IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 January 2024, accepted 21 January 2024, date of publication 13 February 2024, date of current version 21 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3365523

RESEARCH ARTICLE

A Video Streaming Encryption Method and Experimental System Based on Reconfigurable Quaternary Logic Operators

XINYU ZHOU[1], HONGJIAN WANG[1], KUIYAN LI[1], LIFENG TANG[2], NINGCHUN MO[3], AND YI JIN[4]

[1]School of Computer Science and Technology, Donghua University, Shanghai 201620, China
[2]Shanghai Astronautics Electronic Company Ltd., Shanghai 201821, China
[3]Shanghai Meiduo Communication Equipment Company Ltd., Shanghai 200333, China
[4]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

Corresponding author: Hongjian Wang (hongjian.wang@dhu.edu.cn)

This work was supported by Shanghai Pujiang Program under Grant 21PJD001.

ABSTRACT Multiple-valued logics (MVL) have abundant operation functions which can be used for encryption. A reconfigurable MVL operator can perform all MVL functions with a universal circuit structure at fast operation speed, based on which a one-time-pad cryptosystem is expected to be built. However, we find that when the existing MVL encryption method is applied to video data encryption, the color edges in the plaintext image will remain in the ciphertext image, resulting in partial leakage of information. To solve this problem, we propose byte reorganization and random mask strategies, forming an improved MVL encryption method for video streaming. For verifying the effectiveness of the method, we implement an FPGA-based experimental system to encrypt and decrypt real-time video streaming data. In this system, 16-quit reconfigurable quaternary logic operators are implemented to encrypt, decrypt and derive keys. The process of either encryption or decryption only takes 34 clock cycles. The encryption and decryption modules are capable of processing streaming data at a speed of 6.21 Gbit/s, showing that the system has real-time processing capability. For proving that our method is secure, we compare our improved MVL encryption method with existing image encryption methods in terms of common security evaluation metrics. Experimental results show that our method solves the problem of remained color edge and the ciphertext exhibits good statistical properties.

INDEX TERMS Video and image encryption, multiple-valued logic, reconfigurable quaternary logic operator, field programmable gate array.

I. INTRODUCTION

In numerous real-world application situations, including mobile devices, web applications and sensor monitoring, large amounts of streaming data containing private information are generated and transmitted among multiple devices at all times. These streaming data need to be processed in real time on sending and receiving devices, where most of the data are processed without being saved and processed again. Due to these features of streaming data, traditional data encryption algorithms often fail to achieve the satisfactory performance. For example, the commonly used symmetric encryption AES [1] is a block cipher algorithm whose processing speed is often unable to meet the real-time encryption and decryption requirements of streaming data. The stream cipher algorithm in symmetric encryption has faster processing speed, but its security depends on the strength of the key stream generation algorithm, such as the ZUC algorithm [2].

The associate editor coordinating the review of this manuscript and approving it for publication was S. K. Hafizul Islam.

The asymmetric encryption algorithms are much slower and also not suitable for high-speed real-time encryption and decryption application scenarios. Therefore, it is necessary to develop a secure and robust data encryption technique with fast processing speed to protect streaming data.

One-time-pad (OTP) encryption method [3] is considered as an absolutely secure encryption method which enables each plaintext character to be replaced with a ciphertext character with equal probability. Shannon demonstrated that this encryption method is indecipherable when the used keys satisfy the three conditions of 1) being completely random, 2) having the same length as the plaintext, 3) being used only once [4]. In this case, the strict restrictions on the keys make it difficult to implement OTP encryption. Although some encryption methods with faster processing speed have emerged [5], [6], [7], [8], none of these achieve the security level of OTP. However, MVL operations can provide a large variety of symbol replacement rules that meet the key requirements of OTP encryption to achieve high-security streaming data encryption.

In 2021, Wang et al. proposed an OTP block encryption method based on reconfigurable MVL operators for encryption and decryption [9]. This method is abbreviated as the *MVL encryption method* in the following. In this paper, we found that the edge information of images still existed in the ciphertext after the method was directly applied to encrypt video streaming. Through theoretical and experimental analyses of MVL encryption method, we have found the root cause of this defect. When encrypting the same symbols in the long plaintext strings, the frequency of derived key changes gradually decreases, which we call the *derived key convergence* problem.

To solve this problem, we propose an improved MVL encryption method for the characteristics of video data and use FPGA to implement an experimental system based on reconfigurable quaternary logic operators. We show that the improved MVL encryption method generates good-quality ciphertexts at real-time processing speed when applied to video streaming data encryption.

The contributions of this paper are summarized as follows: (1) We add byte reorganization strategy before the MVL encryption. Rearranging multiple adjacent grouped plaintext data achieves visual diffusion in a larger span. (2) In order to achieve a better information confusion effect, we introduce random mask strategy based on linear feedback shift register (LFSR) for disrupting the regular data before the MVL encryption. (3) Based on 16-quit reconfigurable quaternary logic operators, we implement a video streaming encryption/decryption experimental system on FPGA. We combine the two above-mentioned strategies and successfully apply the improved MVL encryption method to real-time video streaming encrypting scenarios, showing that the method is fast. (4) We perform comparative experiments and security analyses, where we compare our improved MVL encryption method with existing image encryption methods in terms of common security evaluation metrics including peak signal-to-noise ratio (PSNR), correlation coefficient, number of pixels change rate (NPCR) and uniform average change intensity (UACI), showing that the method is secure.

The rest of the paper is organized as follows. Section II summarizes the related research work about MVL, its cryptological applications and image encryption methods which are relevant to the proposed method. Section III introduces the basis of this paper and the theory of MVL encryption method. Section IV discusses the derived key convergence problem and analyzes its causes when the original MVL encryption method is directly applied to video data encryption, then proposes the improved method by adding byte reorganization and random mask to address the problem. Section V designs and implements the FPGA-based real-time video streaming encryption/decryption experimental system to verify the proposed method. Section VI analyzes the encryption performance of the improved MVL encryption method on FPGA experimental system. Section VII proves the security of the proposed method by analyzing its key space and comparing it with existing image encryption methods. Section VIII discusses the efficiency of the proposed schema. Section IX concludes the work of this paper.

II. RELATED WORK

A. MVL

The cryptographic theory and related implementation techniques in this paper are based on MVL operations and reconfigurable MVL operators. Compared with classical binary logic, MVL is able to represent more information states and has richer logical operations. MVL has a long history of research [10]. Researchers have made many explorations in the fields of MVL components [11], [12], [13], MVL computers [14], [15] and so on. Zarin et al. summarized the ternary logic construction methods for ten electronic components such as metal-oxide-semiconductor field-effect transistor (MOSFET), resonant tunneling diode (RDT) and single electron transistor [11]. Based on a dedicated hardware description language ARITH [12], Yuki et al. proposed a high-level design method for MVL circuits [13]. MVL computers have also been invented, such as the ternary electronic computer SETUN [14] and the ternary optical computer SD16 [15] based on the optical polarization state.

However, all of the above methods are based on some specific MVL (e.g. ternary) and they need dedicated hardware to implement corresponding components, which hinders the popularization of these technologies. In 2018, Jin et al. proposed a method and technique for constructing reconfigurable MVL electronic operators using traditional binary logic electronic components [16]. This technology provides a universal hardware structure for different MVL operators (ternary, quaternary and etc.) and takes full advantage of mature integrated circuit technology for implementation. Therefore, we use this technique in this paper to implement reconfigurable quaternary logic operators on FPGA and build

a quaternary logic encryption experimental system based on reconfigurable quaternary logic operators.

B. MVL ENCRYPTION

MVL has been applied in the field of cryptography. For example, Han proposed an encryption method based on MVL array transformation [17], which can replace plaintext into ciphertext with different orders of arrangement. Since then, some studies using several simple MVL operators to complete complex array transformation algorithms, which in turn can achieve encryption and decryption [18] or generate pseudo-random numbers [19], [20], have been proposed one after another. These studies try to use MVL for cryptography but did not take advantage of the large quantity of MVL operator types. Also, the security of their encryption schemes needs to be further investigated. For example, Dai et al. demonstrated that the encryption systems based on MVL array transformation can be broken by known-plaintext attacks [21].

In order to leverage the large number of MVL operational rules to further improve the security of MVL cryptosystems, Singh et al. used a Latin square design instead of heterogeneous operations in Vernam ciphers as an attempt to combine MVL with OTP encryption [22]. However, the conditions of Latin square limit its number and, in fact, some MVL operators can be used as encryptors/decryptors even they are not Latin squares. In 2021, Wang et al. proposed a practical MVL encryption method [9], where the variety of MVL operations is much richer than the Latin squares. They used the above mentioned reconfigurable MVL electronic operators [16] to implement their method, which can prestore enough randomly selected generalized keys including encryption/decryption operations, key derivation operations and seed keys for both encryption and decryption parties with few hardware resources. Through the *reservation code*, the generalized keys of both encryption and decryption are automatically negotiated and synchronized to ensure that the security of encryption reaches the level of OTP encryption.

We found that the method proposed by Wang et al. [9] has a weakness: when applied to video streaming encryption, it has the problem of derived key convergence, which leads to partial information leakage from the ciphertext. In this paper, we propose tow strategies, namely byte reorganization and random mask strategies, to solve the problem and improve the MVL encryption method.

C. IMAGE ENCRYPTION

Images including highly correlated and redundant data are difficult to encrypt in such a way that an adversary cannot obtain any meaningful information from encrypted images. Substitution boxes (S-boxes) and random numbers are often used to form substitution-diffusion framework that causes confusion in image encryption [23], [24], [25], [26], [27], [28], [29]. Existing S-boxes schemes all suffer from the drawback that the cryptographic properties, such as nonlinearity and differential approximation probability, of the S-box generator are not guaranteed. Hence, such schemes do not have provable security against modern attacks.

Although many researches have paid a lot of attention to improve the cryptographic strength by various methods, they may not be concerned with the efficiency of encryption. For example, Hayat and Azam proposed the substitution-diffusion image encryption scheme based on elliptic curves for generating dynamic S-boxes and random numbers [23]. But their method can be computationally costly if the underlying elliptic curves are large for pixel-level encryption. Hua et al. constructed S-boxes using the complete Latin squares and chaotic systems [24] but the matrix operations on Latin squares can be time-consuming.

Notably, combining the efficiency and desirable statistical properties of Henon map with dynamic S-Boxes and elliptic curve cryptography, the scheme proposed by Ibrahim et al. is computationally efficient with an encryption throughput close to 60 MB/s [25]. This computational efficiency may still not be applicable to real-time encryption scenarios, such as video streaming encryption scenarios consisting of live images.

Comparing with some of the above encryption methods, the method proposed in this paper not only performs well in important security metrics including histogram, PSNR, correlation coefficient, NPCR and UACI, but also achieves an efficiency of 6.21 Gbit/s on our FPGA experimental system.

III. MVL ENCRYPTION METHOD

This section introduces the MVL encryption method and mode. For the convenience of the discussion, the following MVL operations and MVL inverse operations are defined first.

A. MVL OPERATIONS

Definition 1: A two-input MVL operation is a binary function and satisfies the following conditions. Suppose the set $Z = \{0, 1, 2, \cdots, n-1\}$, if $\forall a \in Z$ and $\forall b \in Z$, $f(a, b) = c \Rightarrow \forall c \in Z$, then the binary function f is a two-input n-valued logical operation (hereafter abbreviated as *n-valued logical operation*).

An n-valued logical operation is the process of taking two n-valued inputs and mapping them by a function to get an n-valued output. The function mapping rules of any n-valued logical operation can be represented by an n-rows and n-columns truth table. Filling n-valued logic results to n rows and n columns of the table, the n-valued logic truth table for a particular n-valued logical operation is formed. For example, Table 1 gives an example of a quaternary logic truth table. When filling in the $n \times n$ spaces of the n-valued logic truth table, there are n choices for each space so the total number of different n-valued logic truth tables is $n^{(n \times n)}$.

B. MVL INVERSE OPERATIONS

Definition 2: For some two-input n-valued logical operations f satisfying the conditions in Definition 1, if there exists a binary function f^{-1} that $\forall a \in Z$ and $\forall b \in Z$,

TABLE 1. Example of quaternary logic truth table, inverse operation truth table, 2-bit symbolic representation.

b \ a	0	1	2	3
0	3	1	2	0
1	0	2	3	1
2	2	1	3	0
3	1	3	0	2

b \ a	0	1	2	3
0	3	1	2	0
1	0	3	1	2
2	3	1	0	2
3	2	0	3	1

b \ a	00	01	10	11
00	11	01	10	00
01	00	10	11	01
10	10	01	11	00
11	01	11	00	10

$f(a,b) = c \Leftrightarrow f^{-1}(c,b) = a$, then this binary function f^{-1} is the inverse n-valued logical operation of the f.

Theorem 1: The inverse n-valued logical operations are also two-input n-valued logical operations.

Theorem 2: Suppose the n-valued logical operation $f(a,b) = c$. All inputs a form the set A and all outputs c form the set C. The condition for the existence of the inverse operation f^{-1} of f is that when fixing b, f satisfies bijection when mapping from the set A to the set C.

Proof: It is known that $f(a,b) = c$. In order to let the result obtained by the inverse operation $f^{-1}(c,b)$ be the unique a, when the set A is mapped to the set C through f, it is necessary to ensure that a corresponds to c one-to-one. According to Definition 1, $A = C = \{0, 1, 2, \cdots, n-1\}$, so f satisfies the bijection.

Theorem 3: The total number of n-valued logical operations with inverse operation is $(n!)^n$.

Proof: According to Theorem 2, the truth table of n-valued logical operations with inverse operation has the following characteristics, c is not repeated in each row. There are $A_n^n = n!$ ways to fill c in each row. Because n-valued logic truth tables have n rows, there are a total of $(n!)^n$ truth tables in which c is not repeated in each row.

In order to distinguish n-valued logical operation with inverse operation from general n-valued logical operation, the n-valued logical operation with inverse operation is denoted as f_e and its inverse operation is denoted as f_d. According to Theorem 1, f_d is also an n-valued logical operation that can be represented by an n-valued logic truth table. The truth table of f_e can be transformed into the truth table of f_d.

Suppose $f_e(a,b) = f_e(z_i, z_j) = z_k$. Then in the truth table f_e, the value of the z_j row z_i column is z_k. At the same time, $f_d(c,b) = f_d(z_k, z_j) = z_i$, the value of the z_j row z_k column in truth table f_d is z_i. When transforming from the truth table f_e to the truth table f_d, fill the z_i into the position of the z_j row z_k column in the truth table f_d. The transformation process is shown in Fig. 1. Table 1 gives an example of a quaternary logic truth table and its inverse truth table.

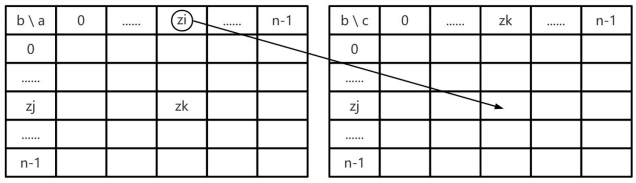

FIGURE 1. MVL truth table transformation.

C. REPRESENTATION OF MVL VALUES

There are various ways to represent n-valued logic values [16]. One single symbol can represent an n-valued logic value. For example, {0,1,2,3} with a total of four single symbols are used above to represent four quaternary logic values. It is also possible to use several symbols to represent an n-valued logic value. For example, the quaternary logic values can be represented by {00,01,10,11}. This representation is shown in Table 1.

With this multi-bit symbol representation, conventional binary high-low signals are also capable of representing arbitrary MVL values. Such as 2 bits data can represent a quaternary logic value and 3 bits data can represent an octonary valued logic value.

D. MVL ENCRYPTION AND DECRYPTION OPERATIONS

The method of encryption using MVL operations is a substitution encryption. For encryption, the plaintext p and the key k are two inputs to the n-valued logical operation f. After the n-valued logical operation $f(p,k) = c$, the plaintext p is encrypted into the ciphertext c. For decryption, the ciphertext is decrypted to get the plaintext by the corresponding inverse operation $f^{-1}(c,k) = p$.

The n-valued logical operations that can be used for encryption must have an inverse operation, so the encryption operation must be f_e and the decryption operation is the corresponding inverse operation f_d. According to Theorem 3, there are $(n!)^n$ operations in n-valued logical operations that can be used as encryption operations. Taking quaternary logical operations as an example, there are 331776 kinds of operations that can be used for encryption. The huge space of MVL operations provides safety guarantee.

E. MVL BLOCK ENCRYPTION METHOD

In 2021, Wang et al. invented MVL-based block encryption method [9]. This encryption method is not only based on MVL operations, but also improves the traditional block encryption mode. It replaces the generalized key for each encryption according to the large MVL operation space to achieve OTP encryption.

The encryption schema is shown in Fig. 2. Both the plaintext and the key are n-valued data in group of m. Encryption is achieved by n-valued logical operations F_e

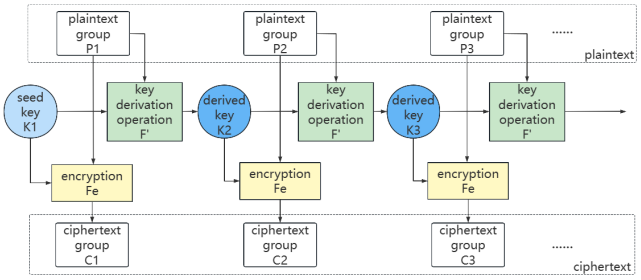

FIGURE 2. MVL encryption mode.

and key derivation is achieved by n-valued logical operations F', where F_e consists of m different n-valued logical operations f_e and F' consists of m general n-valued logical operations. The first n-valued plaintext group P_1 and seed key K_1 are encrypted by $F_e(P_1, K_1)$ to generate the first n-valued ciphertext group C_1, while P_1 and K_1 are derived by $F'(P_1, K_1)$ to generate K_2, which is used as the key of the second group. The encryption and key derivation of subsequent groups are completed sequentially according to the above process.

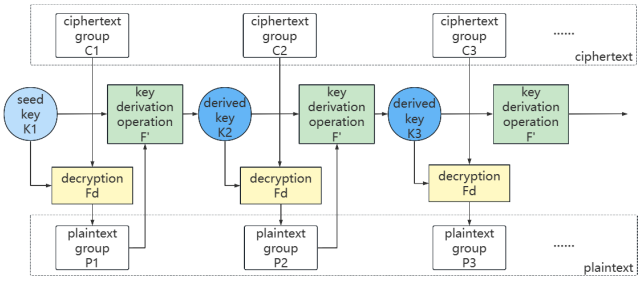

FIGURE 3. MVL decryption mode.

The decryption schema is shown in Fig. 3. In decryption, seed key $K1$ and derivation operation F' are the same as those in encryption. Each f_d composing F_d is the inverse operation of f_e composing F_e. The first ciphertext group C_1 and seed key K_1 are decrypted by $F_d(C_1, K_1)$ to get the first plaintext group P_1. P_1 and K_1 are derived by $F'(P_1, K_1)$ to get the second round of key K_2. The later ciphertext groups are also decrypted in this way.

In the block encryption and decryption mode with m n-valued logical operations, changing the encryption operation F_e directly changes the ciphertext result, while changing the seed key K_1 or derivation operation F' causes the derived key to change, which eventually changes the ciphertext result. The seed key, the encryption operational rules and the derivation operational rules all affect the ciphertext, so these are collectively called *generalized keys* [9], which together form a huge encryption space.

IV. IMPROVED MVL ENCRYPTION METHOD FOR VIDEO STREAMING

The MVL encryption method is able to obtain good results for encrypting text data. However, when trying to apply to the encryption of video streaming, it is found that this encryption mode suffers from the derived key convergence problem, which leads to the image edge information being exposed in the ciphertext. This section first analyzes the derived key convergence problem and then introduces byte reorganization and random mask improvement strategies to improve the encryption results.

A. DERIVED KEY CONVERGENCE PROBLEM

Consider the following case to analyze the problem of MVL encryption mode. It is assumed that the plaintext grouping P and the encryption operation F_e as well as the derivation operation F' remain unchanged during the whole encryption process.

Using the quaternary logical operation F' for a certain key K derivation, the i^{th} quit (denoted K_i) pass through F' to produce the i^{th} quit of K'. In this way it may be possible to make exactly $K'_i = F'_i(P_i, K_i) = K_i$. When generating the next derived key K'', since P, K'_i and F' are unchanged, $K''_i = F'_i(P_i, K'_i)$ is still equal to K_i. This leads to the fact that if the values of some bits in the K' are the same as those in previous K, then those values of the bits remain the same in subsequent derivations. After several derivations, the derived key converges to the same key. Eventually, the whole derived key remains unchanged. Once the derived key remains unchanged, the same plaintext P produces the same ciphertext C through the same F_e. Other n-valued logical operations for encryption are the same.

The specific consequence of the derived key convergence problem in video streaming encryption is as follows. The same plaintext pixel values are replaced with the same ciphertext pixel values, resulting in an encrypted video image where the color is changed but the boundaries between the different colors are still clear. The encryption effect is shown in Fig. 4. Retaining color boundaries means retaining semantic information. One can still distinguish different objects and obtain other information from the encrypted video images by relying on large color blocks, which is a fatal shortcoming for video encryption.

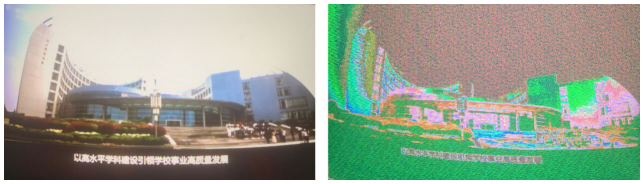

FIGURE 4. Original video and encryption video with derived key convergence.

After analysis, there are a large number of the same video data in video streaming to ensure the smoothness of the video. It is these same video data that cause the derived key convergence problem. Based on the MVL encryption and decryption mode, we introduce byte reorganization and random mask techniques to propose an improved MVL encryption method. Firstly, we reorder the plaintext data

through byte reorganization module among a certain number of plaintext groups. Then, we perform xor operation on the reorganized groups and the pseudo-random sequence. The result of these two steps is sent to the F_e and F'. In the decryption mode, the ciphertext data firstly goes through the F_d. Then we perform xor operation on the pseudo-random sequence which is the same as encryption. The final step is reset the byte reorganization. The improved encryption and decryption mode is shown in Fig. 5.

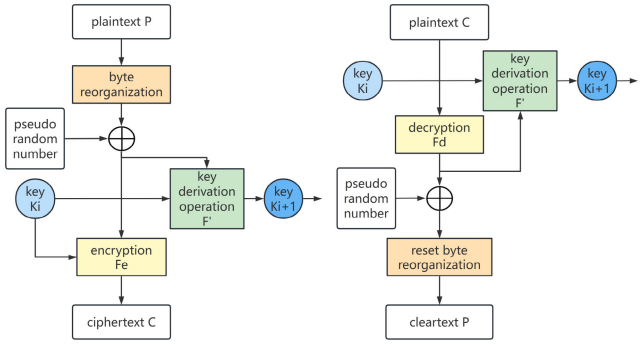

FIGURE 5. Improved encryption and decryption mode with the byte reorganization and random mask.

B. BYTE REORGANIZATION

Byte reorganization is to recombine the data of multiple plaintext groups to destroy the original byte structure and form more chaotic data groups. A byte contains 8 bits of data. In order to get a better chaotic effect, the number of groups l should meet the requirement of $l \ mod \ 8 \neq 0$. Obviously, the larger of l, the larger span of data covered by one byte reorganization process and the better effect should be. However, taking too large value for l also brings disadvantages. The more data is cached during one byte reorganization process, the higher latency causes.

Considering the compromise between chaotic effect and delay, this paper adopts 32 bits as one plaintext group and implements byte reorganization every 33 plaintext groups. The reorganization schematic is shown in Fig. 6. Each byte reorganization process requires caching 32×33 bits data. The data are arranged into 32 rows and 33 columns. The 32 bits data for each plaintext group are arranged sequentially in rows, then the 32 bits data in each column are combined to form a new reorganized group.

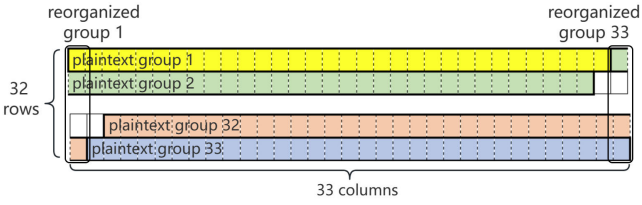

FIGURE 6. Reorganization grouping schematic.

C. RANDOM MASK

In this paper, the reorganized groups are performed xor operation on the pseudo-random sequence to apply the random mask. Theoretically, any random number generator is applicable to the schema proposed in this paper. Since the random mask in this paper serves to mask the regular color block information in the plaintext and thus eliminate the remaining color edges. The security of the ciphertext is guaranteed by the MVL encryption process so the random mask does not require cryptographic security. Considering the easily hardware implementation, LFSR is used to generate pseudo-random sequence as random mask [30]. The LFSR details are described in Section V-D.

V. VIDEO STREAMING ENCRYPTION EXPERIMENTAL SYSTEM

The quaternary logic video streaming system is implemented on Xilinx AXU2CGB to verify the encryption performance and processing speed of the improved MVL encryption method. In experiment, we select 32 bits data as a data group, while implement encryption, decryption and key derivation by 16 quaternary logic operations. The experimental system focuses on analyzing whether the key derivation convergence problem is solved and whether it is suitable for real-time processing, rather than analyzing whether the security level reaches OTP. So only two sets of F_e and F_d for encryption and decryption are preset, while the seed key K_1 and the F' for key derivation are fixed.

A. RECONFIGURABLE QUATERNARY LOGIC OPERATOR

The reconfigurable MVL electronic operator hardware structure invented by Jin et al. [16] is used to implement quaternary logic operations in this paper. The structure of reconfigurable quaternary logic operator is shown in Fig. 7. The 1-quit reconfigurable quaternary logic operator has four reconfiguration registers, RG^0, RG^1, RG^2 and RG^3. Writing reconstruction instructions to the registers determines the quaternary logic operational rule for this 1-quit operator. By rewriting different reconstruction instructions, different quaternary logic operational rules can be reconfigured. Thus, the 1-quit reconfigurable quaternary logic operator can become any one of the 4^{16} quaternary logic operators with different reconfiguration instructions.

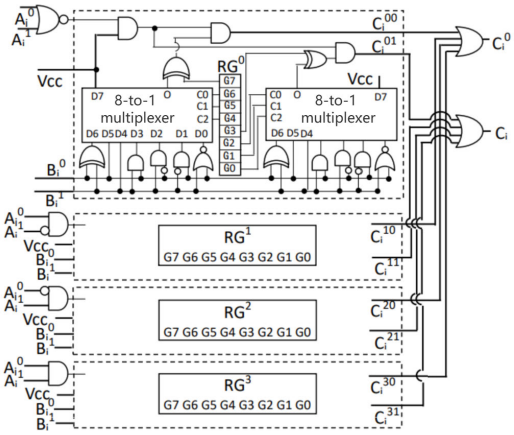

FIGURE 7. 1-quit reconfigurable quaternary logic operator circuit.

By combining 16 1-quit reconfigurable quaternary logic operators together, a 16-quit reconfigurable quaternary logic operator for parallel calculating is formed. We can use the 16-quit quaternary logic operator by following two steps. Firstly, the required reconfiguration instructions were written to each quaternary operator. Then two 32-bit data as inputs for the 16-quit quaternary logic operator in the manner of one quaternary data per 2 bits. The 16 1-quit quaternary logic operators process in parallel based on each respective reconfiguration instruction to obtain a 32-bit quaternary result. Each reconfigurable 1-quit quaternary logic operator is independent of each other, which provides the basis for high-speed processing of streaming data.

B. ENCRYPTION AND DECRYPTION COMPONENTS

The experimental system is built based on the MVL encryption and decryption component structure invented by Wang et al. [9]. As shown in Fig. 8, the encryption component consists of an encryptor and a key derivation operator, while the decryption component consists of a decryptor and a key derivation operator. The encryptor, decryptor and key derivation operator are all 16-quit reconfigurable quaternary logic operators with identical hardware structures.

Write all the reconstruction instructions of the 16-quit quaternary logic operator to the encryptor, decryptor and key derivation operator. The reconstruction instructions ensure encryptor process F_e, decryptor process F_d and key derivation operator process F'.

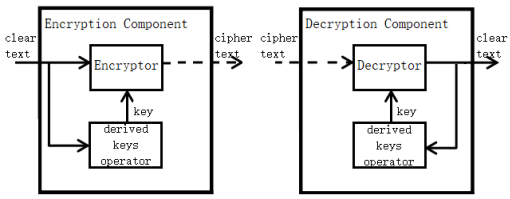

FIGURE 8. MVL encryption and decryption component structure.

C. OVERALL DESIGN OF EXPERIMENTAL SYSTEM

Experimental system is developed on Xilinx AXU2CGB development board. The Xilinx AXU2CGB is a system on chip (Soc) with an integrated FPGA and ARM microprocessor. Thus, the development board can be roughly divided into two parts: programmable logic (PL) and processing system (PS). Based on this structure of the development board, this paper designs and develops an experimental system using both AN5641 MIPI camera and AN970 LCD touch screen as external devices of the development board. Overall design is shown in Fig. 9.

The original video data is captured by the camera and transformed into encrypted video and decrypted video after the improved encryption and decryption processes. By clicking the buttons on the touch screen, users can select the quaternary logic operations used for encryption/decryption and change the video being displayed which could be original video, encrypted video or decrypted video.

The system specifically implements two major groups of functions. The first group includes video streaming processing functions such as format conversion, gamma correction, encryption, decryption and video display. The second group includes several system control functions, which control the encryptor/decryptor to set the quaternary operational rules and control the display selector to show the original, encrypted or decrypted video. The video streaming processing function module is shown in Section V-D and the system control function module is shown in Section V-E.

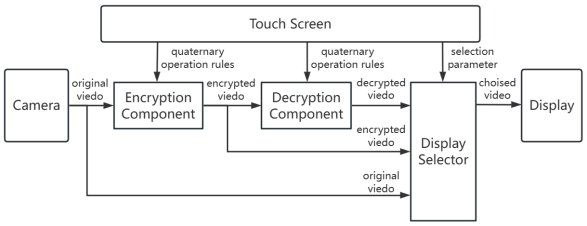

FIGURE 9. Video streaming data system architecture.

D. VIDEO STREAMING PROCESSING

The video streaming processing function module is shown in Fig. 10. The AN5641 MIPI camera is accessed from the PL of the development board. The video is stored in the video register on PL side after format conversion module, gamma correction module, encryption module, decryption module and selection module. The PS reads the video from the register and passes the video data from the mini dp interface according to the DisplayPort protocol. The format conversion module converts the MIPI RAW10 format to RGB format. The format conversion and gamma correction modules are implemented by means of the video processing module provided by Vivado software [31], [32], [33].

In the experiment, we reorganize 33 plaintext groups through a double-buffer pipeline hardware structure. Two buffers of size 1056 bits were created. The plaintext group of 32 bits per clock cycle is deposited into the first buffer sequentially. After 33 clock cycles of caching, the plaintext groups fill the first buffer. At the 34^{th} clock cycle, all the data in the first buffer is deposited into the second buffer. The second buffer outputs the first column of reorganized group according to the scheme in Fig. 6. Meanwhile, the plaintext group of the 34^{th} clock cycle overwrites the data in the first buffer from the beginning. Every clock cycle thereafter, the first buffer deposits a plaintext group and the second buffer outputs a reorganized group. The data from the first buffer is deposited into the second buffer every 33 clock cycles.

The random masks needed for the encryption process and the decryption process are generated by the same LFSR structure. The LFSR consists of 32 registers with 5 taps set shown in Fig. 11. The LFSR is seeded with a preset set of 32-bit random sequences. The seed is written to the registers when the development board is powered on.

E. SYSTEM CONTROL

The system control function module is shown in Fig. 12. The AN970 LCD touch screen is accessed from the PL side of the development board. The display interface of the touch

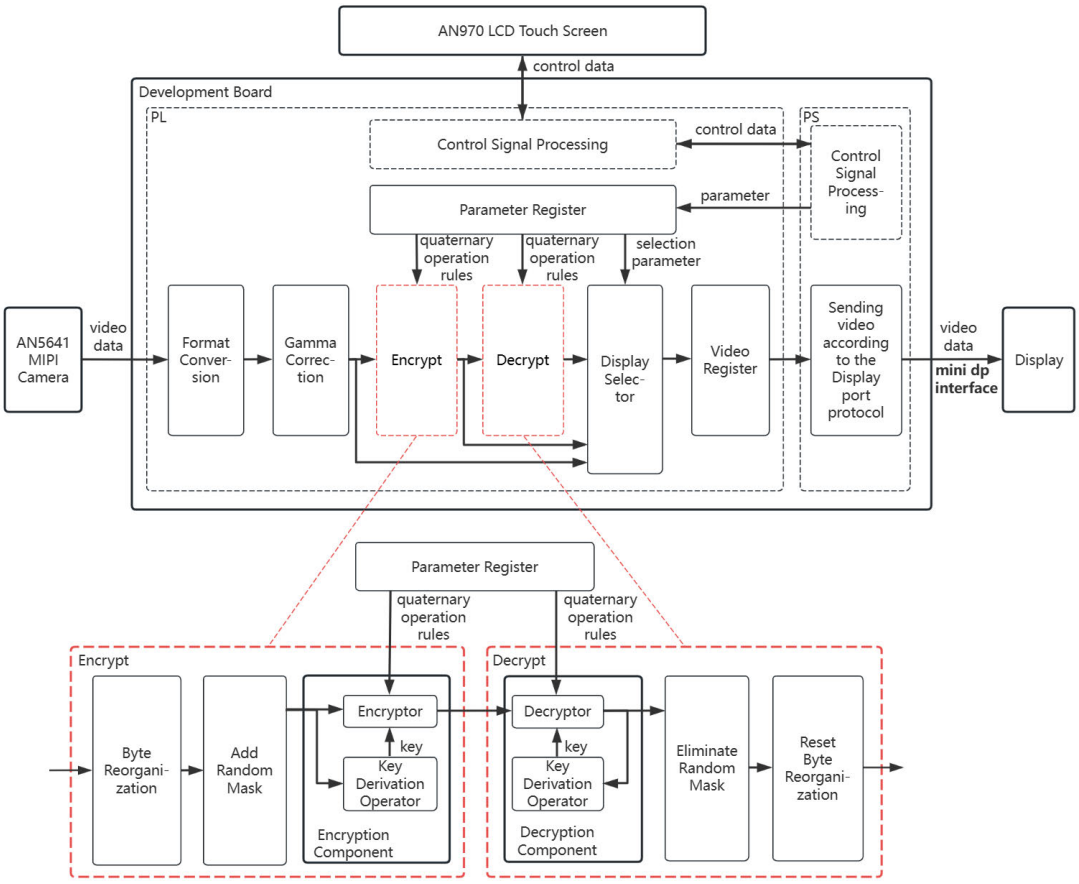

FIGURE 10. Video streaming processing module.

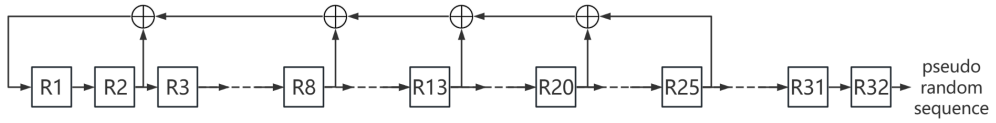

FIGURE 11. Linear feedback shift register structure.

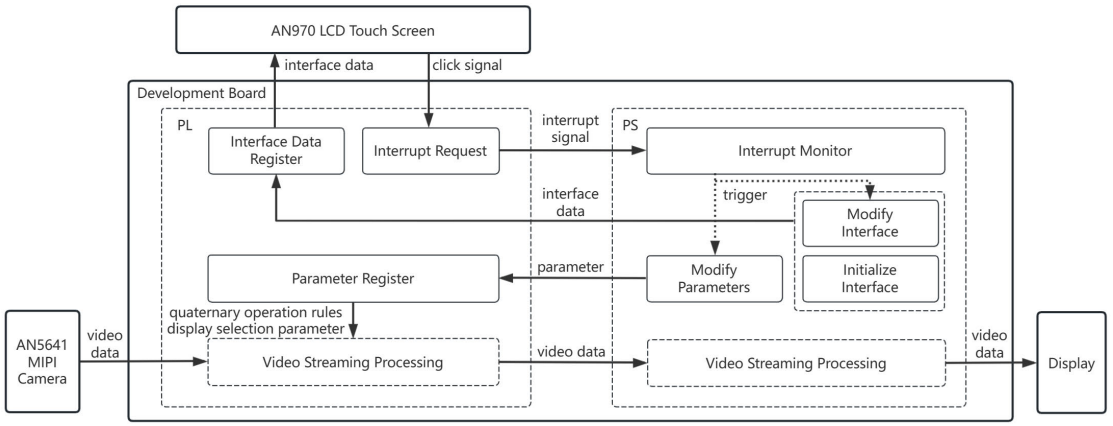

FIGURE 12. System control module.

screen is developed in PS. The interface data is passed to PL and stored in the register, then passed out from the pins of the touch panel to complete the touch panel interface display. By clicking buttons in the touch screen interface, the touch screen generates a signal. When signal is passed to the PL side, PL requests an interrupt. After receiving the interrupt, the PS side triggers two function modules: one is used to modify the value of the parameter register

and the other is used to modify the button style in the interface.

VI. EXPERIMENTAL VERIFICATION

The video streaming encryption experimental system is shown in Fig. 13. In the first row on the touch screen, there are four buttons E_G0, E_G1, D_G0 and D_G1, which control the quaternary operational rules written to the encryptor (E_G0 and E_G1) and decryptor (D_G0 and D_G1). In the last row on the touch screen, there are three buttons P, E and D, which control the system to display original video (P), encrypted video (E) and decrypted video (D), respectively. By this way, the experimental system can output the original video, encrypted video, successful decrypted video with matching rules and failed decrypted video with mismatching rules. The results are shown in Fig. 14.

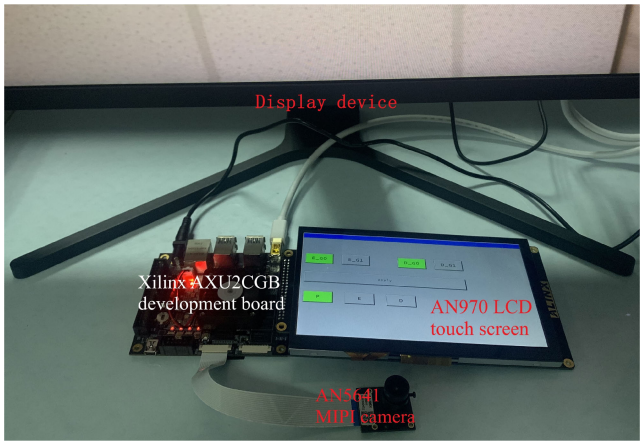

FIGURE 13. Video streaming encryption experimental system.

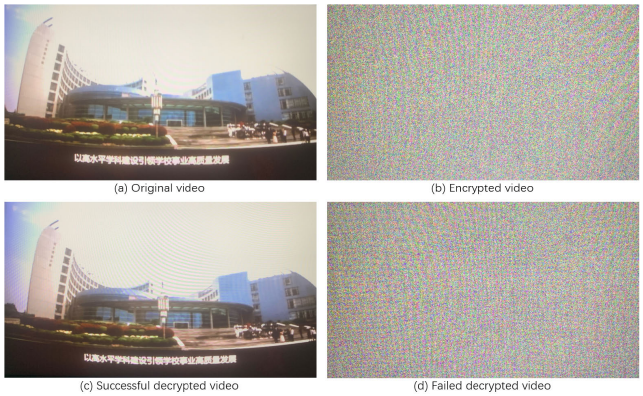

FIGURE 14. Original video, encrypted video and failed decryption video.

To further verify the encryption effectiveness and security of the encryption mode, three illegal decryption operations are performed according to Fig. 15. Experimental results show that none of the three approaches can reveal color edges or any other recognizable information. The inability to recover the color edges again after adding byte reorganization and random masks proves that it is safe to solve the derived key convergence problem in this way.

The byte reorganization module and the random mask module each enhance the degree of obfuscation from a different perspective. The following experiments are conducted

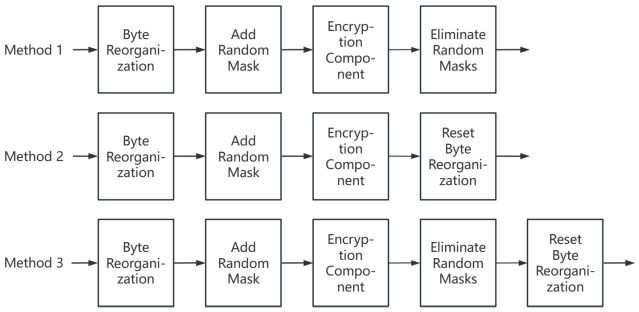

FIGURE 15. Three illegal decryption methods.

TABLE 2. Five encryption and decryption modes.

Mode	Construction
Mode 1	quaternary logic operation + byte reorganization + random mask
Mode 2	quaternary logic operation (proposed by Wang et al. [9])
Mode 3	quaternary logic operation + random mask (as shown in Fig. 16)
Mode 4	quaternary logic operation + byte reorganization (as shown in Fig. 17)
Mode 5	Random mask encryption (based on LFSR pseudo-random sequence)

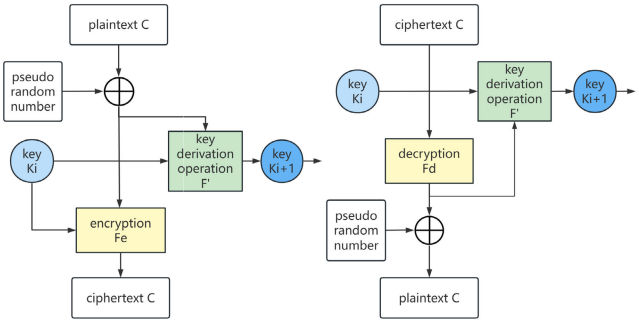

FIGURE 16. Encryption and decryption mode with random mask added to quaternary logic.

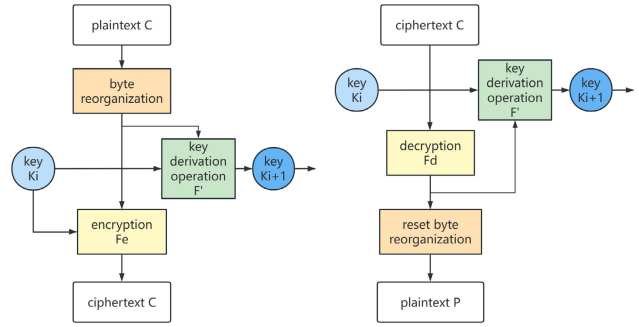

FIGURE 17. Encryption and decryption mode with byte reorganization added to quaternary logic.

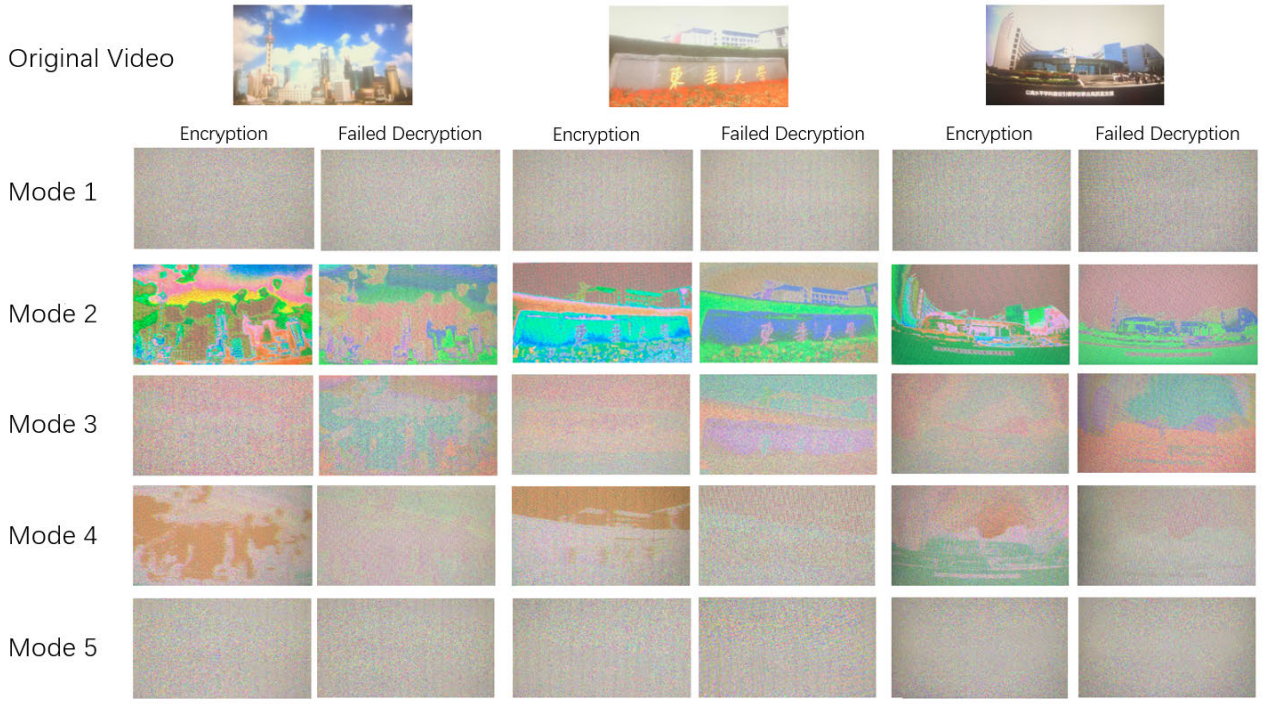

FIGURE 18. Comparison of the encryption and decryption results of five modes.

for the five modes shown in Table 2 to verify the effectiveness of each module.

Five modes are implemented and compared for encryption effect and failed decryption effect. The pseudo-random sequence in Mode 1, Mode 3 and Mode 5 are generated by the LFSR as shown in Fig. 11. Mode 1, Mode 2, Mode 3 and Mode 4 all use 32 bits as a plaintext group and use 16-quit quaternary logical operations for encryption, decryption and key derivation. Mode 1 to Mode 4 demonstrate the effect of failed decryption when the encrypted quaternary logic and the decrypted quaternary logic do not match. Mode 5 sets different seeds for encrypting and decrypting LFSR to generate different pseudo-random sequences to demonstrate the failed decryption effect. The effects are shown in Fig. 18.

Color edges are present in the effect of both encryption and failed decryption in Mode 2. Encryption effect of Mode 3 is fine, but the color block is revealed again after failed decryption. Color edges are still present in the encryption result in Mode 4. In addition, the encryption and failed decryption effects obtained by using only LFSR with random masks are similar to those of Mode 1, but due to the strong periodicity of the LFSR pseudo-random sequence, the use of LFSR in stream ciphers is not secure. Its security is also lower than that of the MVL encryption method with OTP encryption effect.

VII. SECURITY ANALYSIS

A. KEY SPACE

The seed key K_1, the encryption operation F_e and the key derivation operation F' together form the generalized key (K_1, F_e, F') for the MVL encryption method. Thus, the size of the key space is equal to the multiplication of the space sizes of the three generalized key components. When the quaternary logic block encryption with quaternary data in group of 16 (32 bits data as a group), as experimentally implemented in this paper, the K_1 space size $= 2^{32} \approx 4.29 \times 10^9$, the F_e space size $= (4!)^{4\times16} \approx 2.15 \times 10^{88}$ and the F' space size $= 4^{4\times4\times16} \approx 1.34 \times 10^{154}$. Thus, the key space K $= 4.29 \times 10^9 \times 2.15 \times 10^{88} \times 1.34 \times 10^{154} \approx 1.23 \times 10^{251}$. This huge key space can resist brute-force attacks and provide conditions for practical OTP technology based on MVL operators.

B. HISTOGRAM ANALYSIS

A histogram plots the number of pixels at each grayscale intensity level to show the pixel distribution. It is widely used to evaluate the quality of encryption. The histograms of gray images, color images and their encrypted images through improved MVL encryption are shown in Fig. 19. Both the original image and its histogram show that the original image has a very distinct pattern. But after the improved MVL encryption, the histograms of the encrypted images show that the encrypted images are uniformly distributed and one cannot get any useful information from their histograms.

C. PSNR ANALYSIS

PSNR is used to measure the difference between two images. When using PSNR to assess the degree of loss in the decrypted image, the smaller the PSNR is the greater the difference between the original image and encrypted image

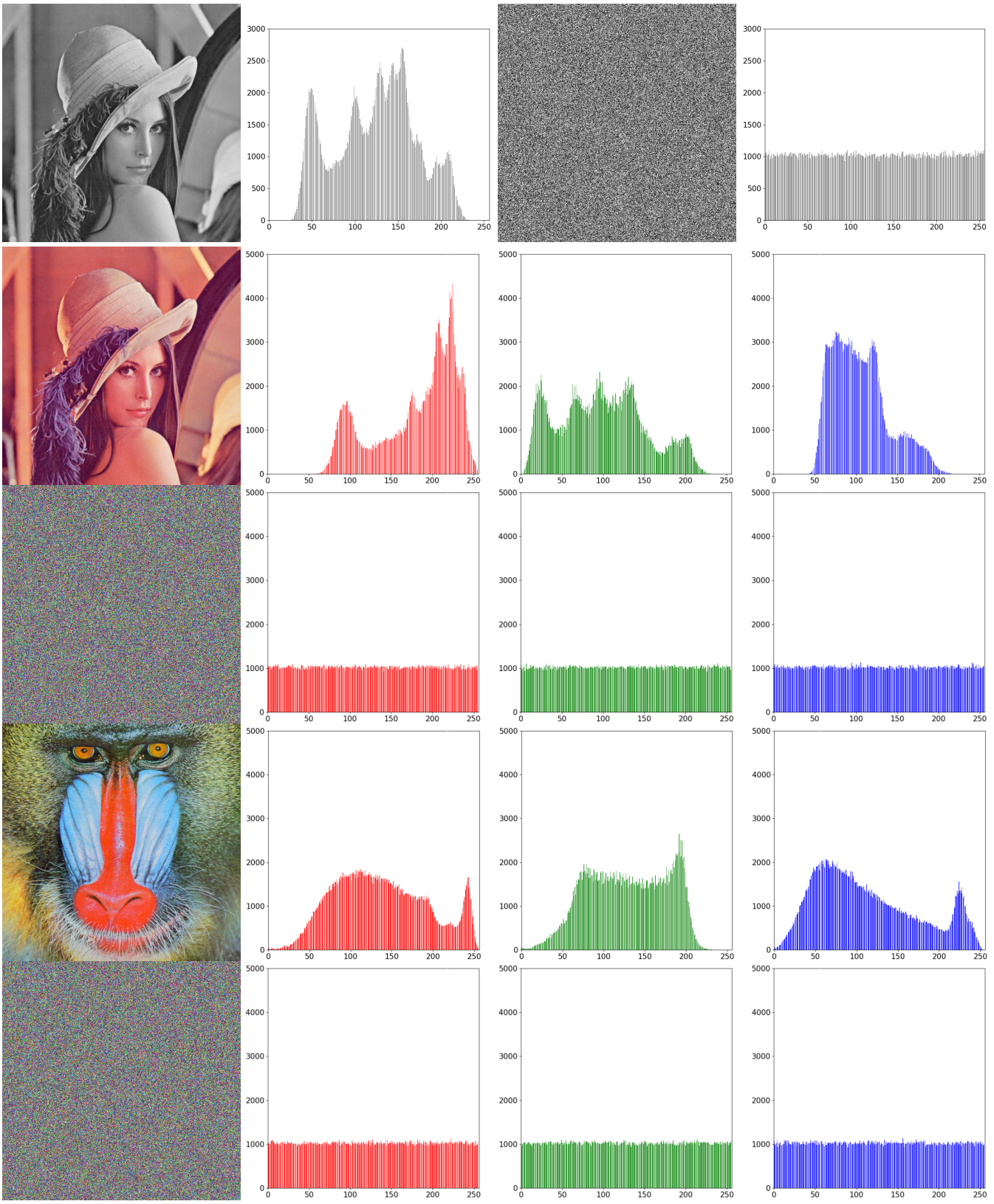

FIGURE 19. Histograms of images and their encrypted images.

is, indicating that the encryption method is more secure. PSNR is defined as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(P_1(i,j) - P_2(i,j))^2 \quad (1)$$

$$PSNR = 10 \times \log_{10}\frac{(2^n-1)^2}{MSE} \quad (2)$$

where P_1 and P_2 represent two images; M and N are the width and height of the image; i and j correspond to the pixel's position in the image; n indicates the number of bits per pixel.

We analyzed PSNR on the standard USC-SIPI image database [34] and Lena image. The results of the PSNR experiments are shown in Table 3.

TABLE 3. The PSNR between original image and encrypted image.

Image	Gray image			Color image					
	5.1.12	Lena	5.3.02	4.1.04			Lena		
	256^2	512^2	1024^2	256^2			512^2		
				R	G	B	R	G	B
[35]	7.30	**9.22**	**8.73**	8.42	**8.51**	9.68	7.86	8.57	**9.61**
Proposed	**7.285**	9.232	8.736	**8.413**	8.528	**9.669**	**7.859**	**8.555**	9.612

In our method, more than half of the PSNR values between original images and encrypted images are lower than the values from [35]. Thus, the encryption quality of our proposed method is similar or even slightly higher.

D. CORRELATION ANALYSIS

Images containing information are usually highly redundant, so adjacent pixels are usually highly correlated. An effective encryption method should have the ability to break these correlations. The correlation coefficient is defined to measure correlation. Correlation coefficient increases as the correlation increases so that the secure encryption method will result in the correlation coefficient of the encrypted image being closer to 0. The correlation coefficient is defined as:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)D(y)}} \quad (3)$$

$$cov(x,y) = E\{[x - E(x)][y - E(y)]\} \quad (4)$$

where $E()$ represents the average and $D()$ represents the standard deviation value of image pixels.

In this paper, horizontally, vertically and diagonally adjacent coefficients are chosen to map the correlation distribution for gray and color Lena images in Fig. 20. Our proposed method breaks the positive correlation of the original image pixels and presents random distribution in the encrypted image.

We compare the correlation coefficient of the encrypted Lena images obtained by our proposed method and other methods in Table 4. Our method can generate encrypted Lena images which have considerably lower correlation coefficient compared to that of the images encrypted by the methods in [24], [25], [35], [36], [37], [40], [41], [43], [46].

TABLE 4. Comparison of the correlation of the encrypted images for gray and color Lena images with dimension 512^2.

Image	Method		Correlation		
			Horizontal	Vertical	Diagonal
Gray Lena	[23]		0.0009	-0.0013	0.0097
	[24]		-0.0005	0.0034	-0.0075
	[25]		0.0021	0.0125	0.1170
	[35]		0.0004	0.0248	0.0034
	[36]		0.0153	0.0139	0.0140
	[37]		0.0141	0.0296	0.0054
	[38]		-0.0016	-0.0006	-0.0052
	[39]		**-0.0003**	**0.0005**	**-0.0005**
	[40]		-0.0021	-0.0040	-0.0051
	[41]		0.0335	-0.0295	-0.0174
	[42]		-0.0005	-0.0007	0.0012
	[43]		-0.0059	0.0211	-0.0146
	Proposed		-0.0004	0.0018	0.0016
Color Lena	[35]	R	0.0017	0.0094	0.0035
		G	0.0130	0.0014	0.0022
		B	0.0079	0.0008	0.0026
	[38]	R	-0.0009	-0.0044	-0.0037
		G	-0.0015	**-0.0002**	0.0025
		B	-0.0008	-0.0006	0.0008
	[44]	R	0.0066	0.0065	0.0308
		G	0.0291	0.0191	0.0140
		B	0.0052	**0.0003**	0.0056
	[45]	R	0.0092	0.0203	0.0073
		G	**0.0002**	0.0025	0.0131
		B	0.0076	0.0006	0.0111
	[46]	R	-0.0031	-0.0060	0.0055
		G	-0.0067	0.0127	-0.0048
		B	-0.0005	-0.0041	-0.0016
	Proposed	R	**0.0002**	**-0.0002**	**0.0004**
		G	0.0007	0.0005	0.0011
		B	**-0.0000**	-0.0017	**-0.0000**

To further analyze the correlation, we perform more experiments on other images including USC-SIPI database [34] and compare our proposed method with [38], which has low correlation coefficient in both gray and color encrypted Lena image. The averages of absolute values of correlation coefficients for 27 gray encrypted images and 10 color encrypted images are shown in Table 5. The method we proposed has 50.29% (86/171) correlation coefficient better than that in [38] as shown in Table 6.

TABLE 5. Averages of absolute values of correlation coefficients for 27 gray and 10 color encrypted images from USC-SIPI database.

Image		[38]			Proposed		
		Hori.	Ver.	Diag.	Hori.	Ver.	Diag.
Gray		**0.0017**	**0.0015**	0.0017	0.0021	0.0017	0.0017
Color	R	0.0036	**0.0022**	0.0042	**0.0012**	0.0033	**0.0021**
	G	**0.0019**	0.0029	0.0025	0.0020	**0.0021**	**0.0019**
	B	0.0026	0.0028	0.0039	**0.0014**	0.0028	**0.0014**

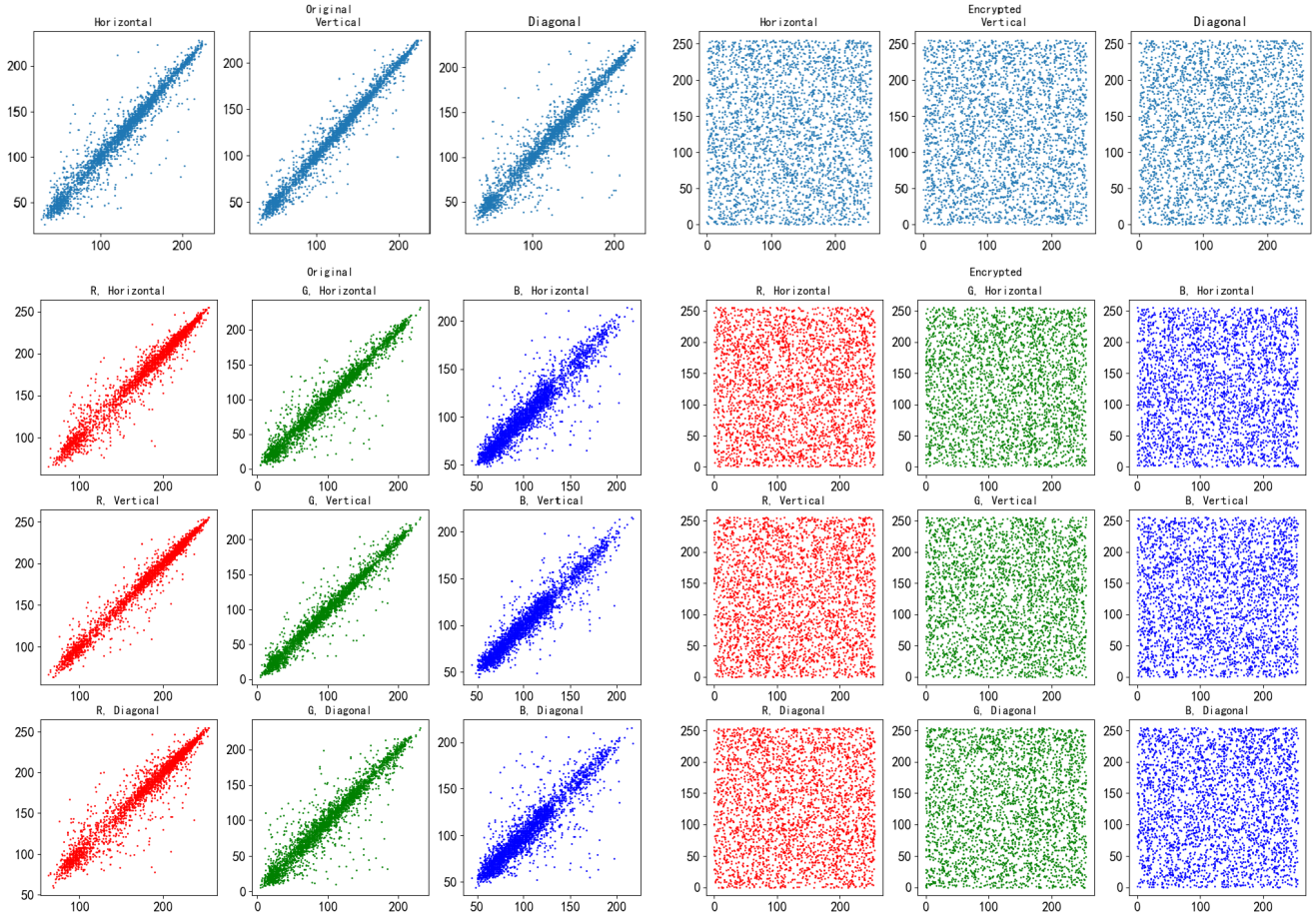

FIGURE 20. Horizontally, vertically and diagonally adjacent coefficient distribution for original gray Lena image (the first row), original color Lena image (the last three rows) and their encrypted images.

E. DIFFERENTIAL ATTACK ANALYSIS

Differential attacks analyze pixel changes in ciphertext by changing specific pixels in the plaintext. If an encryption algorithm has sufficient diffusion such that a significantly different cipher image will be generated with only one pixel change in the original image, then this algorithm can effectively resist differential attacks. The NPCR and UACI are utilized to assess the diffusion of the encryption method. The mathematical expressions for NPCR and UACI are

$$NPCR = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} P(i,j) \times 100\% \tag{5}$$

$$UACI = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{2^n - 1} \times 100\% \tag{6}$$

where C_1 and C_2 represent two encrypted images; M and N are the width and height of the image; i and j correspond to the pixel's position in the image; n indicates the number of bits per pixel.

To increase the obfuscation of the encryption method proposed in this paper, we add an additional module after the random mask step. In this module, the number of bits with value one in the input group is first accumulated. Then the accumulated value performs modulus on the group size to get a number s. Eventually, the first s bits of data in the input group, keeping the same order, are moved to the last s bits.

To accurately calculate NPCR and UACI, we refer to [35]. Firstly, the original image P is encrypted to image C. Then three bits at the beginning, middle and end of P are changed to get P^b, P^m and P^e, respectively. Encrypt these three images to acquire encrypted images C^b, C^m and C^e. Finally, the averages of the NPCR and UACI between C and C^b, C^m, C^e are calculated. We passed all the NPCR and UACI tests on USC-SIPI database [34], as shown in Table 7 and Table 8. Thus, the method we proposed possesses excellent diffusion to resist the differential attack.

VIII. REAL-TIME SPEED ANALYSIS

This section demonstrates the real-time performance of the improved encryption method in terms of the streaming data latency and the maximum data volume that the encryption and decryption components can handle.

TABLE 6. Detailed results of correlation for original images and their encrypted images.

Gray Image	Scale	Original image correlation			Encrypted image correlation [38]			Encrypted image correlation Proposed		
		Hori.	Ver.	Diag.	Hori.	Ver.	Diag.	Hori.	Ver.	Diag.
5.1.09	256^2	0.9020	0.9389	0.9038	-0.0028	-0.0018	0.0039	-0.0059	0.0029	0.0051
5.1.10		0.9050	0.8638	0.8260	0.0025	0.0064	0.0001	**-0.0007**	**0.0055**	-0.0018
5.1.11		0.9571	0.9366	0.8927	0.0039	-0.0003	0.0075	-0.0056	-0.0041	-0.0075
5.1.12		0.9565	0.9740	0.9389	-0.0029	-0.0002	-0.0012	-0.0082	-0.0044	**-0.0002**
5.1.13		0.8722	0.8668	0.7563	-0.0024	-0.0024	0.0027	-0.0060	**-0.0001**	-0.0060
5.1.14		0.9466	0.8992	0.8541	-0.0043	0.0025	0.0010	**0.0005**	**0.0001**	-0.0017
5.2.08	512^2	0.9371	0.8928	0.8562	-0.0032	-0.0024	-0.0044	**-0.0024**	0.0037	**-0.0012**
5.2.09		0.9008	0.8605	0.8035	-0.0013	0.0007	0.0001	**0.0001**	**-0.0001**	0.0033
5.2.10		0.9404	0.9279	0.8981	-0.0011	-0.0001	-0.0026	0.0014	0.0013	**-0.0001**
7.1.01		0.9620	0.9206	0.9075	0.0008	0.0010	-0.0004	0.0015	0.0020	-0.0006
7.1.02		0.9463	0.9459	0.8963	0.0016	-0.0032	-0.0013	-0.0033	**0.0013**	**-0.0004**
7.1.03		0.9456	0.9321	0.9017	0.0024	-0.0008	0.0012	**0.0002**	0.0016	0.0023
7.1.04		0.9768	0.9676	0.9560	-0.0005	0.0005	-0.0022	-0.0008	**-0.0000**	**0.0012**
7.1.05		0.9420	0.9121	0.8934	-0.0017	-0.0002	0.0001	**-0.0005**	-0.0009	0.0002
7.1.06		0.9402	0.9062	0.8861	0.0053	-0.0021	0.0025	**0.0026**	**0.0003**	**0.0002**
7.1.07		0.8862	0.8778	0.8392	-0.0007	0.0050	0.0003	-0.0025	**0.0015**	0.0010
7.1.08		0.9577	0.9293	0.9220	0.0008	0.0028	0.0011	**0.0003**	0.0045	-0.0021
7.1.09		0.9657	0.9305	0.9168	-0.0009	-0.0009	0.0003	0.0027	0.0009	-0.0004
7.1.10		0.9643	0.9474	0.9313	0.0013	-0.0008	0.0031	-0.0015	0.0017	**0.0009**
gray21.512		0.9965	0.9998	0.9964	-0.0025	0.0000	-0.0001	0.0042	0.0004	0.0026
ruler.512		0.4542	0.4643	-0.0290	-0.0012	-0.0019	-0.0024	-0.0014	**0.0000**	**0.0014**
all-black		-	-	-	0.0004	-0.0017	-0.0011	0.0030	-0.0023	**0.0005**
all-white		-	-	-	0.0001	0.0004	0.0005	-0.0013	-0.0004	-0.0006
Lena		0.9761	0.9877	0.9639	-0.0016	-0.0006	0.0052	**-0.0004**	-0.0018	**0.0016**
5.3.01	1024^2	0.9774	0.9813	0.9672	0.0010	0.0013	0.0005	**-0.0004**	-0.0015	**0.0003**
5.3.02		0.9099	0.9034	0.8591	0.0003	0.0002	0.0010	0.0003	0.0012	**-0.0000**
7.2.01		0.9647	0.9470	0.9450	0.0002	0.0007	-0.0010	**0.0001**	-0.0025	-0.0028
Color Image	Channel	Original image correlation			Encrypted image correlation [38]			Encrypted image correlation Proposed		
		Hori.	Ver.	Diag.	Hori.	Ver.	Diag.	Hori.	Ver.	Diag.
4.1.01	R	0.9729	0.9626	0.9488	-0.0070	0.0032	0.0005	**-0.0055**	**0.0020**	0.0024
	G	0.9719	0.9646	0.9499	0.0002	-0.0037	-0.0031	-0.0010	**0.0019**	-0.0080
	B	0.9584	0.9517	0.9376	-0.0071	0.0023	-0.0057	**0.0022**	**0.0013**	**0.0031**
4.1.02	R	0.9493	0.9562	0.9178	0.0058	-0.0019	0.0085	**-0.0033**	0.0075	**0.0006**
	G	0.9308	0.9534	0.9003	-0.0040	-0.0016	-0.0082	**-0.0004**	**-0.0011**	**0.0011**
	B	0.9178	0.9442	0.8890	0.0007	-0.0069	0.0000	-0.0030	**0.0018**	-0.0016
4.1.03	R	0.9779	0.9323	0.9166	-0.0072	0.0007	0.0042	**-0.0003**	-0.0031	-0.0050
	G	0.9748	0.9148	0.8991	-0.0008	0.0046	0.0047	-0.0024	**0.0003**	**-0.0004**
	B	0.9726	0.9168	0.9004	-0.0002	-0.0056	-0.0126	0.0003	**0.0036**	**-0.0001**
4.1.04	R	0.9786	0.9878	0.9683	-0.0053	0.0023	-0.0085	**-0.0008**	0.0063	**-0.0019**
	G	0.9660	0.9820	0.9509	-0.0021	0.0060	0.0000	-0.0077	**-0.0034**	-0.0017
	B	0.9523	0.9717	0.9309	-0.0024	-0.0014	-0.0057	0.0026	-0.0077	**-0.0006**
4.1.05	R	0.9671	0.9354	0.9129	0.0004	-0.0037	0.0058	**-0.0002**	0.0039	**-0.0029**
	G	0.9805	0.9475	0.9322	-0.0011	0.0001	0.0016	**0.0001**	-0.0045	0.0017
	B	0.9820	0.9750	0.9626	-0.0073	-0.0030	-0.0015	**0.0018**	0.0039	0.0034
4.1.07	R	0.9745	0.9763	0.9538	-0.0063	0.0030	-0.0049	**0.0001**	-0.0063	**0.0046**
	G	0.9757	0.9801	0.9603	0.0052	-0.0058	-0.0006	**-0.0042**	**-0.0054**	0.0024
	B	0.9890	0.9880	0.9799	-0.0006	0.0037	0.0049	0.0015	-0.0086	**0.0013**
Lena	R	0.9753	0.9869	0.9636	-0.0009	-0.0044	-0.0037	**0.0002**	**-0.0002**	**0.0004**
	G	0.9745	0.9869	0.9627	-0.0015	-0.0002	0.0025	**0.0007**	0.0005	**0.0011**
	B	0.9530	0.9736	0.9334	-0.0008	-0.0006	0.0008	**-0.0000**	-0.0017	**-0.0000**
4.2.01	R	0.9936	0.9951	0.9894	0.0014	-0.0010	0.0027	-0.0031	**0.0001**	**-0.0008**
	G	0.9812	0.9872	0.9712	-0.0019	-0.0005	-0.0021	-0.0021	0.0018	**-0.0001**
	B	0.9826	0.9791	0.9652	0.0056	-0.0014	-0.0068	**0.0008**	**-0.0009**	**0.0010**
4.2.05	R	0.9726	0.9575	0.9354	-0.0008	0.0012	0.0023	-0.0016	-0.0033	**0.0021**
	G	0.9578	0.9678	0.9327	-0.0021	0.0034	0.0000	**-0.0002**	**0.0001**	0.0001
	B	0.9640	0.9376	0.9176	0.0012	-0.0003	-0.0004	**0.0001**	**-0.0001**	-0.0015
4.2.06	R	0.9558	0.9540	0.9419	-0.0004	-0.0005	0.0005	**0.0002**	**-0.0004**	**0.0002**
	G	0.9715	0.9662	0.9530	-0.0005	0.0029	0.0017	-0.0016	**0.0016**	0.0023
	B	0.9710	0.9693	0.9529	0.0003	0.0024	-0.0005	-0.0018	**-0.0020**	**0.0018**

TABLE 7. Results of NPCR and UACI tests for gray images.

Image	NPCR				UACI			
	[35]	[40]	[47]	Proposed	[35]	[40]	[47]	Proposed
256^2	$N^*_{0.05} = 99.5693\%$				$U^{*-}_{0.05}, U^{*+}_{0.05} = (33.2824\%, 33.6447\%)$			
5.1.09	99.5834	99.5975	99.6459	99.6109	33.5114	33.2879	33.4138	33.4799
5.1.10	99.6261	99.6184	99.6093	99.6060	33.4874	33.5309	33.4354	33.4538
5.1.11	99.5962	99.6064	99.5955	99.6049	33.5485	33.4305	33.5708	33.4752
5.1.12	99.6190	99.6661	99.6002	99.6131	33.5031	33.4458	33.4597	33.5004
5.1.13	99.6343	99.6005	99.6215	99.6100	33.4142	33.4509	33.4388	33.5231
5.1.14	99.6409	99.5915	99.6200	99.6047	33.5586	33.5280	33.4156	33.4755
512^2	$N^*_{0.05} = 99.5893\%$				$U^{*-}_{0.05}, U^{*+}_{0.05} = (33.3730\%, 33.5541\%)$			
5.2.08	99.5968	99.6128	99.6276	99.6105	33.4694	33.4922	33.4751	33.4681
5.2.09	99.6300	99.6022	99.6307	99.6125	33.4914	33.5279	33.5138	33.4834
5.2.10	99.6094	99.6271	99.6196	99.6092	33.4863	33.4087	33.4536	33.4597
7.1.01	99.6042	99.6158	99.5990	99.6101	33.5463	33.4945	33.4556	33.4704
7.1.02	99.6173	99.6316	99.6147	99.6076	33.4669	33.5126	33.5024	33.4503
7.1.03	99.6072	99.5973	99.6089	99.6100	33.4933	33.4546	33.4430	33.4770
7.1.04	99.6127	99.6075	99.6055	99.6095	33.4525	33.5024	33.4209	33.4715
7.1.05	99.5946	99.6237	99.6139	99.6056	33.4745	33.4838	33.4585	33.4526
7.1.06	99.6101	99.6173	99.6154	99.6146	33.4564	33.4615	33.4277	33.4486
7.1.07	99.5988	99.6429	99.6101	99.6115	33.4644	33.5115	33.4080	33.4666
7.1.08	99.6084	99.6290	99.6265	99.6102	33.5041	33.4534	33.4080	33.4546
7.1.09	99.6068	99.6105	99.6196	99.6089	33.5335	33.4140	33.4301	33.4471
7.1.10	99.6025	99.6154	99.6047	99.6133	33.4455	33.4766	33.4168	33.4527
boat.512	99.6085	99.5954	99.6067	99.6128	33.4861	33.4625	33.4011	33.4745
gray21.512	99.6131	99.6041	99.6044	99.6070	33.4517	33.5159	33.4788	33.4669
ruler.512	99.6117	99.5936	99.5937	99.6098	33.4989	33.4415	33.4093	33.4627
1024^2	$N^*_{0.05} = 99.5994\%$				$U^{*-}_{0.05}, U^{*+}_{0.05} = (33.4183\%, 33.5088\%)$			
5.3.01	99.6097	99.6012	99.6024	99.6071	33.4663	33.4775	33.4105	33.4676
5.3.02	99.6102	99.6005	99.6034	99.6117	33.4595	33.4993	33.4621	33.4604
7.2.01	99.6079	99.6099	99.6171	99.6014	33.4776	33.4651	33.4744	33.4673
Pass/All	25/25	25/25	25/25	25/25	25/25	25/25	24/25	25/25

TABLE 8. Results of NPCR and UACI tests for color images.

Image	Channel	NPCR		UACI	
		[35]	Proposed	[35]	Proposed
256^2		$N^*_{0.05} = 99.5693\%$		$U^{*-}_{0.05}, U^{*+}_{0.05} = (33.2824\%, 33.6447\%)$	
4.1.03	R	99.5987	99.6079	33.5078	33.4351
	G	99.6180	99.6082	33.3111	33.4431
	B	99.6114	99.6100	33.3840	33.4474
4.1.06	R	99.6104	99.6103	33.3918	33.4842
	G	99.6295	99.6072	33.4685	33.4591
	B	99.6145	99.6103	33.3752	33.4451
512^2		$N^*_{0.05} = 99.5893\%$		$U^{*-}_{0.05}, U^{*+}_{0.05} = (33.3730\%, 33.5541\%)$	
4.2.03	R	99.6134	99.6067	33.4602	33.4597
	G	99.6152	99.6101	33.4590	33.4512
	B	99.6068	99.6108	33.5150	33.4708
Lena	R	99.6142	99.6099	33.4272	33.4631
	G	99.6115	99.6101	33.4740	33.4559
	B	99.6184	99.6138	33.4717	33.4409
Pass/All		12/12	12/12	12/12	12/12

According to the reorganization scheme, after spending 33 clock cycles on caching, the reorganization groups are sequentially sent to the encryption component by pipelining. The 16-quit quaternary logic operator process in parallel,

so the encryption part takes only 1 clock cycle. The entire video streaming encryption process is completed with only 34 clock cycles of latency. Similarly, decrypting the ciphertext takes 34 clock cycles. The video streaming encryption and decryption process has a total delay of 68 clock cycles.

By reporting implementation timing summary in Vivado software, it can be found that the path delay from the plaintext input to the ciphertext output is 4.793 ns, as show in Fig. 21. The minimum latency for 68 clock cycles is $68 \times 4.793 \times 10^{-9}s \approx 3.25 \times 10^{-7}s$. The maximum amount of data that can be processed by a 16-quit quaternary logic operator is $32 \div (4.793 \times 10^{-9})bit/s \approx 6.676 \times 10^{9}bit/s$>$6.21Gbit/s$. The experimental system works properly if we set the clock period to 5 ns, which is a frequency of 200 MHz.

Summary

Name	Path 2
Slack	∞ns
Source	P[0] (input port)
Destination	C[0] (output port)
Path Group	(none)
Path Type	Max at Slow Process Corner
Requirement	∞ns
Data Path Delay	4.793ns (logic 1.654ns (34.508%) route 3.139ns (65.492%))
Logic Levels	4 (IBUFCTRL=1 INBUF=1 LUT6=1 OBUF=1)

FIGURE 21. Reporting timing summary for encryption.

The improved encryption method latency and processing capability can meet the requirements of most real-time encryption and decryption scenarios. Let's take the camera AN5641 [48] of the experimental system in this paper as an example, which has a built-in OV5640 CMOS image sensor. According to the OV5640 parameter documentation [49], we can calculate that the OV5640 captures $1920 \times 1080 \times 30pixel/s \approx 6.221 \times 10^{7}pixel/s$ at 1080p resolution. The experimental system configured the camera AN5641 for RAW10 output, which uses 10-bit data to represent each pixel. As a result, the amount of data captured by the experimental system's camera was calculated to be $6.221 \times 10^{8}bit/s$, which is an order of magnitude less than the processing capacity of the 16-quit quaternary logic operator.

IX. CONCLUSION

In this paper, we have addressed the problem of key derivation convergence when the original MVL encryption method is applied to video streaming encryption/decryption scenarios. With the goal of improving video and image ciphertext quality, we have combined two improvement strategies, namely byte reorganization and random mask, so that the improved MVL encryption method can be well applied to real-time video streaming data.

To test the proposed method, we have implemented a 16-quit reconfigurable quaternary logic operator on FPGA, based on which we have designed and implemented a real-time video streaming encryption experimental system. Either the encryption or decryption process only takes 34 clock cycles, while the system is capable of processing video streaming data at a speed of 6.21 Gbit/s, showing that MVL operators with parallel computing feature can meet the real-time processing requirements when encrypting video stream data. The experimental results demonstrate the elimination of image edges in the ciphertext, meanwhile show the superiority of the proposed method over other image encryption methods.

The MVL encryption method introduced in this paper focuses on video streaming data encryption scenarios with the aim of increasing the degree of chaos. Theoretically, various obfuscation and diffusion techniques that currently exist in block ciphers can be superimposed on the MVL encryption method. The necessity and feasibility of adding more obfuscation and diffusion techniques to the MVL encryption method will be investigated for future work.

REFERENCES

[1] J. Daemen and V. Rijmen, *The Design of RijndaeL: AES-The Advanced Encryption Standard*. Berlin, Germany: Springer, 2002.

[2] ETSI SAGE. (Jun. 28, 2011). *Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification*. [Online]. Available: https://www.gsma.com/aboutus/wp-content/uploads/2014/12/eea3eia3zucv16.pdf

[3] G. S. Vernam, "Secret signaling system," U.S. Patent 1 310 719, Jul. 22, 1919.

[4] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[5] R. Hasimoto-Beltran, M. D. Calderon-Calderon, and V. H. Olavarría-Jaramillo, "Secure real-time chaotic partial encryption of entropy-coded multimedia information for mobile devices: Smartphones," *IEEE Access*, vol. 10, pp. 15876–15890, 2022.

[6] A. Hafsa, M. Fradi, A. Sghaier, J. Malek, and M. Machhout, "Real-time video security system using chaos-improved advanced encryption standard (IAES)," *Multimedia Tools Appl.*, vol. 81, no. 2, pp. 2275–2298, Jan. 2022.

[7] A. Vazquez-Salazar and A. Ahmadinia, "Partially homomorphic encryption scheme for real-time image stream," in *Proc. 8th IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)/7th IEEE Int. Conf. Edge Comput. Scalable Cloud (EdgeCom)*, Washington, DC, USA, Jun. 2021, pp. 71–76.

[8] H. Cai, J.-Y. Sun, Z.-B. Gao, and H. Zhang, "A novel multi-wing chaotic system with FPGA implementation and application in image encryption," *J. Real-Time Image Process.*, vol. 19, no. 4, pp. 775–790, Aug. 2022.

[9] H. Wang, Y. Jin, S. Jin, and Y. Wang, "An encryption and decryption method, device and communication system thereof," China Patent 10 801 365, Jul. 15, 2021.

[10] V. Gaudet, "A survey and tutorial on contemporary aspects of multiple-valued logic and its application to microelectronic circuits," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 1, pp. 5–12, Mar. 2016.

[11] Z. T. Sandhie, J. A. Patel, F. U. Ahmed, and M. H. Chowdhury, "Investigation of multiple-valued logic technologies for beyond-binary era," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–30, Jan. 2021.

[12] N. Homma, Y. Watanabe, T. Aoki, and T. Higuchi, "Formal design of arithmetic circuits based on arithmetic description language," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. 89, no. 12, pp. 3500–3509, Dec. 2006.

[13] Y. Watanabe, N. Homma, K. Degawa, T. Aoki, and T. Higuchi, "High-level design of multiple-valued arithmetic circuits based on arithmetic description language," in *Proc. 38th Int. Symp. Multiple Valued Log. (ISMVL)*, May 2008, pp. 112–117.

[14] N. P. Brusentsov and J. R. Alvarez, "The setun and the setun 70," in *Proc. IFIP Conf. Perspect. Sov. Russian Comput.*, Berlin, Germany, 2006, pp. 74–80.

[15] Y. Jin, Z. Wang, Y. Liu, S. Ouyang, Y. Shen, and J. Peng, "Ternary optical computer," *Chin. J. Nature*, vol. 41, no. 3, pp. 207–218, Jun. 2019.

[16] Y. Jin, S. Ouyang, Z. Pan, Y. Wang, Y. Shen, J. Peng, S. Zhou, Y. Liu, and X. Chen, "Many-bit, groupable, reconfigurable multiple-valued, electronic operator and its construction method," China Patent 11 567 284, Dec. 20, 2018.
[17] S. Han, "Data encryption and decryption system using multiple-valued logic array," *Chin. J. Comput.*, vol. 16, no. 6, pp. 459–463, Jun. 1993.
[18] A. Sokolov and O. Zhdanov, "Prospects for the application of many-valued logic functions in cryptography," in *Advances in Computer Science for Engineering and Education*, vol. 13. Berlin, Germany: Springer, 2018, pp. 331–339.
[19] M. A. Ali, E. Ali, M. A. Habib, M. Nadim, T. Kusaka, and Y. Nogami, "Pseudo random ternary sequence and its autocorrelation property over finite field," *Int. J. Comput. Netw. Inf. Secur.*, vol. 9, no. 9, pp. 54–63, Sep. 2017.
[20] A. Y. Bykovsky, "A multiple-valued logic for implementing a random Oracle and the position-based cryptography," *J. Russian Laser Res.*, vol. 40, no. 2, pp. 173–183, Apr. 2019.
[21] Q. Dai, X. Zou, and Z. Luo, "Cracking a data encryption and decryption system using multi-valued logic array," *Chin. J. Comput.*, vol. 24, no. 6, pp. 654–656, Jun. 2001.
[22] B. Singh, G. Athithan, and R. Pillai, "On extensions of the one-time-pad," Sci. Anal. Group, Centre Artif. Intell. Robotic, DRDO, Bengaluru, India, Int. Assoc. Cryptol. Res. (IACR), Cryptol. ePrint Arch., Paper 2021/298, 2021. [Online]. Available: https://eprint.iacr.org/2021/298
[23] U. Hayat and N. A. Azam, "A novel image encryption scheme based on an elliptic curve," *Signal Process.*, vol. 155, pp. 391–402, Feb. 2019, doi: 10.1016/j.sigpro.2018.10.011.
[24] Z. Hua, J. Li, Y. Chen, and S. Yi, "Design and application of an S-box using complete Latin square," *Nonlinear Dyn.*, vol. 104, no. 1, pp. 807–825, Mar. 2021, doi: 10.1007/s11071-021-06308-3.
[25] S. Ibrahim and A. Alharbi, "Efficient image encryption scheme using Henon map, dynamic S-boxes and elliptic curve cryptography," *IEEE Access*, vol. 8, pp. 194289–194302, 2020, doi: 10.1109/ACCESS.2020.3032403.
[26] F. Artuger and F. Özkaynak, "An effective method to improve nonlinearity value of substitution boxes based on random selection," *Inf. Sci.*, vol. 576, pp. 577–588, Oct. 2021, doi: 10.1016/j.ins.2021.07.036.
[27] N. A. Azam, U. Hayat, and M. Ayub, "A substitution box generator, its analysis, and applications in image encryption," *Signal Process.*, vol. 187, Oct. 2021, Art. no. 108144, doi: 10.1016/j.sigpro.2021.108144.
[28] S. Ibrahim and A. M. Abbas, "Efficient key-dependent dynamic S-boxes based on permutated elliptic curves," *Inf. Sci.*, vol. 558, pp. 246–264, May 2021, doi: 10.1016/j.ins.2021.01.014.
[29] Y. Wang, Z. Zhang, L. Y. Zhang, J. Feng, J. Gao, and P. Lei, "A genetic algorithm for constructing bijective substitution boxes with high nonlinearity," *Inf. Sci.*, vol. 523, pp. 152–166, Jun. 2020, doi: 10.1016/j.ins.2020.03.025.
[30] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. New York, NY, USA: Springer, 2010.
[31] AMD XILINX. (Mar. 16, 2023). *MIPI CSI-2 Receiver Subsystem Product Guide (PG232)*. [Online]. Available: https://docs.xilinx.com/r/en-U.S./pg232-mipi-csi2-rx
[32] AMD XILINX. (Mar. 16, 2023). *AMD XILINX. Sensor Demosaic LogiCORE IP Product Guide (PG286)*. [Online]. Available: https://docs.xilinx.com/r/en-U.S./pg286-v-demosaic
[33] AMD XILINX. (Mar. 17, 2023). *AMD XILINX. Sensor Demosaic LogiCORE IP Product Guide (PG286)*. [Online]. Available: https://docs.xilinx.com/r/en-U.S./pg285-v-gamma-lut/Common-Interface-Signals
[34] *The USC-SIPI Image Database*. University of Southern California. Accessed: Oct. 15, 2023. [Online]. Available: https://sipi.usc.edu/database
[35] Y. Dong, G. Zhao, Y. Ma, Z. Pan, and R. Wu, "A novel image encryption scheme based on pseudo-random coupled map lattices with hybrid elementary cellular automata," *Inf. Sci.*, vol. 593, pp. 121–154, May 2022, doi: 10.1016/j.ins.2022.01.031.
[36] A. Shakiba, "A novel randomized one-dimensional chaotic Chebyshev mapping for chosen plaintext attack secure image encryption with a novel chaotic breadth first traversal," *Multimedia Tools Appl.*, vol. 78, no. 24, pp. 34773–34799, Dec. 2019, doi: 10.1007/s11042-019-08071-5.
[37] J. Gayathri and S. Subashini, "An efficient spatiotemporal chaotic image cipher with an improved scrambling algorithm driven by dynamic diffusion phase," *Inf. Sci.*, vol. 489, pp. 227–254, Jul. 2019, doi: 10.1016/j.ins.2019.01.082.
[38] N. A. Azam, J. Zhu, U. Hayat, and A. Shurbevski, "Towards provably secure asymmetric image encryption schemes," *Inf. Sci.*, vol. 631, pp. 164–184, Jun. 2023, doi: 10.1016/j.ins.2023.02.057.
[39] N. A. Azam, I. Ullah, and U. Hayat, "A fast and secure public-key image encryption scheme based on mordell elliptic curves," *Opt. Lasers Eng.*, vol. 137, Feb. 2021, Art. no. 106371, doi: 10.1016/j.optlaseng.2020.106371.
[40] A. Mansouri and X. Wang, "A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme," *Inf. Sci.*, vol. 520, pp. 46–62, May 2020, doi: 10.1016/j.ins.2020.02.008.
[41] J. Tang, Z. Yu, and L. Liu, "A delay coupling method to reduce the dynamical degradation of digital chaotic maps and its application for image encryption," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 24765–24788, May 2019, doi: 10.1007/s11042-019-7602-8.
[42] M. Wang, X. Wang, T. Zhao, C. Zhang, Z. Xia, and N. Yao, "Spatiotemporal chaos in improved cross coupled map lattice and its application in a bit-level image encryption scheme," *Inf. Sci.*, vol. 544, pp. 1–24, Jan. 2021, doi: 10.1016/j.ins.2020.07.051.
[43] X. Wang, L. Feng, R. Li, and F. Zhang, "A fast image encryption algorithm based on non-adjacent dynamically coupled map lattice model," *Nonlinear Dyn.*, vol. 95, pp. 2797–2824, Mar. 2019, doi: 10.1007/s11071-018-4723-y.
[44] D. Herbadji, A. Belmeguenai, N. Derouiche, and H. Liu, "Colour image encryption scheme based on enhanced quadratic chaotic map," *IET Image Process.*, vol. 14, no. 1, pp. 40–52, Jan. 2020, doi: 10.1049/iet-ipr.2019.0123.
[45] K. Xuejing and G. Zihui, "A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system," *Signal Process., Image Commun.*, vol. 80, Feb. 2020, Art. no. 115670, doi: 10.1016/j.image.2019.115670.
[46] D. S. Laiphrakpam and M. S. Khumanthem, "A robust image encryption scheme based on chaotic system and elliptic curve over finite field," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8629–8652, Apr. 2018, doi: 10.1007/s11042-017-4755-1.
[47] X. Wang and J. Yang, "A privacy image encryption algorithm based on piecewise coupled map lattice with multi dynamic coupling coefficient," *Inf. Sci.*, vol. 569, pp. 217–240, Aug. 2021, doi: 10.1016/j.ins.2021.04.013.
[48] AMD XILINX. (Apr. 26, 2023). *MIPI Monocular Camera Module AN5641 User Manual*. [Online]. Available: https://www.alinx.com/public/upload/file/AN5641_User_Manual.pdf
[49] OmniVision. *OV5640 Datasheet Product Specification*. Accessed: Dec. 12, 2022. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/OV5640_datasheet.pdf

XINYU ZHOU received the B.E. degree in software engineering from Donghua University, Shanghai, China, in 2021, where he is currently pursuing the M.E. degree with the School of Computer Science and Technology.

His research interest includes the applications of reconfigurable multiple-valued logic processors for cryptographic.

HONGJIAN WANG received the Ph.D. degree in computer science from the University of Technology of Belfort-Montbéliard, France, in 2016.

From 2016 to 2019, he was a Postdoctoral Researcher with Heidelberg University, Germany. Currently, he is an Associate Professor with the School of Computer Science and Technology, Donghua University, Shanghai, China. His research interest includes reconfigurable multiple-valued logic processors and their applications.

KUIYAN LI received the B.E. degree in information security from the Civil Aviation University of China, in 2021. She is currently pursuing the M.E. degree with the School of Computer Science and Technology, Donghua University, Shanghai, China.

Her research interest includes the applications of multiple-valued logics in image encryption.

NINGCHUN MO received the B.E. degree in communication engineering from the Guilin University of Electronic Technology, in 2001, and the M.E. degree in communication engineering from Shanghai University, in 2010.

From 2001 to 2005, she was a Product Designer with Shanghai Meiduo Communication Equipment Company Ltd., where she has been a Project Leader of shortwave communication products, since 2006. She holds five patents. Her research interests include shortwave radio communications and image communications.

LIFENG TANG received the M.E. degree in electrical power system and automation from Hohai University, in 2011.

He has been engaged in the research of satellite, since 2012. His research interests include the remote control and telemetry of satellite and the integrated electronic system of satellite. He is a member of the Remote Telemetry Committee of the China Astronautical Association.

YI JIN received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2003.

Currently, he is a Professor and a Senior Researcher with Shanghai University. His research interests include optical computer and computer architecture.