Securing Digital Communication in the Post-Quantum Era

Prepared For
Allyson Conard

Prepared by
Bowen Dai
November 1, 2024

Executive summary

The development of quantum computing is a thread of future online communication privacy. The main cryptographies that society currently relies on are based on the difficulty of factorizing large numbers and finding discrete logarithmic. However, the algorithm to factor large numbers and to find discrete logarithms using quantum computers had already been published in 1999 [1]. Furthermore, some research institutions already built a prototype of a quantum computer in their laboratory [2], [3]. The advancement in quantum computing makes current cryptography extremely vulnerable since a practical model of quantum computers may be created by any research team at any time. Luckily, there are already many studies on quantum-proof encryption algorithms. The most well-known quantum-resistant algorithms are Learning with Error, N-th degree Truncated polynomial Unit Ring, McEliece and Advanced Encryption Standard. Each of them has different advantages and inconveniences in the implementation and calculation. To ensure the most efficient and secure communication, the hybrid cryptography of Kyber and AES is the best combination because they are both secure under quantum and regular attacks and offer flexibility which allows different users to personalize the strength level and performance according to their need.

Introduction

Quantum-computing attracted many brilliant researchers to work on it since it had been proposed by Benioff in 1980 for the first time [4]. In 1998, Gershenfeld and Chuang at MIT developed a 2-qubit computer based on Nuclear Magnetic Resonance (NMR) [2]. Only three years after the 2-qubit quantum computer, IBM successfully developed an NMR quantum computer with 7-qubit [3]. Qubits are the basic storage unit for quantum computers just like bits are the basic storage unit for regular computers. The difference between a bit and a qubit is that a bit can only store 1 or 0, but a qubit can store 1 and 0 at the same time [5 Ch 4]. This means that a quantum computer can perform multiple similar calculations simultaneously. This increases tremendously the speed to solve many difficult problems.

It sounds like great news for everyone, but the problems that the quantum computer can solve way faster than classical computers include prime factorization and discrete logarithms [1]. The Shor algorithm which allows quantum computers to factorize any given number with polynomial-time had already been implemented by IBM in their 7-qubit quantum computer [3]. Before the apparition of quantum computers, these two problems were considered impossible to solve within a reasonable time constraint even for the fastest classical computer. Therefore, the currently most used cryptosystems, Rivest-Shamir-Adleman (RSA), Elliptical Curve Cryptography (ECC) and Diffie-Hellman key exchange, all rely on the difficulty of factorization or discrete logarithmic [6 Ch 4], [7], [8 Ch 12]. These two encryption systems allow people to securely communicate online without worrying about privacy breaches. In the age of information, millions of transactions happen daily through the internet. It is vital to keep all sensitive information free from any malicious attacks. Otherwise, any PIN for bank accounts, passwords for email and online transaction information can be easily seen and modified by anyone. This hypothetical scenario will become the reality if any research institute successfully implements a quantum computer with enough qubits to factorize the currently used numbers in RSA and ECC.

To prevent this catastrophic situation, it is crucial to develop a suitable cryptosystem which is quantum-resistant, secure, and efficient. More importantly, the new encryption system should be compatible with the current cryptosystem to facilitate the transition. Also, it should be suitable for various devices with different computing power. The prevalence of the Internet of Things could be a potential vulnerability of encryption systems because some devices with fewer computing resources can also be a part of communication [9].

Luckily, serval cryptosystems had already been proven as quantum resistant such as Leaning with Error, Nth-degree Truncated polynomial Ring Unit (NTRU), McEliece and Advanced Encryption Standard (AES). This report includes an analysis of cryptosystem from each of these encryption system on their security and efficiency. LWE is a secure asymmetric cryptography. It is less efficient than symmetric cryptography, but it does not require a secure channel to exchange the key. NTRU is more efficient than LWE. However, a recent study shows that NTRU requires more improvement in the hardware implementation to resist side-channel attacks. McEliece is secure, but it is slightly less resistant to attack compared to other quantum-proof algorithms. AES is the most efficient encryption scheme, but since it is symmetric cryptography, it requires a secure channel to exchange the key. To maximize efficiency and security, the most suggested encryption system is a hybrid cryptography of Kyber which is a specific implementation of LWE and AES. This hybrid model takes advantage of the efficiency of AES and Kyber as an algorithm to encrypt and exchange the key for AES. Therefore, this scheme does not have the key exchange problem and offers secure and efficient communication.

Discussions

Before diving into the analysis of a specific cryptosystem, it is helpful to know the general classification of cryptography to get a general sense of the efficiency of each cryptography that is included in this report.

Symmetric cryptography

In symmetric cryptography, encryption and decryption are the reverse of each other. The key used in both processes is identical. Symmetric cryptography is simple to implement and fast both in encryption and decryption processes. Therefore, this category of cryptography is efficient and compatible with various devices which have weaker computation power. Furthermore, the key for symmetric cryptography is usually shorter than asymmetric cryptography which reduces the space required to implement the algorithm.

The most classical example of symmetric cryptography is Caesar Cipher in which the sender encrypts the message by replacing each letter with another one located a little further in the alphabet. The distance between the original letter and the replaced letter is the offset. Encryption and decryption use the same key: the offset of the alphabet and the direction of the offset. If the offset is 1 and the direction of the offset is forward, then A will be replaced by B, B by C, etc. The recipient needs to know the offset and its direction to decrypt the ciphertext. The symmetric cryptography that this report analyzes is the Advanced Encryption Standard (AES).

As the example of Caesar cipher illustrated, both parties in the communication need to agree on a common key. This requires a secure channel between both parties to share the key. However, if a secure channel exists, it is unnecessary to encrypt the message. The key exchange problem in symmetric cryptography is one of the main motivations for the use of asymmetric cryptography.

Asymmetric cryptography

Asymmetric cryptography solves the key exchange problem elegantly. Encryption and decryption in asymmetric cryptography are not the reverse of each other. Therefore, the sender and the receiver do not rely on any common key to communicate securely. Asymmetric cryptography requires the recipient of the message to generate both private and public keys. The receiver keeps their private key safely and shares the public key with the sender through any public channel without the worry about the threat of eavesdropping because the public key can only be used to encrypt data. It is useless for eavesdroppers who want to know the contents of encrypted messages. The asymmetry cryptography that the report analyzes in detail are Learn with Error (LWE), Nth-degree Truncated polynomial Ring Units (NTRU) and McEliece.

The security of asymmetric cryptography relies on some trapdoor function which is easy to compute in one direction but extremely hard to reverse. For example, the trapdoor function that RSA relies on is the mapping between prime factors and their product: given 2 factors, it is easy to compute their product, but given the product, it is extremely hard to calculate its factors.

Although asymmetric cryptography avoids the key exchange issue, it is generally more computationally intensive than symmetric cryptography. Moreover, the key size of asymmetric cryptography is longer than the key size of symmetric cryptography. This may affect the efficiency of the algorithm.

Hybrid cryptography

Despite that both symmetric cryptography and asymmetric cryptography have critical flaws in the implementation or efficiency of the algorithm, it is possible to combine the advantages of each cryptography to create a secure and efficient encryption system.

The optimal combination of symmetric and asymmetric cryptography is called hybrid cryptography. It uses asymmetric encryption to distribute safely the key needed for symmetric cryptography. After the key is successfully shared between both parties, they can use a symmetric cryptosystem to communicate between them more efficiently. Hybrid cryptography guarantees both security and efficiency of communication because the only part encrypted using asymmetric cryptography is the key to symmetric cryptography which is usually shorter than the message to encrypt.

Before the exchange of information, both parties should generate a pair of private and public keys and share the public key. Then in each exchange, the sender should encrypt twice the key of symmetric cryptography using their private key and the public key of the receiver. Next using symmetric cryptography, the sender encrypts the main message and sends the double encrypted key and cipher text to the receiver. The receiver could decrypt the double encrypted key using their private key and the public key of the sender. Thus, they can also decrypt the ciphertext using the decrypted key [10].

This hybrid model is not only an optimal combination of both types of cryptography but also ensures the authentication of the message. Unlike asymmetric cryptography in which anyone can send a message to the receiver because the public key is public to all, this hybrid cryptography only allows legitimate users to communicate between them because the key of symmetric encryption is encrypted twice using the key of both parties. Hybrid cryptography is efficient and secure. It also ensures the authenticity of the message. It is also the encryption scheme for most digital communication nowadays. To choose the best alternative encryption algorithm for this hybrid model, it is important to have a good understanding of the possible choices.

Learn with error

Mechanism of the cryptosystem

Key generation

To set up the public and private key for LWE, the receiver first generates a system of linear equations in the following form [11]:

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = c_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = c_2 \\ \vdots \end{gathered}$$

where a_{ij} is the coefficient in front of each unknown x_i in the i-th equation and c_i is the constant part of the solution

The private key is the solution to this system of equations which is the value for each variable, so all the equations are satisfied. Then the receiver adds a randomly generated small error to the constant part of each solution and chooses a large number m to take the modulo for the constant of each equation. The operation of modulo m transforms c_i to r_i such that r_i is the remainder of $\frac{c_i}{m}$. After the operation, the overdetermined system of equations is extremely hard to solve for an attacker because of the error in each equation.

The private key in this cryptosystem is the solution of the equation system and the public key is the system of equations with error.

Encryption

To encrypt a message, the sender must first transform its message into a list of 0 and 1. Then the sender randomly combines serval equations in the public key. Then add the m/2 if the sender wishes to encrypt 1 or add nothing if the sender wishes to encrypt 0. The encrypted message will be the sequence of equations modified by the sender. Since the combination of the equations grows exponentially with the number of equations, it is extremely time-consuming for an eavesdropper to check if each equation is modified by the sender or not.

Decryption

To decrypt the ciphertext, the recipient can use their private key to calculate the value of the equation from the sender. Since combining the equations does not affect the value of the solution, no matter how the sender chooses to combine the equations, the private key is also a solution of the equation. By comparing the value calculated using the private key and the value sent by the sender, it is easy to know if the sender encrypted a 1 or a 0. If the difference between the calculated value and the value from the sender is very small, the sender did not modify the result of this equation. Therefore, this equation encrypts 0. Otherwise, if the difference between the two values is big compared to m, the sender modified the equation which means that the equation is an encrypted 1 [11].

Cryptographic Suite for Algebraic Lattice-Kyber

Based on the encryption and decryption system of LWE, the algorithm Cryptographic Suite for Algebraic Lattice-Kyber (Kyber) has been created. This encryption system has been submitted to the National Institute of Standards and Technology (NIST) post-quantum cryptography (PQC) project and has been selected by NIST as the algorithm to be standardized in 2022 [12]. Kyber has different parameter sets aimed at different security levels. Specifically, Kyber-512 aims the security equivalent to AES-128, Kyber-768 aims to AES-192 and Kyber-1024 aims AES-256[13]. The recommended parameter set by the official site of Kyber is Kyber-768 since it achieves more than 128 bits of security against all known classical and quantum attacks [13]. Furthermore, Kyber is already in use by industries such as Cloudflare, Amazon and IBM [14]-[16]. In both theory and practice, Kyber demonstrates high performance and security. In addition, Kyber is also very suitable for the hybrid encryption model since the official site recommends users implement Kyber with established pre-quantum security [13]. In addition, Kyber offers different parameter sets. Thus, users can adjust conveniently the encryption to find the balance between performance and strength according to their needs.

Nth-degree Truncated polynomial Ring Unit (NTRU)

Mechanism of the cryptosystem

Key generation

To generate the pair of private and public keys, the receiver should first choose two coprime numbers p and q as well as two polynomials with the degree at most N-1 and coefficients of –1, 0 or 1 *f* and *g* [17]. The extra requirement about f is that there must exist f_p and f_q which are two other polynomials such that $f \cdot f_p = 1 \ (mod\ p)$ and $f \cdot f_q = 1 \ (mod\ q)$. The private key in NTRU contains the polynomial *f*, f_p and *g*, and the public key is the product $h = p \cdot f_p \cdot g \ (mod\ q)$.

Encryption

To encrypt the message, the sender needs to first transform the message into a polynomial. For example, first, transform the message into a sequence of numbers then use these numbers as the coefficients of the polynomial. Then the sender should generate a random polynomial r to hide their plaintext. The encrypted message e can be obtained by the following formula: $e = r\ h + m \ (mod\ q)$.

Decryption

To recover the message, the recipient should cancel the term *r h* in the encrypted message using their private key and modular arithmetic. First, they multiply the encrypted message by the polynomial f in their private key, then take the modulo *q*:

$$\begin{aligned} & fe && \\ &= f(rh + m) && \text{since } e = r \cdot h + m \\ &= f\left(rpf_q g + m\ \right) && \text{since } h = p \cdot f_q \cdot g \\ &= frpf_q \cdot g + f \cdot m && \text{distribute } f \\ &= rpg + f \cdot m && \text{since } f \cdot f_q = 1 \end{aligned}$$

Next, take modulo *p*

$$\begin{aligned} & rpg + f \cdot m && \\ &= fm \qquad (mod\ p) && \text{since } rpg = 0\ (mod\ p) \\ & f_p fm \quad\ (mod\ p) && \text{multiply by } f_p \\ &=\ m \qquad\ (mod\ p) && \text{since } f_p f\ = 1\ (mod\ p) \end{aligned}$$

After these operations, the recipient can successfully recover the message m [17].

Security of NTRU

NTRU is selected as a finalist of round 3 in the PCQ project as well due to its efficiency of the calculation using the polynomial ring [18], [19]. However, in 2023, a recent study suggests that NTRU has a fatal vulnerability under non-profiled side channel attack (SCA). Even considering the countermeasures provided by NTRU and the weak attacker model in which attackers have only limited access to data, computational resources and knowledge of the internal system, SCA can recover the key of NTRU using electromagnetic analysis (EM) of the hardware [19]. Thus, this algorithm needs more improvement in terms of hardware implementation to be practical.

McEliece

Mechanism of the cryptosystem

Key generation

To know more about the key generation of McEliece, it is important to introduce a few concepts in coding theory [21]. The minimum distance of a linear code *C* is the smallest Hamming weight which is the number of nonzero digits in the code of a nonzero code word in *C* which is equivalent to the minimum distance between two distinct codewords.

$$d = \min_{c\neq 0}\{wt(c)\} = \min_{c\neq b}\{dist(b,c)\}$$

The minimum distance determines the maximum number of errors which can be corrected. If *d = 2t + 1*, then any vector with less than t errors can be corrected. Intuitively, if the distance between the code with less than t error and the correct code is shorter than half of the minimum distance between distinct codes, then the error correcting algorithm always corrects the code with errors to the closest correct code.

Let *G* be a generator matrix for Goppa code *Γ* of dimension k and length *n* with the minimum distance *2t+1*. Then the private key is the generator matrix *G* and an efficient *t*-error correcting decoding algorithm for *Γ*, a *n* by *n* permutation matrix *P* and a *k* by *k* non-singular matrix *S*. Permutation matrix is a matrix with only a single 1 in each column and row, and all other coordinates are 0. This kind of matrix swaps the order of elements of another matrix when they are multiplied together. The nonsingular matrix is a square matrix with a non-zero determinant. Such a matrix is invertible and maintains the full information.

The public key G' is the product of the permutation matrix *P*, the generator matrix *G* and the non-singular matrix *S*. Also, all the dimensions of the matrix and the maximum number of errors, *n*, *k* and *t* are public.

Encryption

To encrypt the message, first transform the message to a sequence of number lists with length *k*. Let's use m to denote a sequence of the message number, *e* to denote a randomly generated error vector with weight at most t, and the encrypted message y is $mG' + e$.

Decryption

Since multiplying P permutes the order of the elements, it is possible to reverse it. Therefore, knowing P, it is possible to calculate the inverse of P, P^{-1} which is also a permutation matrix and cancels the effect of P. Then multiply P^{-1} to the encrypted message y:

$$\begin{aligned} yP^{-1} &= (mG' + e)P^{-1} \\ &= m\,G'P^{-1} + eP^{-1} \\ &= mSGPP^{-1} + eP^{-1} \\ &= mSG + eP^{-1} \end{aligned}$$

Since P^{-1} is also a permutation matrix, eP^{-1} is still an error with weight at most t and it can be corrected by the error correcting algorithm. As the recipient knows the generator matrix G, they can use this information to efficiently decode the message and then use the inverse of S to recover the message [21].

Classic McEliece

A detailed implementation of McEliece cryptography is Classic McEliece. It was also a finalist of NIST's PQC project thanks for its outstanding efficiency. Furthermore, Classic McEliece is IND-CCA2 secure against all random oracle model (ROM) attacks whether they are regular ROM or quantum ROM [22]. This means that Classic McEliece is indistinguishable under adaptive chosen ciphertext attacks, so attackers cannot distinguish pairs of cyphertexts based on their adaptively chosen ciphertext. Classic McEliece also performs better in permutation security, and compression of secret keys and it is more secure under chosen-cipher attacks compared to other cryptographies based on similar mathematic problems such as Task Force [23]. The security of the McEliece algorithm has always extremely stable since 1978 [22]. However, in comparison with other finalists of the PQC project, the W-cost and D-cost of Classic McEliece are generally slightly lower than other algorithms [24]. This means the work cost and data cost in attacks against Classic McEliece are slightly lower than other algorithms.

Advanced Encryption Standard (AES)

Mechanism of the cryptosystem

AES is a symmetric block cipher which means that the encryption and decryption process is applied to a block of bits instead of one single bit [25]. More specifically, AES encrypts or decrypts 128 bits each time. The main encryption method of AES is based on substitution and permutation. It processes each block of data multiple times with a similar encryption process. Since AES is a symmetric cryptography, the encryption and decryption processes are very similar. The original key of AES may have different lengths such as 128 bits, 192 bits or 256 bits. The length of the original key determines the number of encryption rounds. A longer original key means more encryption rounds per block of data: AES-128 uses 10 rounds to encrypt each block, AES-192 uses 12 rounds and AES 256 uses 14 rounds.

Before entering the encryption rounds, the plaintext has first been arranged into a 4 by 4 bytes matrix, then goes through the process of adding a round key with the initial round key generated by the original key using the key expansion algorithm. Besides the last round, each round of encryption, each round of encryption consists of four steps: sub bytes, shift rows, mix columns and add round key which will be explained in detail in the next section. Except the mix column step is missing, the last encryption round is the same as a regular encryption round. In each round, a round key is generated from the original key. As AES is a symmetric cryptography, the decryption process is the same as the encryption process except each step will be the inverse of the encryption.

In any add key step, the encrypted data will become the result of the input data and the round key under the operation XOR. In sub bytes step, each byte in the matrix will be substituted by another according to their position and S-Box. To shift the rows, each row will be rotated to a certain number of positions. The first row is intact, the second row will be shifted by 1 byte to the left and the third row will be shifted by 2 bytes, fourth row by 3 bytes. Last, the mix columns is a multiplication between the input matrix and the following matrix [25]:

$$M = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Security

The algorithm of AES is originally proposed by Daemen and Rijmen. This algorithm was the winner of the competition of NIST to find an alternative to DES in 2000 and has been standardized around the globe [26]. This encryption algorithm is the most suitable symmetric encryption for post-quantum cryptography because it is compatible with the current system and has a low requirement for memory and computation. Since it has been the standard for encryption for more than twenty years, the compatibility across different hardware, software and systems with AES is better than any other encryption. Furthermore, this encryption system offers different key sizes so users can balance the performance and strength based on their needs [25]. Most importantly, AES ensures both security and performance. Using only a single-core CPU, AES can encrypt around 3GB of data per second [27]. The best attack known for AES is biclique cryptanalysis [28]. However, this attack does not threat the strength of AES because even with a biclique attack, the time complexity is still $2^{126.13}$ for the least strong AES, AES-128.

Conclusion & Recommendations

Since the development of quantum computers keeps advancing, society should be prepared for potential quantum attacks on the currently used cryptosystem. To prevent the threat of quantum attack, finding an efficient, secure alternative encryption system is important for anyone using online transaction and communication services. According to the analysis of the most well-known quantum-proof cryptography, the system with the best performance and ensuring the confidentiality and authenticity of the communication is the hybrid cryptography of Kyber and AES. In addition, thanks to the current use of a similar encryption scheme, the transition to the new cryptosystem is easier and faster to implement than any other cryptosystem.

References

[1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999, doi: 10.1137/s0036144598347011.

[2] N. A. Gershenfeld and I. L. Chuang, "Bulk Spin-Resonance Quantum Computation," *Science (American Association for the Advancement of Science)*, vol. 275, no. 5298, pp. 350–356, 1997, doi: 10.1126/science.275.5298.350.

[3] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature (London)*, vol. 414, no. 6866, pp. 883–887, 2001, doi: 10.1038/414883a.

[4] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," *Journal of statistical physics*, vol. 22, no. 5, pp. 563–591, 1980, doi: 10.1007/BF01011339.

[5] S. Akama, *Elements of Quantum Computing : History, Theories and Engineering Applications*, 1st ed. 2015. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-08284-4.

[6] R. Banoth and R. Regar, *Classical and Modern Cryptography for Beginners*, 1st ed. 2023. Cham: Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-32959-3.

[7] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987, doi: 10.1090/s0025-5718-1987-0866109-5.

[8] W. Diffie and M. E. Hellman, "New Directions in Cryptography," in *Democratizing Cryptography*, New York, NY, USA: ACM, 2022, pp. 365–390. doi: 10.1145/3549993.3550007.

[9] P. Anand, Y. Singh, A. Selwal, P. K. Singh, and K. Z. Ghafoor, "IVQFIoT: An intelligent vulnerability quantification framework for scoring internet of things vulnerabilities," *Expert systems*, vol. 39, no. 5, 2022, doi: 10.1111/exsy.12829.

[10] S. Bhat and V. Kapoor, "Secure and Efficient Data Privacy, Authentication and Integrity Schemes Using Hybrid Cryptography," in *Advances in Intelligent Systems and Computing*, vol. 870, Singapore: Springer, 2018, pp. 279–285. doi: 10.1007/978-981-13-2673-8_30.

[11] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, pp. 1–40, 2009, doi: 10.1145/1568318.1568324.

[12] I. T. L. Computer Security Division, "Selected Algorithms 2022 - Post-Quantum Cryptography | CSRC | CSRC," *CSRC | NIST*, Jan. 03, 2017. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[13] P. Schwabe, "CRYSTALS," *pq-crystals.org*. https://pq-crystals.org/

[14] "Securing the post-quantum world," *The Cloudflare Blog*, Dec. 11, 2020. https://blog.cloudflare.com/securing-the-post-quantum-world/

[15] "Round 2 post-quantum TLS is now supported in AWS KMS," *Amazon Web Services*, Nov. 16, 2020. https://aws.amazon.com/blogs/security/round-2-post-quantum-tls-is-now-supported-in-aws-kms/

[16] "World's first quantum computing safe tape drive," *IBM Research Blog*, Feb. 09, 2021. https://research.ibm.com/blog/crystals-quantum-safe

[17] A. Kamal, K. Ahmad, R. Hassan, and K. Khalim, “NTRU Algorithm: Nth Degree Truncated Polynomial Ring Units,” in *Functional Encryption*, Switzerland: Springer International Publishing AG, 2021, pp. 103–115. doi: 10.1007/978-3-030-60890-3_6.

[18] I. T. L. Computer Security Division, “Round 3 Submissions - Post-Quantum Cryptography | CSRC | CSRC,” *CSRC | NIST*, Jan. 03, 2017. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions

[19]L. Bettale, J. Eynard, S. Montoya, G. Renault, and R. Strullu, “Security Assessment of NTRU Against Non-Profiled SCA,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13820, Cham: Springer International Publishing, 2023, pp. 248–268. doi: 10.1007/978-3-031-25319-5_13.

[21] T. Lange, “Selected Areas in Cryptology,” *Hyperelliptic.org*, 2021. https://hyperelliptic.org/tanja/teaching/pqcrypto21/ (accessed Nov. 02, 2024).

[22] “Classic McEliece: Intro,” *Mceliece.org*, 2019. https://classic.mceliece.org/index.html

[23] “Classic McEliece vs. NTS-KEM,” 2018. Available: https://classic.mceliece.org/nist/vsntskem-20180629.pdf

[24] Y. Li and L. P. Wang, “Security analysis of the Classic McEliece, HQC and BIKE schemes in low memory,” *Journal of information security and applications*, vol. 79, pp. 103651-, 2023, doi: 10.1016/j.jisa.2023.103651.

[25] J. Daemen and V. Rijmen, “AES Proposal: Rijndael.” Available: https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf#page=1

[26] I. T. L. Computer Security Division, “Announcing Approval of FIPS 197 AES | CSRC,” *CSRC | NIST*, Dec. 18, 2016. https://csrc.nist.gov/news/2001/announcing-approval-of-fips-197-aes

[27] “AES-NI SSL Performance Study @ Calomel.org,” *calomel.org*. https://calomel.org/aesni_ssl_performance.html

[28] B. Tao and H. Wu, “Improving the Biclique Cryptanalysis of AES,” *Information Security and Privacy*, pp. 39–56, 2015, doi: https://doi.org/10.1007/978-3-319-19962-7_3.