A CRYPTOGRAPHY SYSTEM AND ITS PARAMETERIZED VLSI GENERATOR FOR REAL-TIME MULTIMEDIA

HUN-CHEN CHEN[1], JUI-CHENG YEN[1], CHIEN-WAN HUN[2] & MING-FUNG HWANG[3]
[1]Department of Electronic Engineering, National United University, Miaoli, Taiwan, R.O.C.
[2]Department of Mechanical Engineering, National United University, Miaoli, Taiwan, R.O.C.
[3]Liberal Arts Center, Da-Yeh University, Changhua, Taiwan, R.O.C.

ABSTRACT

In this paper, a new cryptography system is proposed, which combines the methods of position permutation and value transformation for encryption/decryption. Three good features are involved in this system: (1) High security evaluated with the measure of fractal dimension, (2) Content of encrypted image is sensitive to the initial key, and (3) this system can easily defense against the exhaustive search attack. For the applications with real-time in multimedia system, the parameterized hardware design and its very large-scale integrated circuit (VLSI) generator software are developed. The proposed VLSI generator can be parameterized by the parameters of system-type, packet size, throughput, and security to create proper architecture for the application. All the architectures generated from the VLSI generator have been verified strictly. Except for passing the entire coding guideline check and 100% code coverage, with the 0.18 um cell library, all the configurations of architecture are synthesized and verified for speed, area, and power consumption as well as delivering the essential scripts. Regarding verification of all configurations, the throughput can be ranged between 1.59 and 2.25 Gbps with the hardware cost of 0.54 and 3.92 mm^2. Compared with the existing designs, the proposed design possesses wide range of performance and benefit for most of applications in multimedia system.

Keywords: Decryption, encryption, multimedia system, parameterized VLSI.

1 INTRODUCTION

Nowadays, owing to rapid increase in bandwidth, it is more and more prevalent to transmit multimedia signal over internet. Meanwhile, illegal data access has become more serious. The data security [1–12] has become the critical and imperative issue. The real-time realization of cryptography system for application of multimedia system should be focused on as well as the algorithm development. Some data encryption/decryption algorithms and their hardware designs for real-time have been proposed in the last decade. Based on the SAFER algorithm with 128-bit key, Schubert et al. [5] proposed the reusable cryptographic VLSI core with 251.8 Mbit/s throughputs. Mitsuyama et al. [6] have implemented the high-performance VLSI with burst mode and 128-bit block ciphers for AES. Lin [7] has designed and realized the IP core for TDCEA Encryption/Decryption algorithm for the Video Surveillance System. Lewis et al. [8] proposed the architecture design of application-specific processor for cryptographic system and its VLSI implementation. On the analysis of ciphertext-only attack and computation complexity points of view, in this paper, based on the Shannon product theory, we proposed a multimedia cryptographic system (MCS). MCS consists of four functions of (i) Every N–1 bytes of input data are randomly expanded to N bytes; (ii) Random swapping on the expanded data is made in multi levels; (iii) The eight bit-planes are randomly XORed or XNORed with two random operands; (iv) Two rounds of 2D 64-bit rotation operation. All the involved operations are controlled by a binary sequence being generated from a chaos-based pseudorandom bit generator (PBG). As the computation complexity analysis of ciphertext-only attack with the key lengths of 271 and 283 and the packet sizes of 16 and 32 bytes for decryption, we can see that MCS is hard to be attacked within short period of time.

ISSN: 2041-9031 (paper format), ISSN: 2041-904X (online), http://journals.witpress.com
DOI: 10.2495/SAFE-V2-N1-80-95

Regarding real-time realization of the proposed cryptographic system, the parameterized hardware architecture is designed with the techniques of pipeline and parallel processing, and its VLSI generation software is developed such that the hardware of MCS can be automatically created for the application with required performance. We have verified all the configurations in VerilogHDL created by the VLSI generator with 0.18 um CMOS technology. Each of the configurations has been qualified for coding style and code coverage. The qualification results of all the configurations show no error and few petty warnings under RMM coding guidelines [13] as well as near 100% code coverage. The verification results of high-level synthesis reveal that the throughput can be ranged between 1.59 and 2.25 Gbps with the corresponding hardware cost of 0.54 and 3.92 mm^2. Comparing with the existing designs [5–9], the performance of the proposed design is better than the others in terms of the evaluation index of data rate per area (DRPA). Obviously, with the provided high security, the proposed high-performance VLSI design is easily integrated in most of high-quality multimedia system with the requirement of real-time and reasonable hardware cost.

The rest of paper is organized as follows. In Section 2, we illustrate the proposed cryptographic system in detail; the MATLAB simulation results that are involved. In Section 3, we present the proposed parameterized hardware design of MCS. In Section 4, we show the verification of proposed design with different parameters and comparison with some existing designs. Finally, we conclude this paper in Section 5.

2 THE PROPOSED CRYPTOGRAPHIC SYSTEM

2.1 Multimedia Cryptography System

Based on the Shannon product theory, a new MCS is proposed as Fig. 1. We use two PBGs to generate binary sequences. The binary sequences are used as operands and control signals of the processing elements in MCS. The original MCS under the sequence, every 15 bytes of

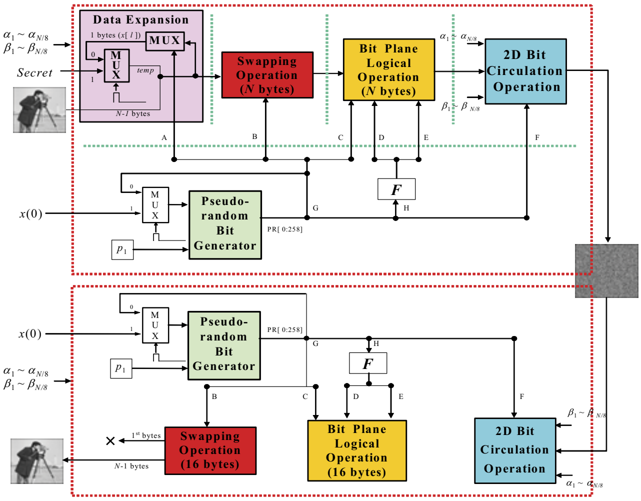

Data Expansion
1 bytes (x[l])
MUX
M U X
0
1
temp
N-1 bytes
$\alpha_1 \sim \alpha_{N/8}$
$\beta_1 \sim \beta_{N/8}$
Secret
Swapping Operation (N bytes)
Bit Plane Logical Operation (N bytes)
2D Bit Circulation Operation
A
B
C
D
E
F
G
H
F
x(0)
Pseudo-random Bit Generator
PR[0:258]
p_1
Swapping Operation (16 bytes)
Bit Plane Logical Operation (16 bytes)
1st bytes
N-1 bytes

Figure 1: The block diagram of MCS.

input data undergo the following subsystems: (i) data expansion to have 16-byte packet, (ii) swapping operation, (iii) bit-plane logical operation, and (iv) 2D bit rotation operation. The detail description for MCS is elaborated in the following.

2.1.1 PBGs in MCS

Kocarev and Jakimoski construct a class of chaos-based PBGs in [14]. We adopt the first PBG with $p = 5$, $m = 2$, $M = 256$, and $k = 2$ as the PBG in MCS.

The initial state of PBG is the 259-bit $x(0)$ and the output is denoted as 259-bit PR[0:258], where 259 comes from M+k+1. The output of PBG PR[0:258] is used to be the operands and the control signal for the MCS operation, where A denotes $PR\left[0:\log_2^N-1\right]$, B denotes $PR\left[\log_2 N:\log_2 N+\frac{N}{2}\log_2 N-1\right]$, C denotes $PR\left[\log_2 N+\frac{N}{2}\log_2 N:\log_2 N+\frac{N}{2}\log_2 N+15\right]$, D denotes $f_{PR}\left[0:N-1\right]$, E denotes $f_{PR}\left[N:2N-1\right]$, F denotes $PR\left[4N+1:8N\right]$, G denotes $PR\left[0:8N\right]$, H denotes $PR\left[0:8N-1\right]$, and I denotes $f_{PR}\left[0:2N-1\right]$, where N denotes the packet size. The operand $f_{\mathrm{PR}}[0{:}\ 2N-1]$ is mapped by PR[0: 4(2N–1)+3)] as $f_{PR}[k] = (\mathrm{PR}[4k] \oplus \mathrm{PR}[4k+1]) \oplus (\mathrm{PR}[4k+2] \oplus \mathrm{PR}[4k+3])$ for $0 \le k \le 2N-1$.

2.1.2 Data expansion in MCS

Every N-1 bytes of input data are extended to N bytes and defined the N bytes as a packet. For first packet on the input of swapping operation stage, the extended byte is the parameter *Secret*. In the following packets, with the corresponding bits of control sequence, the extended byte is randomly chosen from the previous packet.

2.1.3 Swapping operation in MCS

Definition 1: The operation $Swap_w(g(m), g(n))$ is defined to swap $g(m)$ and $g(n)$, if w is equal to 1 or preserve their original positions, if w is equal to 0.

The swapping operation for a 16-byte packet $g(n)$s, $0 \le n \le 15$, contains four levels. They are defined as follows:

#1) $Swap_{\mathrm{PR}[k]}(g[i], g[j])$ for $(i, j, k) \in \{(0, 8, 4), (1, 9, 5), (2, 10, 6), (3, 11, 7), (4, 12, 8), (5, 13, 9), (6, 14, 10), (7, 15, 11)\}$,

#2) $Swap_{\mathrm{PR}[k]}(g[i], g[j])$ for $(i, j, k) \in \{(0, 4, 12), (1, 5, 13), (2, 6, 14), (3, 7, 15), (8, 12, 16), (9, 13, 17), (10, 14, 18), (11, 15, 19)\}$,

#3) $Swap_{\mathrm{PR}[k]}(g[i], g[j])$ for $(i, j, k) \in \{(0, 2, 20), (1, 3, 21), (4, 6, 22), (5, 7, 23), (8, 10, 24), (9, 11, 25), (12, 14, 26), (13, 15, 27)\}$,

#4) $Swap_{\mathrm{PR}[k]}(g[i], g[j])$ for $(i, j, k) \in \{(0, 1, 28), (2, 3,29), (4, 5, 30), (6, 7, 31), (8, 9, 32), (10, 11, 33), (12, 13, 34), (14, 15, 35)\}$.

2.1.4 Bit plane logical operation in MCS

Definition 2: The ith bit plane BP_i, $0 \le i \le 7$ of $g(n)$s, $0 \le n \le 15$ is defined to be the set of all the ith bits of $g(n)$s from least significant bit.

The logical operation on the bit planes is defined as follows:

```
FOR $i$ = 0 TO 7 DO
    Switch (2×PR[36+2$i$] + PR[37+2$i$])
        Case 3: $\mathrm{BP}_i \oplus$ D;
        Case 2: $\mathrm{BP}_i$ XNOR D;
    Case 1: BP$_i$ ⊕ D;
    Case 0: BP$_i$ XNOR E;
End
```

2.1.5 Two-Dimensional bit rotation operation in MCS

Let M be an 8 × 8 binary matrix. Two mappings are defined as the following.

Definition 3: The mapping $RotateX_i^{p_i,r_i} : M \rightarrow M'$ is defined to rotate each bit in the ith row of M, $0 \leq i \leq 7$ in the left direction r_i bits if p_i equals 1 or in the right direction r_i bits if p_i equals 0, where $0 \leq r \leq 7$.

Definition 4: The mapping $RotateY_j^{q_j,s_j} : M \rightarrow M'$ is defined to rotate each bit in the jth column of M, $0 \leq j \leq 7$ in the up direction s_j bits if q_j equals 1 or in the down direction s_j bits if q_j equals 0, where $0 \leq s \leq 7$.

Regarding the first eight bytes of the packet as the 8 × 8 binary matrix M_1 and the second eight bytes as M_2 after the logical operation, this function block in MCS performs the operations of

$$\left(\prod_{j=0}^{7} RotateY_j^{q_j,s_j}\right) \bullet \left(\prod_{i=0}^{7} RotateX_i^{p_i,r_i}\right)(M_1) \text{ and } \left(\prod_{j=0}^{7} RotateY_j^{q_j,s_j}\right) \bullet \left(\prod_{i=0}^{7} RotateX_i^{p_i,r_i}\right)(M_2)$$

sequentially.

For $0 \leq i, j \leq 7$ in M_1, p_i = PR(65+2i), q_j = PR(81+2j), $r_i = \alpha_1 + \beta_1 \times$PR(66+2$i$), and $s_j = \alpha_1 + \beta_1 \times$PR(82+2$j$). For $0 \leq i,j \leq 7$ in M_2, p_i = PR(97+2i), q_j = PR(113+2j), $r_i = \alpha_2 + \beta_2 \times$PR(98+2$i$), and $s_j = \alpha_2 + \beta_2 \times$PR(114+2$j$).

2.1.6 Evaluation of MCS algorithm

We evaluate the proposed MCS algorithm by three respects of MATLAB simulation result of image, initial key sensitivity of MCS, and computation complexity of MCS. Parameters in the evaluation are α1 = 2, β1 = 1, α2 = 3, β2 = 2 and *Secret* = 255 for N = 16 and α1 = 5, β1 = 1, α2 = 1, β2 = 2, α3 = 2, β3 = 2, α4 = 3, β4 = 4, and *Secret* = 255 for N = 32 as well as the 255 bits of $x(0)$.

2.1.6.1 MATLAB simulation result of image

Figure 2 shows the MATLAB simulation results of image encryption and decryption.

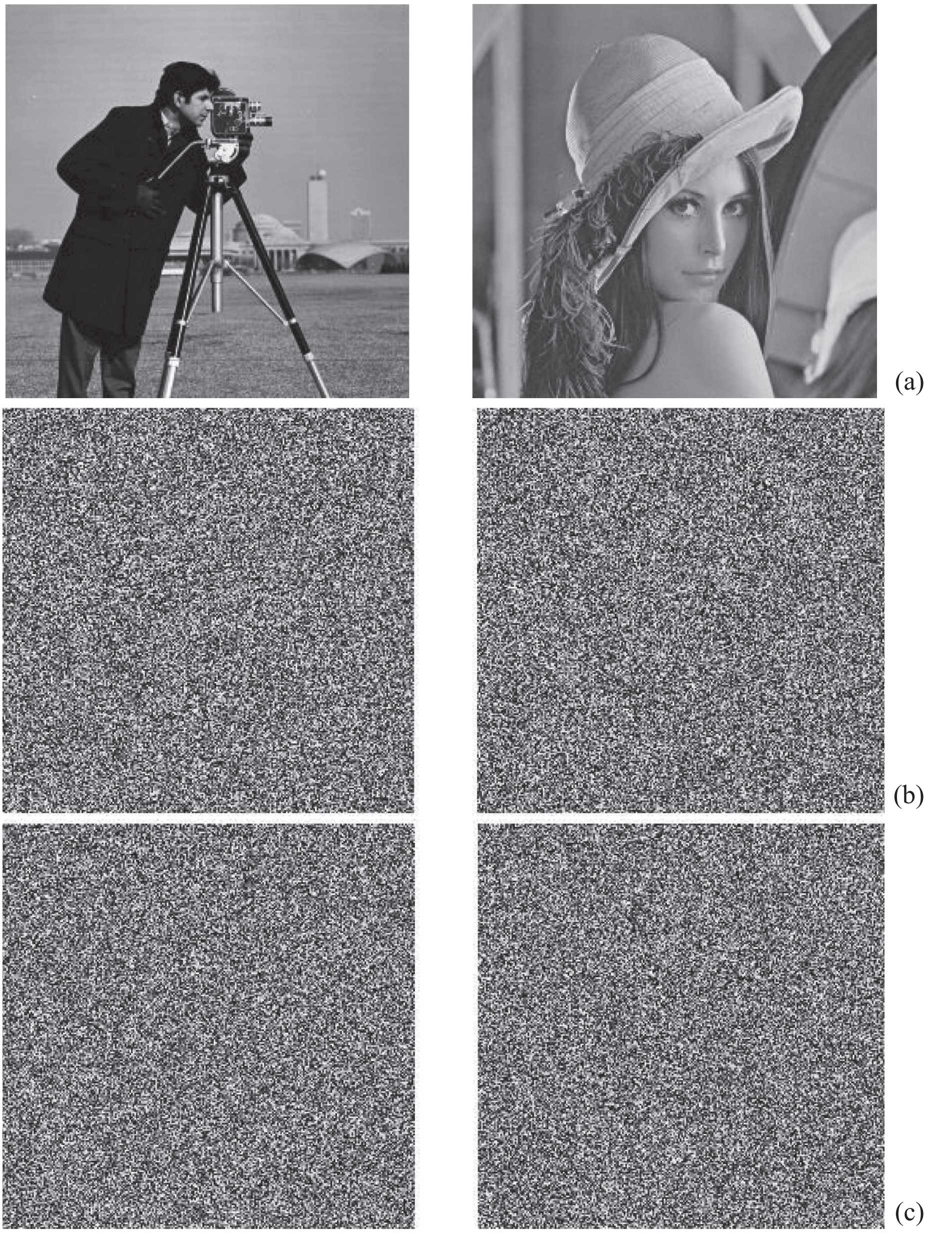
However, since it is not so easy to identify the encrypted results from the images, we have further verified the encrypted results for the security by the metric of fractal dimension [15]. Table 1 shows the fractal dimensions of the different cases of processes in MCS, where SO•DE denotes the process with *Swapping operation* and *Data expansion*, BP•SO•DE denotes the process of SO•DE followed by *Bit-plan logic operation*, BR(H)•BP•SO•DE denotes the process of BP•SO•DE followed by the horizontal one of *2D Bit Rotation Operation*, and BR(V)•BR(H)•BP•SO•DE denotes the process of BR(H)•BP•SO•DE followed by the vertical one of *2D Bit Rotation Operation.* As shown in Table 1, we can see that the fractal dimensions of the encrypted images are ranged between 2.9949412 and 2.9956699. They are much close to the optimal fractal dimension of 3, namely the encrypted data in image are randomly so much to have high security. Moreover, the computation of the fractal dimensions (*fd*s) of encryption under $x(0)$ and x[cj] is listed in Table 2. It shows that the encryption results of MCS are completely disorderly.

2.1.6.2 Initial key sensitivity of MCS

Now, we consider the security problem. We regard MCS being a composite function MCS under the control of PR[0:258]. The MCS is composed of the following four cascaded operations: data expansion operation (*DE*), swapping operation (*SP*), bit-plane logical operation (*BP*), and 2D bit-rotation operation (*BR*). Hence,

$$MCS = BR\ 0\ BP\ 0\ SP\ 0\ DE.$$

So, it belongs to the Shannon product cipher. The input data will be diffused and confused by the randomly cascaded operations under the control signal of PR[0:258]. Moreover, in the



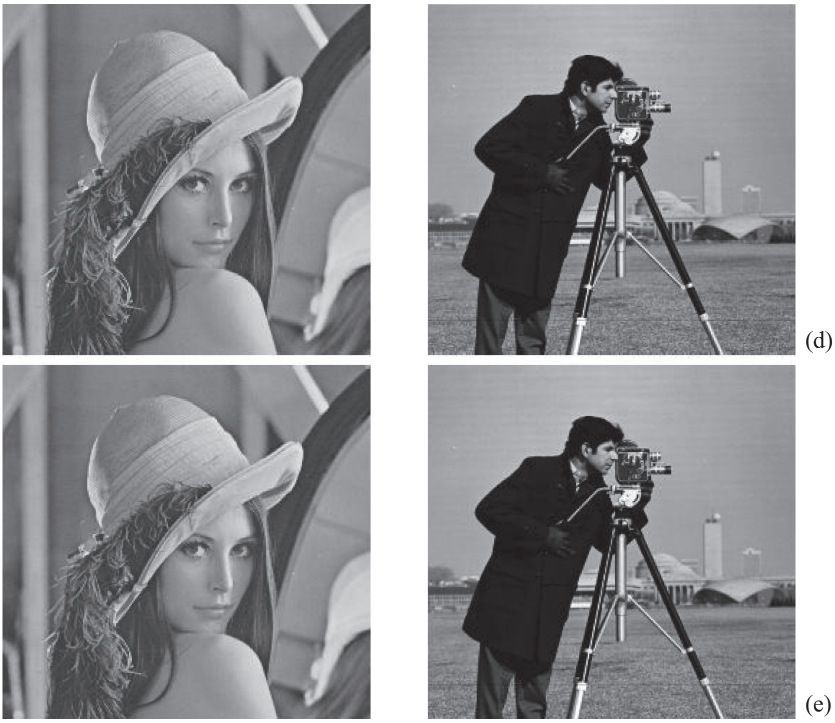


Figure 2: (a) the original images of 'Cman' and 'Lenna', (b) encryption results with $N = 16$, (c) encryption results with $N = 32$, (d) decryption results with $N = 16$, and (e) decryption results with $N = 32$.

Table 1: The fractal dimensions on the different cases of process.

Cman				
	SO•DE	BP•SO•DE	BR(H)•BP•SO•DE	BR(V)•BR(H)•BP•SO•DE
$N = 16$	2.8248303	2.9968482	2.9954209	2.9949412
$N = 32$	2.8625898	2.9912535	2.9947739	2.9952661
Lenna				
	SO•DE	BP•SO•DE	BR(H)•BP•SO•DE	BR(V)•BR(H)•BP•SO•DE
$N = 16$	2.9264195	2.9936732	2.9962255	2.9951462
$N = 32$	2.958629	2.9954241	2.9950501	2.9956699

Table 2: The *fractal dimension*s (*fds*) for the *initial keys of* $x(0)$ and $x[cj]$.

Cman								
N=16	j	$x(0)$	20	50	80	110	140	170	200
	fds	2.99494	2.99581	2.99572	2.99466	2.99489	2.99536	2.99545	2.99542
N=32	j	x(0)	20	50	80	110	140	170	200
	fds	2.99527	2.99488	2.99547	2.99529	2.99547	2.99529	2.99547	2.99529

chaotic systems [16], it is well known that (i) it has sensitive dependence on initial conditions, (ii) the trajectories are dense, bounded, but non-periodic in the state space, and (iii) it has noise-like spectrum. Especially, Kocarev and Jakimoski have proved that the adopted PBG is cryptographically secure.

To demonstrate the parameter sensitivity of MCS by MATLAB simulation, the root mean square difference (***RMSD***) is computed. Let f'_A and f'_B be the encryption results of the image $\boldsymbol{f}$ of size $\boldsymbol{L} \times \boldsymbol{P}$ pixels under $\boldsymbol{x_A}(0)$ and $\boldsymbol{x_B}(0)$, respectively. The ***RMSD*** is defined as

$$RMSD \equiv \left(\frac{1}{L \times P} \sum_{i=0}^{L-1} \sum_{j=0}^{P-1} \left(f'_A(i,j) - f'_B(i,j) \right)^2 \right)^{\frac{1}{2}}.$$

Randomly generate $x(0)$ and take the complement of the jth bit of $x(0)$ denoted as $x[\mathrm{c}j]$ as well as the complement of the ith and jth bits of $x(0)$ denoted as $x[\mathrm{c}ij]$; $0 \le j \le 258$. After applying MCS to ‘Cman’ and ‘Lenna’ under $x(0)$ and $x[\mathrm{c}j]$, the *RMSD*s between the results of $x(0)$ and $x[\mathrm{c}j]$ as well as $x(0)$ and $x[\mathrm{c}ij]$ for encryption are listed as Tables 3 and 4, respectively. Tables 3 and 4 reveal that the encryption results are quite different under 1-bit or 2-bit variation. Moreover, Tables 5 and 6 show the *RMSD* for decryption.

Table 3: The *RMSD*s for the encrypted results with the *initial keys of* $x(0)$ and $x[\mathrm{c}j]$.

Cman									
$N = 16$	*i*	20	50	80	110	140	170	200	230
	RMSD	103.941	104.159	104.430	103.818	104.015	103.939	103.595	103.625
$N = 32$	*i*	20	50	80	110	140	170	200	230
	RMSD	102.276	103.567	104.497	104.166	104.068	103.897	103.892	103.956
Lenna									
$N = 16$	*i*	20	50	80	110	140	170	200	230
	RMSD	104.223	104.664	104.615	104.395	104.424	104.084	103.935	103.593
$N = 32$	*i*	20	50	80	110	140	170	200	230
	RMSD	103.318	104.041	104.211	104.032	104.068	104.044	103.925	103.916

Table 4: The *RMSD*s for the encrypted results with the *initial keys of* $x(0)$ and $x[\mathrm{c}ij]$.

Cman									
N =16	(*i,j*)	(10,130)	(20,140)	(35,155)	(50,170)	(65,185)	(80,200)	(95,215)	(110,230)
	RMSD	104.144	104.177	104.168	104.702	104.211	104.187	104.457	104.262
N =32	(*i,j*)	(10,130)	(20,140)	(35,155)	(50,170)	(65,185)	(80,200)	(95,215)	(110,230)
	RMSD	104.222	103.792	104.361	104.471	104.496	104.197	104.537	104.015
Lenna									
N =16	(*i,j*)	(10,130)	(20,140)	(35,155)	(50,170)	(65,185)	(80,200)	(95,215)	(110,230)
	RMSD	104.411	104.766	104.427	104.500	104.533	104.692	104.163	104.351
N =32	(*i,j*)	(10,130)	(20,140)	(35,155)	(50,170)	(65,185)	(80,200)	(95,215)	(110,230)
	RMSD	104.241	104.092	104.513	104.397	104.870	104.393	104.566	104.279

Table 5: The *RMSD*s for the decrypted results with the *initial keys of* $x(0)$ and $x[cj]$.

Cman									
$N=16$	*i*	20	50	80	110	140	170	200	230
	RMSD	104.573	104.644	104.207	104.647	104.573	104.315	103.979	103.853
$N=32$	*i*	20	50	80	110	140	170	200	230
	RMSD	103.863	104.531	104.609	104.608	104.405	104.017	103.968	104.178
Lenna									
$N=16$	*i*	20	50	80	110	140	170	200	230
	RMSD	104.700	104.729	104.077	104.161	104.456	103.636	103.546	103.847
$N=32$	*i*	20	50	80	110	140	170	200	230
	RMSD	103.803	103.895	104.406	104.920	103.935	104.302	104.002	104.117

Table 6: The *RMSD*s for the decrypted results with the *initial keys of* $x(0)$ and $x[cij]$.

Cman									
$N=16$	(i,j)	(10,130)	(20,140)	(35,155)	(50,170)	(65,185)	(80,200)	(95,215)	(110,230)
	RMSD	104.349	104.589	104.200	104.578	104.484	104.173	104.314	104.533
$N=32$	(i,j)	(10,130)	(20,140)	(35,155)	(50,170)	(65,185)	(80,200)	(95,215)	(110,230)
	RMSD	104.541	104.409	104.404	104.366	104.410	104.879	104.195	104.306
Lenna									
$N=16$	(i,j)	(10,130)	(20,140)	(35,155)	(50,170)	(65,185)	(80,200)	(95,215)	(110,230)
	RMSD	104.322	104.752	104.753	104.588	104.443	104.826	104.362	104.548
$N=32$	(i,j)	(10,130)	(20,140)	(35,155)	(50,170)	(65,185)	(80,200)	(95,215)	(110,230)
	RMSD	104.421	104.482	104.444	104.658	104.424	104.991	104.352	104.390

2.1.6.3 Analysis of computation complexity

We have analyzed the computation complexity for MCS in this sub-section. The packet with the size of *N* is processed by MCS for each sample. Thus, for a computation of packet, $log_2 N \times \frac{N}{2}$ swapping operations, 8*N* XOR/XNOR operations, and 2*N* Barrel-shift operations are needed. We show the operations in more detail and how many times of these operations are performed with the two packet sizes in the encryption algorithm in Table 7. It reveals that the proposed MCS can be a low-cost algorithm.

Regarding the analysis of ciphertext-only attack, as shown in Table 8, due to the key lengths being 271 and 283, respectively, for decryption with the packet sizes of 16 and 32 bytes causes the large amount of computation, we can see that MCS should not be easy to attacked within reasonable duration.

2.2 Parameterized Multimedia Cryptographic System

As shown in Fig. 3, based on the original MCS algorithm above, except for processing type and packet size, the extended MCS algorithm can be configured with various throughput, hardware cost, and security. In the parameterized multimedia cryptographic system (PMCS), the

Table 7: Computation complexity for encrypting a packet.

	MUX			MUL (259	SWAP	XOR/ XNOR	Barrel shifter
Packet size	2/1	16/1	32/1	bits)	(times)	(bits)	(times)
16 bytes	146	1	0	1	32	128	32
32 bytes	306	0	1	1	80	256	64

Table 8: Computation complexity needed by attacking a image frame with size of $M \times M$.

Packet size operation		16 bytes	32 bytes
MUX	2 to 1	$2^{271} \times M^2 \times 9.6667$	$2^{283} \times M^2 \times 9.8387$
	16 to 1	$2^{271} \times M^2 \times 0.06667$	0
	32 to 1	0	$2^{283} \times M^2 \times 0.0322$
MUL (259 bits)		$2^{271} \times M^2 \times 0.06667$	$2^{283} \times M^2 \times 0.032258$
SWAP (times)		$2^{271} \times M^2 \times 2.1333$	$2^{283} \times M^2 \times 2.5806$
XOR/XNOR (bits)		$2^{271} \times M^2 \times 8.5333$	$2^{283} \times M^2 \times 8.258$
Barrel shifter (times)		$2^{271} \times M^2 \times 2.133$	$2^{283} \times M^2 \times 2.064$

```
If Processing_type = Encryption then
    For i = 0 to (#_of_Data -1) do
        Data_Dispatch ( i mod Series)
        Data_Extension (Packet_size - 1)
        For j = 1 to Security do
              Swapping_Operation(Packet_size)
              Bit_Plane_Logical_Operation(Packet_size)
              2D_Bit_Rotation_Operation(Packet_size)
        End
    End
Else
    For i = 0 to (# of Data -1) do
        Data Dispatch ( i mod Series)
        For j = 1 to Security do
              2D_Bit_Rotation_Operation (Packet_size)
              Bit_Plane_Logical_Operation(Packet_size)
              Swapping_Operation(Packet_size)
        End
    End
```

Figure 3: Representation of the parameterized MCS algorithm.

parameter of *Processing_type* is used to determine the type of PMCS, that is, encryption or decryption. The parameter of *Series* in *Data_Dispatch* module is used to determine how many packets are issued in each sampling. The parameter of *Packet_size* can be 16 bytes or 32 bytes. The parameter of *Security* is used to vary the security of PMCS. In the following section, all the hardware architectures corresponding to the configurations of PMCS will be elaborated in detail.

3 HARDWARE DESIGN OF PMCS

Corresponding to four configurations of PMCS algorithm for encryption, architectures with low hardware cost, high throughput, high security, and high throughput and security are designed as shown in Figs. 4 to 7. Depending on the requirement of application, the architecture can be configured with the techniques of pipeline and parallel pipeline. The four-stage pipeline kernel in each architecture is composed of data extension and multi-level data swapping for position permutation, and one-level XOR/XNOR operation and 2D circulation for bit-recirculation with random direction and random number of bits for value transformation, where for balancing the pipeline architecture, we further split the design of 2D circulation stage to two stages.

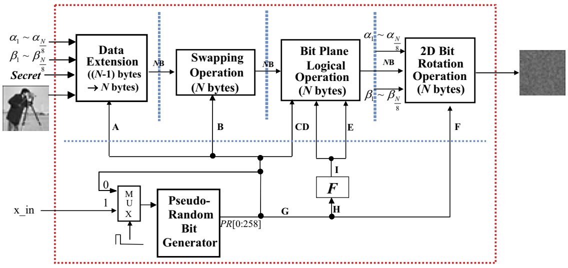

$\alpha_1 \sim \alpha_{\frac{N}{8}}$
$\beta_1 \sim \beta_{\frac{N}{8}}$
Secret
Data Extension ((N-1) bytes → N bytes)
NB
Swapping Operation (N bytes)
NB
Bit Plane Logical Operation (N bytes)
$\alpha_1 \sim \alpha_{\frac{N}{8}}$
NB
$\beta_1 \sim \beta_{\frac{N}{8}}$
2D Bit Rotation Operation (N bytes)
A
B
CD
E
F
I
F
H
G
0
1
M U X
x_in
Pseudo-Random Bit Generator
PR[0:258]

Figure 4: Low hardware cost architecture.

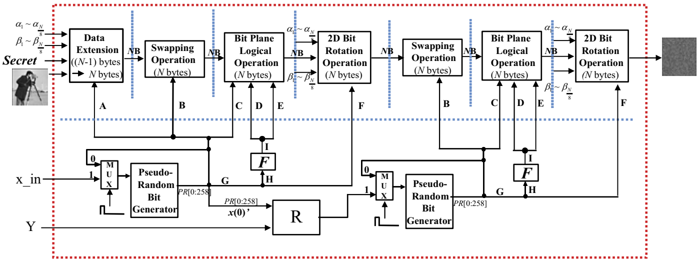

$\alpha_1 \sim \alpha_{\frac{N}{8}}$
$\beta_1 \sim \beta_{\frac{N}{8}}$
Secret
Data Extension ((N-1) bytes → N bytes)
NB
Swapping Operation (N bytes)
NB
Bit Plane Logical Operation (N bytes)
NB
2D Bit Rotation Operation (N bytes)
NB
Swapping Operation (N bytes)
NB
Bit Plane Logical Operation (N bytes)
NB
2D Bit Rotation Operation (N bytes)
A
B
C
D
E
F
I
F
H
G
x_in
M U X
Pseudo-Random Bit Generator
PR[0:258]
PR[0:258]
x(0)'
R
Y

Figure 5: High security architecture.

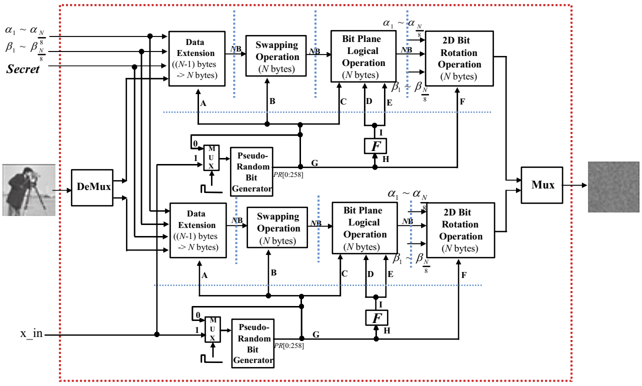

$\alpha_1 \sim \alpha_{\frac{N}{8}}$
$\beta_1 \sim \beta_{\frac{N}{8}}$
Secret
DeMux
Data Extension ((N-1) bytes -> N bytes)
NB
Swapping Operation (N bytes)
NB
Bit Plane Logical Operation (N bytes)
NB
2D Bit Rotation Operation (N bytes)
A
B
C
D
E
F
I
F
H
G
M U X
Pseudo-Random Bit Generator
PR[0:258]
Mux
x_in

Figure 6: High throughput architecture.

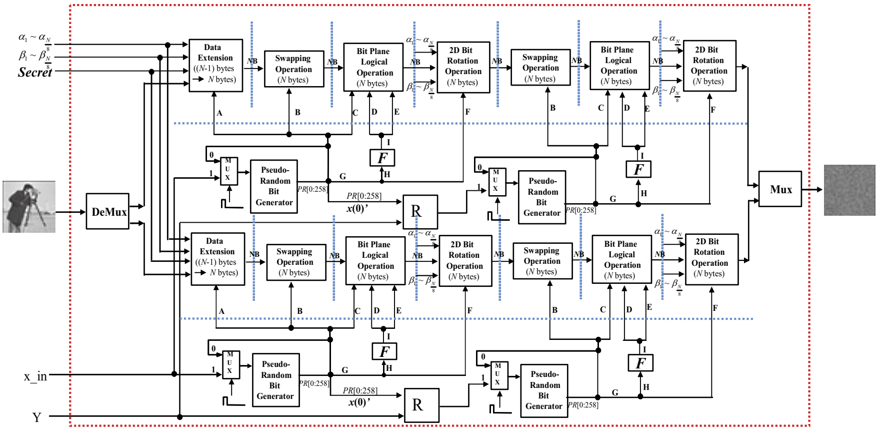

$\alpha_1 \sim \alpha_{\frac{N}{8}}$
$\beta_1 \sim \beta_{\frac{N}{8}}$
Secret
DeMux
Data Extension ((N-1) bytes → N bytes)
NB
Swapping Operation (N bytes)
Bit Plane Logical Operation (N bytes)
2D Bit Rotation Operation (N bytes)
A
B
C
D
E
F
G
H
I
F
M U X
0
1
Pseudo-Random Bit Generator
PR[0:258]
x(0)'
R
Mux
x_in
Y

Figure 7: High security and high throughput architecture.

Detailed designs of the building blocks in each configuration of the PMCS are shown in Fig. 8. The architecture of the PBG, shown in Fig. 9, is used for the generation of chaos-based pseudorandom bit sequence; where 259-bit control signal is generated as the key of PMCS to randomly decide the operations in each pipeline stage. Observing the formulation of the chaos-based pseudorandom bit sequence generation, we concatenate the result of multiplication by wiring rather than by the complex operations of modular operation and truncation operation to minimize the hardware cost of PBG.

For the clocking issue of proposed design, since the computation time for generating the chaos-based pseudorandom bit sequence is much longer than that needed in encryption, the concept of multiple clocks is adopted in the proposed design to realize a slower clock source in the PBG by dividing the original clock with a certain factor, where the dividing factor is determined by the consumption rate of data processing stages. Regarding data issuing, each packet of data for encryption is composed of 15 or 31 bytes and one *Secret* byte. Except the *Secret*, the rest of bytes are sampled serially, and then concatenated with the *secret* at first stage such that a 16- or 32-byte packet is issued to encrypt. Thus, two clocks are also needed in this part of circuitry. While one is used for data sampling, the other is used as the pipeline clock. For maintaining to continuously issue data in the system, two clocks must be synchronized with each other. Namely, the period of 15 cycles for data sampling must be synchronized with the period of 16 or 32 cycles for sending out the 16 or 32 encrypted bytes. However, since synchronization of the two clocks is not so easy, the manner of stalling to sample for one cycle every 16 clock cycles is adopted in our design.

3.1 Timing Analysis

Figure 10 shows the pipeline process of encryption for MCS. The latency is $t_0+t_1+t_2+t_3+t_4+t_5$, and the sample rate can be the largest one of t_0, t_1, t_2, t_3, t_4, t_5 for a packet.

Regarding timing of data sampling in the data expansion stage, as shown in Fig. 11, we can see that N–1 bytes are sampled for each packet. The Nth cycle is used to expand the packet to N bytes, where the extra byte can look as part of the external key.

Figure 12 shows the control pipeline. The processing stages in MCS are respectively controlled by 259 bits of control signal generated from PRBG module. N control signals are

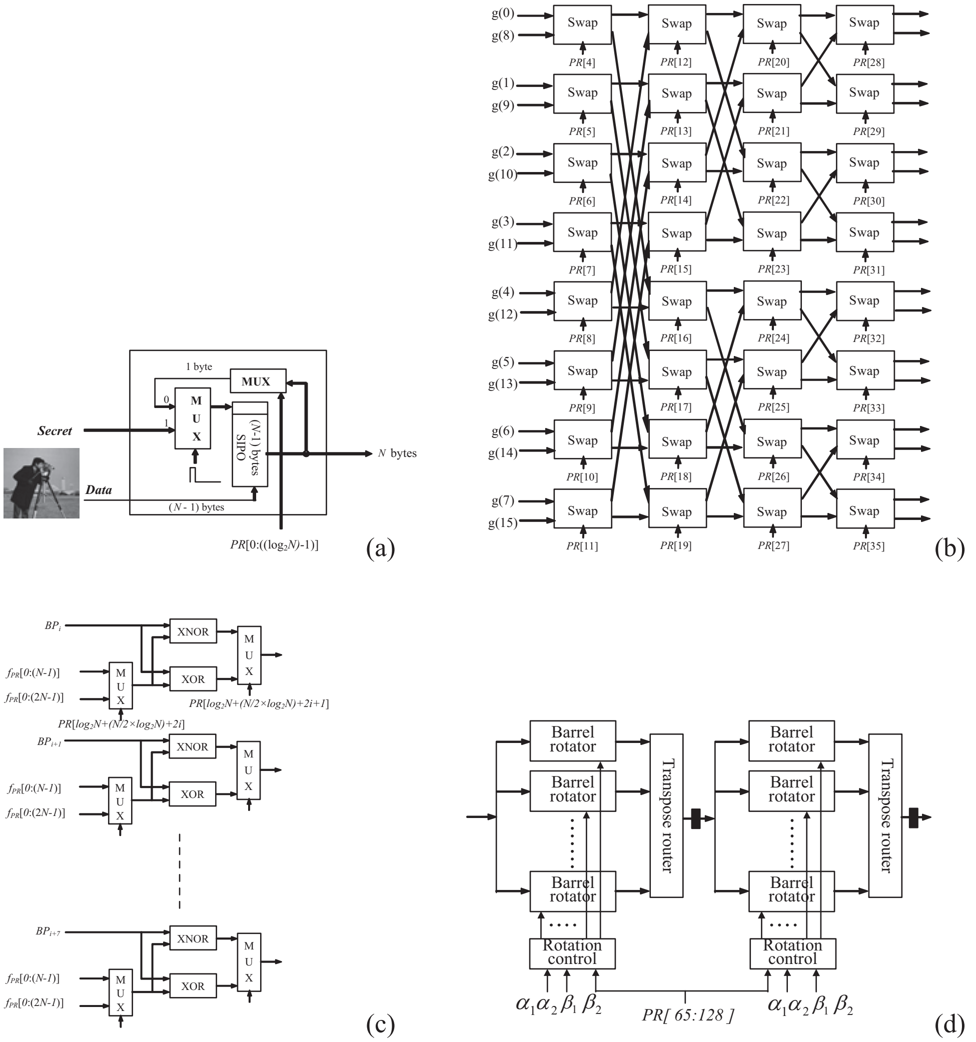

1 byte
MUX
0
1
M U X
(N-1) bytes SIPO
Secret
Data
(N - 1) bytes
N bytes
PR[0:((log2N)-1)]
(a)
g(0) g(8) g(1) g(9) g(2) g(10) g(3) g(11) g(4) g(12) g(5) g(13) g(6) g(14) g(7) g(15)
Swap
PR[4] PR[5] PR[6] PR[7] PR[8] PR[9] PR[10] PR[11]
PR[12] PR[13] PR[14] PR[15] PR[16] PR[17] PR[18] PR[19]
PR[20] PR[21] PR[22] PR[23] PR[24] PR[25] PR[26] PR[27]
PR[28] PR[29] PR[30] PR[31] PR[32] PR[33] PR[34] PR[35]
(b)
BPi
BPi+1
BPi+7
fPR[0:(N-1)]
fPR[0:(2N-1)]
XNOR
XOR
M U X
PR[log2N+(N/2×log2N)+2i]
PR[log2N+(N/2×log2N)+2i+1]
(c)
Barrel rotator
Transpose router
Rotation control
α1α2β1 β2
PR[65:128]
(d)

Figure 8: The detail design of (a) SIPO and *Secret* insertion stage, (b) swapping operation stage, (c) bit-plane logical operation stage, and (d) 2D bit-circulation stage used in the proposed MCS for 16-byte packet size.

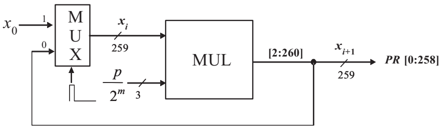


Figure 9: The architecture of PBG.

needed for N packets. The control signal $PR_n[0{:}8\times N]$ for nth packet is generated at t_n. The control signal for packet is fed through MCS stage by stage, thus the part of control signal $PR_n[(6N+1){:}(8N)]$ for the nth packet arrives at vertical one of the 2D Rotation Operation stage at t_{n+4}.

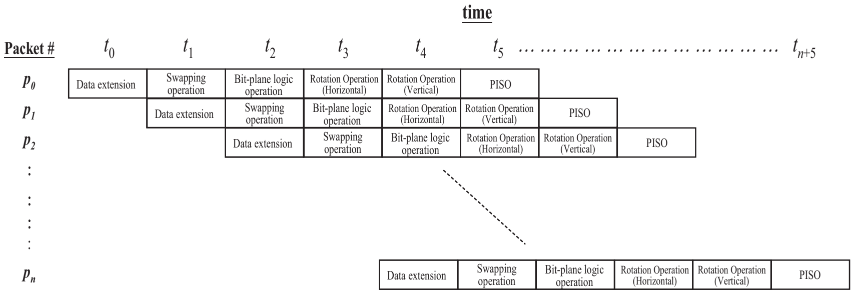


Figure 10: Pipeline of the encryption process for MCS.

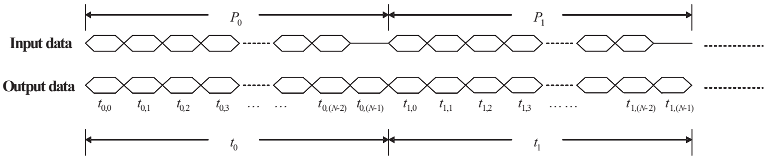
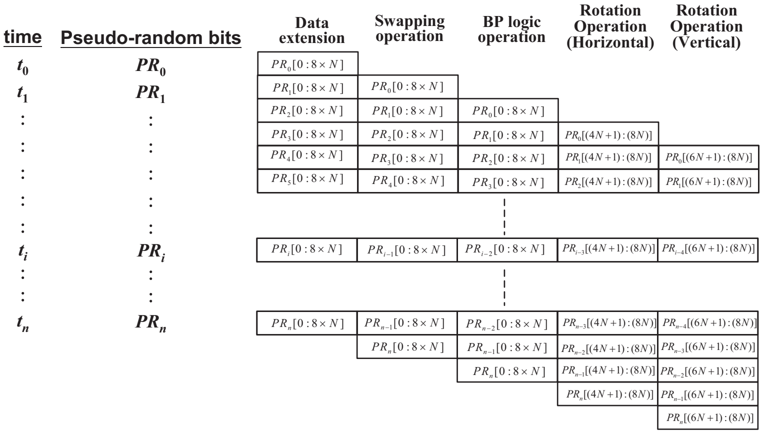


Figure 11: Timing of data sampling for data expansion stage.



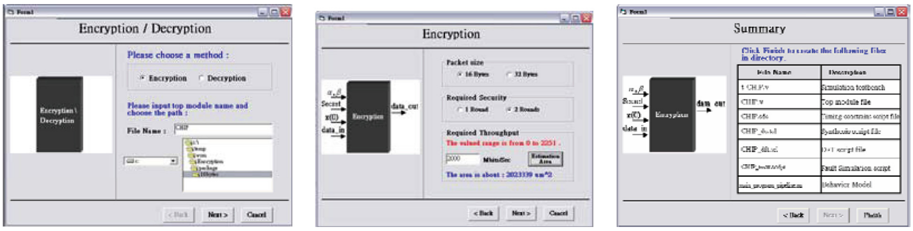
time	Pseudo-random bits	Data extension	Swapping operation	BP logic operation	Rotation Operation (Horizontal)	Rotation Operation (Vertical)
t_0	PR_0	$PR_0[0:8\times N]$				
t_1	PR_1	$PR_1[0:8\times N]$	$PR_0[0:8\times N]$			
:	:	$PR_2[0:8\times N]$	$PR_1[0:8\times N]$	$PR_0[0:8\times N]$		
:	:	$PR_3[0:8\times N]$	$PR_2[0:8\times N]$	$PR_1[0:8\times N]$	$PR_0[(4N+1):(8N)]$	
:	:	$PR_4[0:8\times N]$	$PR_3[0:8\times N]$	$PR_2[0:8\times N]$	$PR_1[(4N+1):(8N)]$	$PR_0[(6N+1):(8N)]$
:	:	$PR_5[0:8\times N]$	$PR_4[0:8\times N]$	$PR_3[0:8\times N]$	$PR_2[(4N+1):(8N)]$	$PR_1[(6N+1):(8N)]$
:	:					
t_i	PR_i	$PR_i[0:8\times N]$	$PR_{i-1}[0:8\times N]$	$PR_{i-2}[0:8\times N]$	$PR_{i-3}[(4N+1):(8N)]$	$PR_{i-4}[(6N+1):(8N)]$
:	:					
:	:					
t_n	PR_n	$PR_n[0:8\times N]$	$PR_{n-1}[0:8\times N]$	$PR_{n-2}[0:8\times N]$	$PR_{n-3}[(4N+1):(8N)]$	$PR_{n-4}[(6N+1):(8N)]$
			$PR_n[0:8\times N]$	$PR_{n-1}[0:8\times N]$	$PR_{n-2}[(4N+1):(8N)]$	$PR_{n-3}[(6N+1):(8N)]$
				$PR_n[0:8\times N]$	$PR_{n-1}[(4N+1):(8N)]$	$PR_{n-2}[(6N+1):(8N)]$
					$PR_n[(4N+1):(8N)]$	$PR_{n-1}[(6N+1):(8N)]$
						$PR_n[(6N+1):(8N)]$

Figure 12: The control pipeline for encryption in MCS.

4 VERIFICATION OF THE PMCS HARDWARE GENERATOR

We have implemented the parameterized VLSI generator of proposed PMCS with different synthesis options. To facilitate usage of the VLSI design, we provide a VLSI generator with graphic user interface (GUI), as shown in Fig. 13, such that synthesizable RTL code, test-benches, and synthesis script of the desired configuration of PMCS can automatically be generated to meet the requirements of applications.

For a fair verification, based on 0.18 um CMOS technology, we verify all the designs with different configurations by cell-based IC design flow. Table 9 shows throughput and area cost of the proposed parameterized VLSI design for all configurations. We can see that the throughput can be ranged between 1.59 and 2.25 Gbps with the area of 0.54 and 3.92 mm^2.



Please choose a method :

● Encryption ○ Decryption

Please input top module name and choose the path :

File Name : CHIP

Packet size

● 16 Bytes ○ 32 Bytes

Required Security

○ 1 Round ● 2 Rounds

Required Throughput

The valued range is from 0 to 2251 .

2000 Mbits/Sec Estimation Area

The area is about : 2023339 um^2

Click Finish to create the following files in directory.

File Name	Description
t CHIP.v	Simulation testbench
CHIP.v	Top module file
CHIP.sdc	Timing constraint script file
CHIP_dc.tcl	Synthesis script file
CHIP_dft.tcl	DFT script file
CHIP_tmax.tcl	Fault Simulation script
main_program_pipeline.m	Behavior Model

Figure 13: GUI of the proposed VLSI generator for generating the desired data encryption/ decryption hardware with synthesizable RTL code, test-benches, and synthesis script.

Table 9: Throughput, area, and power consumption corresponding to the configurations of proposed VLSI generator.

Processing Type	Pachet size Architecture	Throughput (Mbits/s)		Area (mm^2)		Power consumption (uW/MHz)	
		16	32	16	32	16	32
Encryption	Low cost	1597	1703	0.5357	0.9293	266	324
	High Security	1648	1651	1.0967	1.8698	473	579
	High throughput	2177	2186	0.9958	1.7848	409	536
	High security and throughput	2223	2251	2.0233	3.5117	787	863
Decryption	Low cost	1648	1653	0.5125	0.8750	284	386
	High security	1597	1651	1.1442	2.0301	553	707
	High throughput	2177	2249	0.9426	1.6948	455	579
	High security and throughput	2177	2249	2.1381	3.9220	882	1088

It reveals that the performance is sufficient for most of the requirement in high-quality multimedia applications with reasonable hardware cost.

Regarding qualification of the proposed VLSI, all the configurations of the design have to be passed by both RMM coding guidelines and code coverage. There is no error and only with few petty warnings under the coding guidelines. For the code coverage verification, we adopted *VN-cover* to check the code coverage. The proposed VLSI for all configurations has been certified to have the code coverage near to 100% with the provided test-benches.

In the following, we show the performance evaluation of proposed design and the other existing designs [5-9]. To eliminate the factor of different fabrication technologies, the index of normalized area (NArea) defined by [17] is adopted as eqn (1), where the silicon area is normalized to 0.35 um technology.

$$NArea = \frac{Area}{(Techno\log y / 0.35)^2}. \quad (1)$$

Table 10: Comparison of the proposed design and existing designs [5–9].

Design	Technology (um)	Area (mm^2)	Data rate (Mbits/s)	Normalized area (NArea) (mm^2)	Date-rate/ NArea (DRPA) (Mbps/ mm^2)	Encryption algorithm	Cipher block length/ key length (bit)
Design [5]	0.7	29	251.8	7.25	34.73	SAFER K-128	64/ 128
Design [6]	0.35	13.69	1320	13.69	96.42	AES	128/ 128
Design [7]	0.35	3.59	856	3.59	238.44	TDCEA	64/40
Design [8]	0.18	6.25	17	13.4	1.27	AES	128/ 256
Design [9]	0.35	3.63	500/519	3.63	137.7/143	SXRDEA	56/72
Proposed ($N = 32$)	0.18	0.9293	1703	3.513	484.77	MCS	248/291

In addition, as shown in eqn (2) we further provide the index of DRPA [17] to reflect the efficiency of hardware cost for encryption and decryption.

$$DRPA = \frac{Data - rate}{NArea}\left(\frac{Mbps}{mm^2}\right). \tag{2}$$

According to the two indices, we have summarized the comparison of the proposed designs and the existing ones [5-9] as Table 10. It reveals that the proposed design is better than some of the existing designs in providing higher data processing rate at lower hardware cost. Although the comparison cannot reveal for all aspects, such as exact security, the simulation result reflects the efficiency of proposed algorithm and its hardware design is good enough for applications with requirement of real-time.

5 CONCLUSIONS

In this paper, a new cryptographic system with high security is proposed. Moreover, for the applications with real-time in multimedia system, the high-performance parameterized architecture and its VLSI generation software are developed. Four architectures corresponding to the parameterized cryptographic system are designed for low hardware cost, high throughput, high security, and high throughput and security, which consist of pipeline or parallelpipeline architectures, and verified with 0.18 um CMOS technology. The verifications of all configurations show that the throughput of proposed designs can be ranged between 1.59 and 2.25 Gbps with the area of 0.54 and 3.92 mm^2. Comparing with the existing designs, the performance of proposed designs is better than the others in term of the evaluation index DRPA. It reveals that the proposed high-performance VLSI design is easily integrated in most of high quality multimedia system with the requirement of real-time and reasonable hardware cost.

REFERENCES

[1] Data encryption standard, FIPS PUB 46, National Bureau of Standards, Washington, D.C., Jan. 1997.

[2] Daemen, J. & Rijmen, V., AES Proposal: Rijndael, AES Algorithm submission, Sep. 1999.

[3] Yi, X., Tan, C.H., Kheong, C.K. & Syed, M.R., Fast Encryption for Multimedia, *IEEE Trans. on Consumer Electronics*, **47**, pp. 101–107, 2001. doi: http://dx.doi.org/10.1109/30.920426

[4] Lu, C.C. etc., Integrated design of AES Encrypter and Decrypter, *Proc. ASAP'2002*, pp. 277–285, 2002.

[5] Schubert, A., Meyer, V. & Anheier, W., Reusable cryptographic VLSI core based on the SAFER K-128 algorithm with 251.8 Mbit/s throughput, *Proc. 1998 IEEE Workshop on Signal Processing System,* pp. 437–446, 1998.

[6] Mitsuyama, Y., Andales, Z., Onoye, T. & Shirakawa. I., VLSI implementation of high performance burst mode for 128-bit block ciphers, *Proc. ISCAS'2001*, **2**, pp. 344–347, 2002.

[7] Lin, S.W., The Design and Realization of the IP Core for Data Encryption in the Video Surveillance System, *Master Thesis, Computer Science and Information Engineering, National Chung Cheng University*, 2003.

[8] Lewis, M., Simmons, S, A VLSI implementation of a cryptographic processor, *Proc. IEEE, CCECE 2003*, pp. 821–826, 2003.

[9] Chen, H.C., Yen, J.C., Wu, S.M. & Zhong, J.K., On the parameterized IP core design of the new cryptographic system, *Proc. IEEE, ISPACS'2005*, pp. 337–340, 2005.

[10] Selimis, G., Lazarou, N., Michail, H.E. & Koufopavlou, O., VLSI design and implementation of reconfigurable cryptographic systems for symmetric encryption, *Proc. IEEE, ICECS 2005*, pp. 1–4, 2005.

[11] Michail, H.E., Kakarountas, A.P., Milidonis, A.S., Panagiotakopoulos, G.A., Thanasoulis, V.N. & Goutis, C.E., Temporal and System Level Modifications for High Speed VLSI Implementations of Cryptographic Core, *Proc. IEEE, ICECS 2006*, pp. 1180–1183, 2006.

[12] Megalingam, R.K., Gopakumar, G., Luke, D., Jyothi, K.S. & Ajit, A., A VLSI implementation and analysis of cryptographic algorithms for security and privacy in communication networks, *Proc. IEEE, ICMET 2010*, pp. 521–525, 2010.

[13] Keating, M. & Bricaud, P., *Reuse Methodology Manual*, Kluwer Academic Publishers, 2002.

[14] Kocarev, L. & Jakimoki, G., Pseudorandom bits generated by chaotic maps, *IEEE Trans. on Circuits and Systems – Part I*, 50, pp. 123–126, 2003.

[15] Chen, C.C., Daponte, J.S. & Fox, M.D., Fractal feature analysis and classification in medical imaging, *IEEE Trans. on Medical Imaging*, **8**, pp. 133–142, 1989. doi: http://dx.doi.org/10.1109/42.24861

[16] Parker, T.S. & Chua, L.O., Chaos - A tutorial for engineers, *Proc. IEEE*, **75**, pp. 982–1008, 1987. doi: http://dx.doi.org/10.1109/PROC.1987.13845

[17] Bass, B.M., A low-power high performance 1024-point FFT processor, *IEEE Journal of Solid-State Circuit*, 34, pp. 380–387, 1999. doi: http://dx.doi.org/10.1109/4.748190