Post-quantum cryptography based on RISC V Classic McEliece optimization design

1023041101 Xiaoyu Hu

***Abstract*—With the advent of the Internet of everything era, people's material life has been greatly improved. However, in reality, with the development of science and technology, there are many information security problems, and McEliece algorithm, as a quantum secure encryption and decryption algorithm, can provide strong security. Risc-v is a free and open instruction set architecture based on reduced instruction set principle, and is the 5th generation RISC, which has a wide application prospect in the field of Internet of Things and embedded market. By implementing McEliece algorithm on RISC-V architecture, it can not only guarantee the security of data, but also promote the wide application of the algorithm in hardware, further promote the integration of cryptography and computer architecture, and provide a solid foundation for future quantum secure communication. The main objective of this paper is to optimize Mceliece algorithm on RISC-V architecture. In order to achieve this goal, we first take advantage of the unique characteristics of the RISC-V instruction set by using some efficient instructions to complete the key computational parts of the Mceliece algorithm, including the M-bit finite field multiplication, reduction and square operations. Further, we focus on optimizing the polynomial multiplication part of Mceliece algorithm. In order to achieve better performance, we have tried many different polynomial algorithms and tested them. Our goal is to find a more efficient polynomial multiplication algorithm to replace the original schoolbook polynomial multiplication. Through this optimization measure, we once again improve the performance of the algorithm.**

***Index Terms*—RISC-V Meceliece**

I. Introduction

RISC-V is an instruction set architecture based on open principles, and its design is concise, clear, modular, and extensible and flexible. With the rapid development and wide application of RISC-V in various fields, the application of classical McEliece[1] algorithm in RISC-V instruction set architecture can not only maintain the security of the algorithm, but also greatly improve the execution efficiency and performance at the hardware level.

First, by implementing the McEliece algorithm on the RISC-V instruction set, we can take full advantage of the advantages of the RISC-V architecture, such as a streamlined and standardized instruction set and pipelined processing[2] . The simplicity of the RISC-V instruction set allows the processor to execute instructions more efficiently, while normativity ensures software portability. With the advantages of RISC-V, the encryption and decryption operation of McEliece algorithm can be greatly accelerated and the overall performance can

Identify applicable funding agency here. If none, delete this.

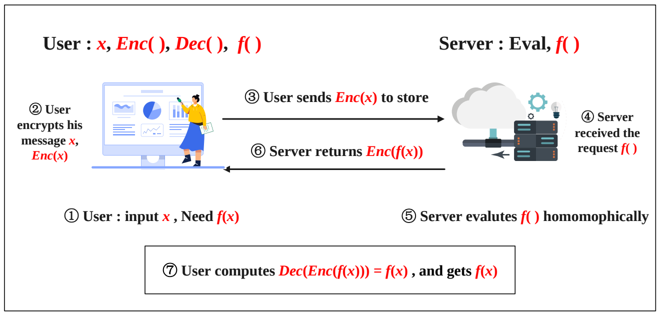
User : x, Enc( ), Dec( ), f( )
Server : Eval, f( )
② User encrypts his message x, Enc(x)
③ User sends Enc(x) to store
⑥ Server returns Enc(f(x))
④ Server received the request f( )
① User : input x , Need f(x)
⑤ Server evalutes f( ) homomophically
⑦ User computes Dec(Enc(f(x))) = f(x) , and gets f(x)

Fig. 1: Cryptography

be improved. Secondly, the implementation and optimization of McEliece algorithm[3] is of great significance for data security. In today's information age, the confidentiality and integrity of data is of Paramount importance to individuals and organizations. McEliece algorithm, as a cryptographic algorithm against quantum computing attacks, can provide strong security. The implementation of RISC-V architecture can not only ensure the security of data, but also provide efficient and reliable encryption and decryption services for users. In addition, the implementation of McEliece algorithm on RISC-V architecture will also promote the research and development in the field of cryptography. By combining McEliece algorithm with RISC-V instruction set, the algorithm can be widely used in hardware and further promote the integration of cryptography and computer architecture. This will provide more practical opportunities and optimization space for cryptography researchers and engineers, and promote the further development of cryptography technology. In particular, the optimization of the classical McEliece algorithm needs to consider several aspects. First of all, the mathematical principle of the algorithm needs to be deeply studied and optimized according to the characteristics of RISC-V architecture. For example, with the vector instruction set extension of RISC-V, multiple operations can be performed in parallel, thus improving the parallel processing capability of the algorithm. Secondly, the key steps of the algorithm need to be optimized. Through targeted optimization, the computational complexity and storage requirements can be reduced, and the execution efficiency of the algorithm can be further improved. In practical applications, the optimized McEliece algorithm based on RISC-V architecture can play an important role in many fields. For example, in cloud and edge computing environments, the privacy and security of user data can be protected by applying the optimized McEliece algorithm to the data encryption and decryption process. In addition, in the Internet of Things and intelligent transportation systems, the application of this algorithm to data protection during communication can ensure that the transmitted data is not stolen or tampered with.

A. Related Works

As an open and extensible instruction set architecture with cross-platform compatibility, RISC-V instruction set architecture has been studied since its birth, and its characteristics and improvement strategies have been explored. Tang Junlong [3] referred to the ARM architecture, modified the GCC source code [4] to get the target machine description file, transplanted GCC to the RISC-V processor platform, and correctly generated the riscv-none-embed-gcc compiler. riscv-none-embed-gcc is optimized by using a strength-weakened peehole optimization method to solve the problem of high CPU calculation cost caused by inefficient use of registers in the process of intermediate code generation, and reduce the volume of target code. Wu Xinlong [5] mainly optimized by reducing program volume. RISC-V instruction set subextension Zce developed a series of instructions, among which a series of instructions represented by LWGP were used to reduce the number of instructions when loading/storing byte data. From this Angle, Wu Xinlong analyzed the optimization principle of the instruction represented by LWGP to the code volume and implemented it on the LLD linker. Miao Ruixia [6] In order to realize the research and optimization of convolutional algorithms of neural networks, The low precision fixed-point quantization method for embedded platform is adopted to reduce the bit width of data while maintaining the accuracy, thus reducing the memory limitation of the neural network model deployed on embedded devices. RISC-V instruction architecture is adopted to complete the underlying implementation of the convolutional neural network model to shorten the execution time of the operation. Zhang Xu [7] proposed a hardware-accelerated co-processor design solution of AES algorithm based on RISC-V, aiming at the problems existing in the scalability and performance of AES algorithm on the Internet of Things and embedded devices, which realized the call of several custom extended instructions in software programs by means of inline assembly functions. On this basis, the circular structure of AES algorithm is implemented.

B. Contributions and Paper Organization

This design aims to complete the optimization of the classical Mceliece algorithm on RISC-V. This paper consists of six chapters, and the structure of the paper is as follows:

- Chapter 1 introduces the research background and significance of classical Mceliece algorithm optimization on RISC-V, and explains the correlation analysis and research status at home and abroad, and introduces the structure of the paper.Chapter 2 introduces the relevant preparatory knowledge, including error correcting code, RISC-V instruction set and VisionFive2, and the knowledge related to the subject in number theory.
- Chapter 3 introduces the classic Mceliece encryption and decryption algorithm, explains its concept and its key generation, encryption and decryption algorithm specific flow.In Chapter 4, the optimization implementation of M-bit finite domain computation on RISC-V instruction set is introduced, including the optimization implementation of multiplication, reduction, square, etc.
- Chapter 5 introduces the optimization of polynomial multiplication, explains the schoolbook multiplication, and a variety of karatsuba polynomial multiplication.In Chapter 6,The optimization effect is tested, and the work done is summarized and prospected.

II. Preliminaries

Table I displays the symbols used along with their detailed explanations.

A. Error-correcting code

TABLE I: Symbol Explanation

Symbol	Explanation
Z_n	The residue ring modulo n
Z_n^*	The unit group Z_n
$(mod\ n)$	Modulo n, the $len(n)$ = 2048
$parameters$	The public parameters necessary for encryption, decryption or the group operation on ciphertexts
m	The plaintext to be encrypted, a message $m \in Z_n$
c	The encrypted ciphertext c
w	The window size of division of large integers algorithm
d	The scalars in homomorphic scalar multiplication algorithm
x_i, y_i, z_i	The i-th significant bit of big integer x, y, z
l	The number of integers for n, and $l = \frac{n}{32}$
$x[i:j]$	The integers of h indexed from i to j
len$x$	The bit length of number x

Error correction code is a coding technique used to detect and correct errors during data transmission and storage. It implements error detection and correction by adding redundant information to the data. Error-correcting codes are widely used in communication, storage and computing to ensure the reliability and integrity of data. During data transmission, the signal may be affected by interference, noise, or transmission errors, resulting in data corruption or errors. Error correction codes allow the receiver to detect and correct errors by introducing redundant bits into the raw data. It is based on specific coding rules and algorithms to realize error detection and correction by encoding and decoding data. The key idea of error-correcting code is to encode the original data by coding algorithm at the sending end and send the encoded data to the receiving end together. After receiving the data, the receiver uses the decoding algorithm to decode and correct the received data. If there is an error in the data, the error correction code can detect the error location and correct it, restoring the original data. According to different standards [14], error correction codes can be divided into the following categories: 1. According to the relationship between the information element and the check element: linear code and nonlinear code. 2. Based on the error element correction type: random error correcting code, burst error correcting code, random error correcting code, and burst error correcting code. 3. According to the different processing methods of information: convolutional code and block code. 4. According to the relationship between code words: cyclic code and non-cyclic code.

Algorithm 1 Encryption()

```
Input:
    a message $m \in Z_n$ , a public key $pub.key = n$, and
    parameters $parameters = n$.
Output:
    a ciphertext $c$.
1: Choose $r$ uniformly at random from $Z_n^*$.
2: Compute $c = (nm + 1)r^n(mod\ n^2)$.
3: Output $c$.
```

B. RISC-V instruction set and VisionFive2

RISC-V Instruction set Architecture (ISA) was proposed by Andrew Waterman and others at the University of California, Berkeley in 2010, which is characterized by a small variety of CPU instructions and simple functions, compared to the mature Intel X86 architecture,RISC-V architecture's biggest advantage is that the streamlined instruction set makes it suitable for high-speed running scenarios. RISC-V is a high-quality, license-free, open instruction set architecture that is maintained and promoted by the RISC-V Foundation. Its design goal is to achieve open source code and open instruction set. Chips designed using RISC-V architecture have lower areal power consumption, a more concise structure, and a better balance between areal power consumption and performance, making them more cost-effective in practical applications. Today, RISC-V is widely used in many fields, including Internet of Things devices, desktop computers and high-performance computers. Its application scope continues to expand, has been widely recognized by the industry. At the same time, RISC-V, as a member of RISC, will inevitably have some drawbacks of reduced instruction set, such as the size of the binary program. As an architecture designed for educational purposes, simplicity is an essential characteristic of the RISC-V architecture. RISC-V architecture is designed in a combination of basic instruction set and extended instruction set, and fully adopts the idea of modularity. The basic instruction set is the core of the RISC-V processor and defines the most basic integer instruction set that microarchitectures must implement. Using the 32-bit architecture as an example, you only need to implement the RV32I integer instruction set, which contains 47 instructions. By implementing this basic instruction set, the compiler can simulate almost all RISC-V processor functions. In addition to the basic instruction set, the RISC-V architecture defines a series of optional extended instruction sets. In practical applications, extended instruction sets can be freely selected and combined according to requirements to build RISC-V instruction sets suitable for specific application requirements. This flexibility allows the RISC-V architecture to adapt to different application scenarios and provide more highly customized solutions.

C. CUDA and PTX ISA

In 2006, NVIDIA released CUDA (Compute Unified Device Architecture) to facilitate programming on GPUs for researchers [4]. PTX (Parallel Thread Execution) is an efficient parallel instruction set in NVIDIA GPU architecture, possessing numerous advantages. Firstly, PTX instruction set is a universal set applicable to NVIDIA GPUs. Secondly, these instructions are designed for parallelism, enabling multiple threads to execute the same instructions concurrently on the GPU. In summary, PTX instruction set offers developers a flexible and efficient way to harness the parallel computing capabilities of NVIDIA GPUs. By writing and optimizing PTX code, developers can accelerate various complex computational tasks on the GPU, thereby enhancing program performance.

TABLE II: PTX ISA Instructions

PTX Instructions	Meanings
$s = add(c).(cc)(a, b)$ [1]	$s = a + b$
$s = sub(c).(cc)(a, b)$ [2]	$s = a - b$
$s = mul(c).lo.(cc)(a, b)$	$s = (a \times b)_{lo}$ [3]
$s = mul(c).hi.(cc)(a, b)$	$s = (a \times b)_{hi}$
$s = mad(c).lo.(cc)(a, b, c)$	$s = (a \times b)_{lo} + c$
$s = mad(c).hi.(cc)(a, b, c)$	$s = (a \times b)_{hi} + c$

[1] The instruction followed by a c means that the carry in the carry register participates in the operation.
[2] The $.cc$ means that if a carry is generated in the operation result, it is stored in the carry register.
[3] The $.lo$ indicates the lower 32 bits of the product result, the $.hi$ means the higher 32 bits result.

III. Methodology

This section focuses on the acceleration of SHA2 algorithm on RISC-V platform, that is, how to complete the acceleration of SHA2 algorithm on RISC-V platform. We provide the detailed steps of the implementation process, from the compilation of the core steps of SHA-2 algorithm by using its instruction set on RISC-V platform, and the optimization of its flow and structure by using the characteristics of RISC-V instruction set for SHA2 algorithm, so that its running speed is significantly improved compared with before.

A. Classic Mceliece encryption and decryption algorithm

The classical McEliece[5] encryption and decryption algorithm is a public-key cryptosystem known for its strong security and ability to resist quantum computing attacks. This algorithm was proposed by Robert McEliece in 1978 and has become one of the classic algorithms in public key cryptography. Based on the theory of linear code, McEliece algorithm constructs a secure public key cryptosystem by using the error correction ability of linear code and the

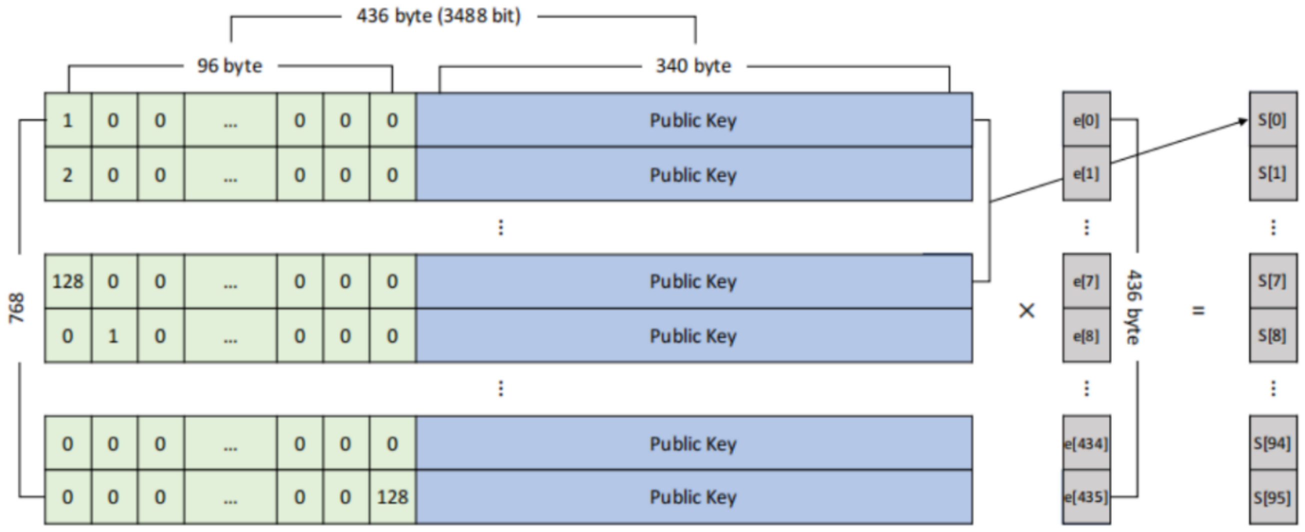

Fig. 2: the scheme of MEceliece

difficulty of solving difficult coding problems. Unlike other public-key cryptography algorithms, McEliece's algorithm is based on linear algebra and coding theory, and does not rely on difficult problems in number theory. This makes McEliece algorithm have excellent security in both traditional computer and quantum computer environments. In traditional computers, because the McEliece algorithm is designed based on the difficult coding problem, an attacker needs to solve a series of difficult mathematical problems in order to crack the encrypted information. In the context of quantum computers, McEliece algorithm adopts code-based encryption mechanism, which is difficult to attack quantum computers and can resist the attacks of quantum algorithms such as Shor algorithm [6]. Classic McEliece is based on binary Goppa codes. The code mainly has the following three characteristics: (1) the composition structure is a kind of random linear code; (2) The decoding algorithm is very fast; (3) The general linear problem that Goppa code security depends on is the NPC problem. The public key of a Goppa code is the product of its generation matrix and a random permutation matrix, and the private key is a random binary intensive Goppa code. During encryption, the error vector is introduced, during decryption, the error vector is recovered by decoding algorithm and private key, and then the plaintext is obtained. Using the above construction method, the public key cryptography schemes based on error correction code such as CFS signature scheme and Niederreiter encryption scheme are constructed. This cryptosystem has a faster encryption and decryption speed than the public key cryptosystem in use today. So far, there is no effective attack method against the cryptosystem based on Goppa code. The security of McEliece algorithm is based on an important problem, that is, the difficult coding problem of linear code. The problem requires that the attacker be able to find the encoded form of the original message, but this requires them to solve an NP-hard problem of finding a low-dimensional subspace in a high-dimensional vector space. For reasonably selected parameters, it is almost impossible for an attacker to solve the problem in an acceptable amount of time. Another important feature of the McEliece algorithm is its robustness and resistance to attack. Because of the linear code, McEliece algorithm has a very high error correction ability and can effectively resist the error and noise in transmission. The design concept of McEliece algorithm and the encryption mechanism based on coding make it have certain flexibility and expansibility. By selecting the right linear code parameters, you can adjust them to your specific security needs, thus striking a balance between security and performance. This flexibility makes the McEliece algorithm suitable for a variety of application scenarios, including data transmission, E-mail encryption, digital signature and other fields.

B. Optimization of M-bit finite field computation on RISC-V instruction set

Algorithm 2 The Improved Montgomery Multiplication

```
Input:
    The unsigned char data[];
    The intermediate variable A-H from state;
Output:
    A-H after iteration.
1:
2: for $i = 0$ to $16$ do
3:
       Load $w_i$ from data:
5:     $w_i$=GETU32(data);
6:     data+=4;
7:     Iterate over A-H using $w_i$
8: end for
```

In the classical Mceliece encryption and decryption algorithm with parameters m = 12, n = 3488, t = 64, the 12-bit finite field is a very important data structure[7], and many important computational steps are developed and derived around it. In the classic Mceliece implementation, F2 is first used to generate a polynomial field, the m-degree polynomial representation above the field element, the coefficient is the element, that is, 0,1, the multiplication in the field, the multiplication of the polynomial, the addition of the calculation. In finite fields, both addition and subtraction can be represented by an XOR operation, and multiplication can be represented by a phase and operation. The data structure used here is gf, or short int, where each bit corresponds to a polynomial coefficient. For example, the polynomial corresponding to 010000011000 is. In this finite polynomial field, the addition and subtraction operations in the field are the XOR operations, and the multiplication operations in the field are the and operations. The 12-bit finite polynomial multiplication also plays a very important role in the algorithm flow. In the public key generation algorithm , the calculation of the transpose matrix cannot be separated from the 12-bit finite field multiplication. In private key generation algorithm, Gaussian reduction of matrix can not be separated from 12-bit finite field multiplication. Decomposes the decoder using the permanent Bellekamp Massey (BM) algorithm. In BM algorithm, the most time-consuming operations are multiplication and inversion on finite fields. In addition, 12-bit finite polynomial multiplication is also the basis for operations such as 12-bit finite field square, inversion, and bit-finite field multiplication. Therefore, the optimization of 12-bit finite field multiplication has a great impact on the overall performance improvement. At the same time, the square operation of the 12-bit finite field is also extremely important to the algorithm. Not only is this operation used many times in the encryption and decryption algorithm, but the inversion operation also depends on the square operation. In the classic Mceliece encryption and decryption algorithm, 12-bit finite field multiplication is roughly divided into the following steps: 1. Complete the multiplication of two polynomials, here mainly using the textbook polynomial multiplication, that is, each term of the polynomial is multiplied by each term of another polynomial, and the result is added. 2. At this time, due to the possible range of the multiplication result, the multiplication is completed on the top and the polynomial reduction is completed by modularization reduction of the multiplication result value. In the classical Mceliece encryption and decryption algorithm implemented with parameters m = 12, n = 3488, t = 6, the finite polynomial field is, so the result needs to be reduced with the polynomial f (z) =. 3. Remove the result by dividing the last 12 bits to ensure that the result is within the limited range.

C. Optimization of polynomial multiplication

The polynomial multiplication[8] here is more time-consuming and will be discussed based on this below. In the classical McEliece encryption algorithm, polynomial multiplication is a very important operation. In McEliece encryption algorithm, multinomial multiplication is widely used in the calculation process of generating check matrix, encryption key and decryption key. However, due to the use of large integers and large matrices in classical McEliece encryption algorithms, polynomial multiplication is very expensive to compute. Multiplication operations for large integers require a lot of bit operations and carry operations, while multiplication rules for large matrices require a lot of multiplication and addition operations. Therefore, polynomial multiplication tends to be one of the most time-consuming operations in the entire encryption algorithm. This high computational cost is an important feature of the classical McEliece encryption algorithm, which provides it with high security. Due to the high computational cost of polynomial multiplication, it is difficult for attackers to crack encryption keys through exhaustive search and other methods. This gives classical McEliece encryption algorithms an advantage against various attacks, such as linear code-based attacks and cryptanalytic attacks. However, because of the high computational cost of polynomial multiplication, the classical McEliece encryption algorithm often has some performance limitations in practical applications. It requires longer encryption keys and longer ciphertext length, which increases the overhead of communication and storage. In addition, the high computational cost of polynomial multiplication also leads to slower encryption and decryption processes.

Karatsuba polynomial multiplication [21] is an efficient polynomial multiplication algorithm, which has significant advantages over traditional Schoolbook multiplication. In the field of computer science and mathematics, polynomial multiplication is an important computational task, and Karta-

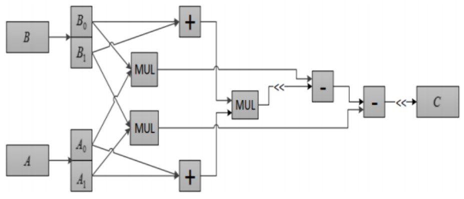

Fig. 3: The structure of Karatsuba

suba algorithm greatly improves the efficiency of polynomial multiplication by using dive-and-conquer strategy to split large numbers. The Karatsuba multiplication algorithm is the first subquadratic method, which is a subquadratic method proposed by Karatsuba and Ofman. This multiplication is first applied to large integers, but can also be applied to polynomials. Since then, this classical method has been extensively studied. Compared with Schoolbook multiplication, Kartasuba algorithm adopts the dive-and-conquer strategy [11] to decompose the original polynomial multiplication problem into smaller scale subproblems, which can reduce the computational complexity by solving these subproblems recursively. This divide-and-conquer strategy reduces unnecessary double calculation and improves the efficiency of the algorithm. Compared with the time complexity of Schoolbook multiplication, it has obvious advantages. Kartasuba algorithm shows good performance in practical application. Because of its efficient calculation method, Kartasuba algorithm is widely used in digital signal processing, polynomial interpolation, polynomial inversion and polynomial solving. In large-scale polynomial multiplication calculation, Kartasuba algorithm can significantly improve computing speed and reduce resource consumption, so it is widely used in modern computer science and engineering fields. Kartasuba polynomial multiplication has obvious advantages over traditional Schoolbook multiplication through the application of dive-and-conquer strategy. It has shown good performance in computational complexity, computational efficiency and practical application, and has become one of the important algorithms in polynomial multiplication. The classical Karatsuba algorithm divides the input multiplier and multiplicand into two binomial polynomials, and then uses only three small multiplications to calculate by mathematical merging, saving one multiplication operation .

IV. Performance Evaluation

The experiment was carried out on the VisionFive2 single-board computer based on the RISC-V instruction set. When the random seed is generated, the AES256 pseudo-random number is repeatedly generated, so that the public key, private key and error vector remain unchanged under the same number of rounds each time the program is run. In this design, the running time of the classic Mceliece reference implementation version and the optimized version with parameters m = 12, n = 3488, t = 6 was calculated. The program generated 10 rounds of random seeds each time it was run, and carried out key generation according to this, and conducted 100 encryption and decryption operations for each round of keys. The test counted key generation, encryption, decryption, and the average run time of each round. After many tests, the running time of each part is shown in the table, in milliseconds.

A. Performance Evaluation

As shown, we conducted experiments to measure the performance of Paillier, including encryption, decryption, and homomorphic addition on RTX 4090. We can notice that when the number of threads come to 128 × 512, the throughput differs only slightly from the peak value. Simultaneously, the computational latency is significantly lower compared to the latency when the number of threads is 256 × 512.

TABLE III: Peak performance of MEceliece

	Encryption	Decryption	Homomorphic Addition
Throughput (kops/s)	79.63	141.56	46,104.42
Latency (ms)	102.88	57.87	0.18
Number of Threads	256 × 512		

V. Conclusion

RISC-V is an instruction set architecture based on open principles, its design is concise, clear, modular, and has scalability and flexibility. With the rapid development and wide application of RISC-V in various fields, the application of classical McEliece algorithm in RISC-V instruction set architecture can not only maintain the security of the algorithm, but also greatly improve the execution efficiency and performance at the hardware level. The implementation and optimization of McEliece algorithm in RISC-V is of great significance for data security. Based on this goal, the design completes the following tasks: (1) Complete the correct implementation of classic Mceliece on the development board based on RISC-V architecture. (2) Using embedded assembly and the characteristics of RISC-V instruction set, complete the optimization implementation of M-bit finite domain computation on RISC-V instruction set, including multiplication, reduction, square and other operations. (3) Complete the optimization of polynomial multiplication, and experiment a variety of different polynomial multiplication algorithms.

In this paper, the classical Mceliece is optimized on the RISC-V instruction set. The experimental results show that the proposed method achieves certain expectations, but there are also some shortcomings, and the performance improvement of the algorithm can be further explored. In future work, we will focus on further optimizing the performance of Mceliece algorithm on RISC-V architecture. We can further explore the performance optimization method of the matrix reduction part to further improve the performance of Mceliece algorithm on RISC-V architecture. By improving and optimizing the key computational parts of the algorithm, we can further improve the performance of key generation, encryption, and decryption. In addition, we will consider embedding more RISC-V instructions to further speed up the execution of the algorithm. By taking advantage of the openness and flexibility of RISC-V architecture, we can design more suitable instructions for Mceliece algorithm and further improve the performance of the algorithm. In addition to performance optimization, we will continue to focus on the security of the algorithm. With the rise of quantum computers, there is a growing need for quantum-secure communication, and we will work to improve the quantum security of Mceliece's algorithm. By exploring cryptosystems that are more resistant to quantum computing attacks, we can provide a stronger foundation for future quantum-secure communications.

References

[1] R. J. McEliece, "A public-key cryptosystem based on algebraic," *Coding Thv*, vol. 4244, pp. 114–116, 1978.

[2] M.-S. Chen and T. Chou, "Classic mceliece on the arm cortex-m4," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 125–148, 2021.

[3] A. DE PICCOLI *et al.*, "Optimized representations in cryptographic primitives," 2022.

[4] NVIDIA, "CUDA C programming guide 12.0," https://docs.nvidia.com/cuda/, 2017.

[5] T. Sumi, K. Morozov, T. Takagi, and J. Shikata, "Efficient implementation of the mceliece cryptosystem," in *computer security symposium*, 2011, pp. 582–587.

[6] M. Sim, S. Eum, H. Kwon, H. Kim, and H. Seo, "Optimized implementation of encapsulation and decapsulation of classic mceliece on armv8," *Cryptology ePrint Archive*, 2022.

[7] M. Nursalman, A. Sasongko, Y. Kurniawan *et al.*, "Generalizations of n-term karatsuba like formulae in gf (2 n) with nayk algorithm." *IAENG International Journal of Computer Science*, vol. 44, no. 4, 2017.

[8] P. W. Shor, "Polynomial time algorithms for discrete logarithms and factoring on a quantum computer," in *Algorithmic Number Theory: First International Symposium, ANTS-I Ithaca, NY, USA, May 6–9, 1994 Proceedings 1*. Springer, 1994, pp. 289–289.