technology*, vol. 8, no. 2, 2017, Art. no. 1000183, https://doi.org/10.4172/0976-4860.1000183.

[15] B. Koziel, R. Azarderakhsh, and D. Jao, "On secure implementations of quantum-resistant supersingular isogeny Diffie-Hellman," in *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, Mclean, VA, USA, Feb. 2017, pp. 160–160, https://doi.org/10.1109/HST.2017.7951824.

[16] A. Genêt, N. L. de Guertechin, and N. Kaluđerović, "Full key recovery side-channel attack against ephemeral SIKE on the Cortex-M4." 2021, [Online]. Available: https://eprint.iacr.org/2021/858.

[17] O. Babatunde and O. Al-Debagy, "A Comparative Review Of Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6)," *International Journal of Computer Trends and Technology*, vol. 13, no. 1, 2014, https://doi.org/10.14445/22312803/IJCTT-V13P103.

[18] J. Schwenk, "IP Security (IPSec)," in *Sicherheit und Kryptographie im Internet: Von sicherer E-Mail bis zu IP-Verschlüsselung*, J. Schwenk, Ed. Wiesbaden, Germany: Vieweg+Teubner Verlag, 2005, pp. 118–151, https://doi.org/10.1007/978-3-322-95321-6_5.

[19] J. Voas and I. Bojanova, "NIST: Building a Solid Foundation," *IT Professional*, vol. 16, no. 2, pp. 13–16, Nov. 2014, https://doi.org/10.1109/MITP.2014.21.

[20] D. Moody *et al.*, "Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process," NIST, NIST Pubs 8309, Jul. 2020, https://doi.org/10.6028/NIST.IR.8309.

[21] W. Castryck and T. Decru, "An efficient key recovery attack on SIDH." 2022, [Online]. Available: https://eprint.iacr.org/2022/975.

[22] R. Oudompheng and G. Pope, "A Note on Reimplementing the Castryck-Decru Attack and Lessons Learned for SageMath." 2022, [Online]. Available: https://eprint.iacr.org/2022/1283.

[23] J. Ding and D. Schmidt, "Rainbow, a New Multivariable Polynomial Signature Scheme," in *Applied Cryptography and Network Security*, Berlin, Heidelberg, 2005, pp. 164–175, https://doi.org/10.1007/11496137_12.

[24] N. Drucker and S. Gueron, "Speed Up Over the Rainbow," in *ITNG 2021 18th International Conference on Information Technology-New Generations*, 2021, pp. 131–136, https://doi.org/10.1007/978-3-030-70416-2_17.

[25] C. Paquin, D. Stebila, and G. Tamvada, "Benchmarking Post-Quantum Cryptography in TLS." 2019, [Online]. Available: https://eprint.iacr.org/2019/1447.