1. Introduction

Security is imperative for many systems, such as the water-based automatic security marking platform \cite{1}, cognitive radios \cite{2-5}, smart grid \cite{6}, physical layer \cite{7}, and cloud and fog computing \cite{8}. Moreover, in addition to high-performance packet switching \cite{9,10}, commercial home gateways \cite{11} or switches/routers necessitate high-throughput crypto processors. Also, security is of paramount importance for many consumer electronics \cite{12,13}. The private and public key crypto processor was proposed \cite{12}. However, to the best of our knowledge, the high-throughput crypto
processor for feedback operation modes, such as the cipher-block chaining (CBC) mode, has never been studied.

In cryptography, a mode of operation describes how to repeatedly apply the single-block operation of a cipher to securely operate on data larger than a block. In the CBC mode, each block of plaintext is exclusively-ORed (XORed) with the previous ciphertext block before being encrypted, as presented in Figure 1, where $\oplus$ denotes the XOR logic operation, ENC denotes the block cipher, and an initialization vector must be used in the first block to make each message unique. The block cipher operates on a whole block and requires that the data be padded to a full block if it is smaller than the block size. Consequently, in the CBC mode, each ciphertext block relies on all plaintext blocks processed up to that point.

### 1.1 Advanced encryption standard

In 2001, National Institute of Standard and Technology (NIST) invited proposals for the new algorithm of the advanced encryption standard (AES) [14]. The Rijndael algorithm, designed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, was finally selected as the AES specification and became a FIPS standard.

Nowadays, the AES algorithm is the most popular symmetric block cipher, within which both the outbound and inbound respectively use the same main key for encryption and decryption. Additionally, the AES is an iterative algorithm and uses a round function repeatedly. The number of iterations is determined by the number of rounds. In encryption, each round is composed of four different processing steps: substitute bytes (SubBytes), shift rows (ShiftRows), mix columns (MixColumns), and add round key (AddRoundKey), while the last round does not contain the MixColumns step.

The decryption is a reverse process of encryption and the round keys are used in the reverse order. In decryption, each round is also composed of four different processing steps: inverse substitute bytes (InvSubBytes), inverse shift rows (InvShiftRows), inverse mix columns (InvMixColumns), and add round key (AddRoundKey), while the last round does not contain the InvMixColumns step.

Compared to the software solution [15], the hardware implementation [16-22] is more suitable for high-throughput data applications. Among hardware implementations, the non-linear SubBytes transformation realized using look-up tables (LUTs) [16,17] requires a large area compared to those using the composite field arithmetic (CFA) [18-22]. The study [18] proposed a pipelined and unfolded AES circuit using the GF($2^4$) 2, where GF( . ) denotes the Galois field. The work [19] combined and optimized all building blocks in the devised S-box of SubBytes [18]. The work [20] proposed to use of a single-round circuit of AES repeatedly. The whole AES processing is performed in GF($2^4$) 2 and pipelining registers are inserted between iterative rounds [21]. The work [22] adopted the pre-computation technique and proposed a three-block design of the S-box. A 2-stage pipelined round circuit is finally obtained [22]. The works [21,22] unroll the whole AES circuit.

![Figure 1. CBC operation mode of encryption.](image-url)
1.2 Data encryption standard

The data encryption standard (DES) algorithm is also a symmetric key encryption algorithm, which was established as a data encryption standard by the NIST in 1976. However, with the advancement of technology, the key length of DES was too short to be easily cracked. Therefore, a triple data encryption algorithm (TDEA or 3DES) was invented to solve this problem by increasing the key length. Nowadays, there are more and more specialized hardware circuits developed to handle the data encryption task. The work combined two sets of single-round DES circuits in parallel and used two different-phase clocks to control the circuit. The work unrolled the DES circuit and inserted registers between each round of the DES circuit to implement a pipelined architecture. The works decreased the critical path of single-round DES by removing the XOR logic gate. Furthermore, the work combined multiple rounds of DES to reduce the number of cycles for encryption.

1.3 Motivation

With the development of communication technology, not only the high data throughput rate is an important issue, but data security is also highly noticed. The CBC mode is considered to be more secure than traditional electronic codebook (ECB) mode. Therefore, how to realize the CBC mode circuit with high throughput has become a critical issue.

On a network router/switch, packet data of multiple independent network channels are multiplexed into a port. To increase the clock rate of a crypto engine, a pipelined engine is typically designed for the ECB mode of operation. However, for the popular CBC mode, owing to the XOR operation of a plaintext block with the previous ciphertext block, pipelining will not increase the data throughput. Rather, throughput may be lowered due to pipelining because of unbalanced path delays and the overhead of register access time. Meanwhile, inserting pipelining registers will cause a waste of hardware resources.

To solve the impact of packet data dependency and enhance the throughput of the CBC mode encryption system, the novelty of this work is that we use an architecture for data scheduling to eliminate the data dependency. In addition, the pipelined architecture focuses on balancing the latency of crypto engines to achieve high throughput. More specifically, this study provides several contributions outlined below. 1) With the proposed architecture, we can easily schedule the input data of multiple network channels to remove the impact of data dependency and allocate hardware resources flexibly to each network channel. 2) The pipelined stages are fully utilized. Therefore, only a copy of the proposed circuit is required to encrypt the packet data of multiple network channels at the same time. 3) We propose new pipelined AES and 3DES circuits to achieve high throughput. 4) The proposed scheme is verified using both AES-CBC and 3DES-CBC.

The rest of this work is organized as follows. Section 2 illustrates the design consideration and proposed architecture for the CBC mode of operation. Section 3 describes the proposed crypto engines, including the AES-CBC and 3DES-CBC. Section 4 demonstrates the implementation result and comparison. Finally, Section 5 draws conclusions.

2. Design considerations

Many works in the literature focus on designing the ECB mode crypto engine, instead of the popular CBC mode. However, due to the CBC mode of operation, there exists data dependency in the data to be encrypted. Conventional data scheduling of CBC mode using a 3-stage pipelined engine is presented in Figure 2, where pt and ct denote the plaintext and ciphertext, respectively. At the first cycle, the first plaintext, pt0, is fed into the first pipelined stage to start encryption. Next cycle, pt0 will be shifted to the second pipeline stage, and concurrently, the second plaintext, pt1, will be fed into the first pipeline stage. However, due to the CBC mode, pt1 cannot start the encryption until pt1 has been completely encrypted at cycle 4. Therefore, pt1 will be held at the first pipeline stage for two cycles to wait for the first ciphertext. As displayed, the pipeline stages cannot be
fully utilized for the CBC mode of operation.

Figure 2. Conventional data scheduling of CBC mode using a 3-stage pipelined engine.

To improve the throughput of CBC encryption mode, the parallel architecture[29] was proposed in Figure 3(A). In order to encrypt multiple network channels at the same time, multiple sets of pipelined circuits are used in parallel. The major difference of this work between the parallel architecture is that, to maintain the same throughput as the parallel architecture while reducing the required chip area, the multi-channel crypto engine using the folded architecture[17] was proposed in Figure 3(B). This work implements the multi-channel crypto engines based on the AES and 3DES. The pseudocode of the proposed algorithm is presented in Table 1. The encryption algorithms of AES and DES can be found in the literature and are briefly introduced in Section 1, and hence, they are omitted here for clarity.

Objectives of this work

The objective of this study is to investigate a high-throughput CBC mode crypto circuit, which can be embedded in commercial home gateways or switches/routers. Concurrently, the area efficiency of block ciphers can be improved as well. However, the CBC mode encounters the problem of data dependency. To solve this issue, a data scheduling mechanism of network packets is proposed to eliminate the data dependency of input data for CBC mode pipelined crypto engines. The pros and cons of the proposed method are summarized in Table 2.

Table 1. Pseudocode of the proposed algorithm

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>plaintext of each channel</td>
<td>ciphertext</td>
</tr>
<tr>
<td>1 for each channel;</td>
<td>3 end</td>
</tr>
<tr>
<td>2 encrypt plaintext;</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Pros and Cons of the proposed method.

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>High throughput and area efficiency</td>
</tr>
<tr>
<td></td>
<td>Round-robin data scheduling is needed</td>
</tr>
</tbody>
</table>
3. Crypto engines

3.1 Pipelined AES-CBC engine

To increase the clock rate, an intuitive solution is to use a pipelined architecture. However, making path delays balanced is the key point of pipelined circuits. Considering the path delays of combinational circuits, we propose to implement the 7-stage pipelined single-round circuit of AES in Figure 4, where the detailed design of S-box (for SubBytes) sub-block can be found[19] and it is omitted here. The AddRoundKey is simply the XOR operation shown in Figure 4. The ShiftRows and InvShiftRows are rewiring their inputs and do not cost any logic gates.

Figure 4. Pipelined architectures of single-round circuit of AES.

Notice that the dashed lines are locations where the registers are placed.

Next, as shown in Figure 6, every round of the AES circuit with a 128-bit key is unrolled to achieve the highest throughput so that one ciphertext can be obtained in every clock cycle.

Figure 5 displays the detailed architecture of the integrated Mixcolumns and InvMixcolumns for the encryption and decryption, respectively, where $a, b, c, d$ denote the inputs, $a', b', c', d'$ denote the outputs of Mixcolumns, $W, X, Y, Z$ denote the outputs of InvMixcolumns, $\oplus$ denotes the XOR operation, and $\times$ denotes the circuit that multiplies input by 2. Notably, the last round of AES needs not the Mixcolumns operation so it has 6-stage pipelining, which is omitted here. Notice that the dashed lines are locations where the registers are placed.

Figure 5. Detailed architecture of the integrated Mixcolumns and InvMixcolumns for the encryption and decryption, respectively.
3.2 Pipelined 3DES-CBC engine

To balance the path delay of every pipelined stage, we start with the S-box because it has the longest delay in the 3DES-CBC circuit. The 64-to-1 multiplexer that composes the S-box is replaced with a tree structure composed of different-size multiplexers, and registers (marked by the dashed lines) are placed between the multiplexers, as shown in Figure 7.

![Figure 7](image-url)  
Figure 7. The architectures of S-box for 3DES with different pipeline stages, (A) 2-stage, (B) 3-stage, (C) 4-stage, and (D) 6-stage.

We investigate four kinds of S-box architectures: 2-stage pipeline with 8-to-1 multiplexers, 3-stage pipeline with 4-to-1 multiplexers, 4-stage pipeline with 2-to-1 multiplexers and 4-to-1 multiplexers, and 6-stage pipeline with 2-to-1 multiplexers. Applying them to the 3DES-CBC circuits, critical paths of the pipelined architectures are shown in Table 3. Furthermore, in order to facilitate the evaluation of the critical path for different pipelined architecture, the XOR logic gates and multiplexers of each critical path are replaced with equivalent NAND gates.

Table 3. The critical paths of each pipelined architecture and their NAND gate equivalents.

<table>
<thead>
<tr>
<th>Pipelined Architecture</th>
<th>Critical Path</th>
<th>Equivalent NAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-Stage Pipeline</td>
<td>1 XOR, 1 8-to-1 MUX</td>
<td>12 NANDs</td>
</tr>
<tr>
<td>3-Stage Pipeline</td>
<td>1 XOR, 1 4-to-1 MUX</td>
<td>9 NANDs</td>
</tr>
<tr>
<td>4-Stage Pipeline</td>
<td>1 4-to-1 MUX</td>
<td>6 NANDs</td>
</tr>
<tr>
<td>8-Stage Pipeline</td>
<td>1 2-to-1 MUX</td>
<td>3 NANDs</td>
</tr>
</tbody>
</table>

According to the critical path target, the 3DES-CBC can be designed. First, the single-round circuit of the $i$-th round of the 3DES-CBC is shown in Figure 8, where the 2-stage pipelined S-box is assumed, $L_i$ and $R_i$ denote the left and right 32-bit halves of the $i$-th round, and the blocks E and P denote the expansion and permutation functions, respectively. Notably, expansion and permutation functions are just rewiring of their inputs and do not cost any logic gates except buffers if required. The red dashed lines are the positions where the registers are placed. S-box part 1 and S-box part 2 respectively represent the two parts of the 2-stage S-box shown in Figure 7(A).

Second, in addition to pipelining a single-round circuit, we unroll and pipeline 16 rounds of a DES and it is reused for 48 rounds of operation of 3DES-CBC, as shown in Figure 9, where “pipelined 3DES round” represents the pipelined single-round circuit.
4. Implementation result and comparison

For comparison purpose, the ASIC implementation of AES-CBC is based on the TSMC 45 nm standard cell library and the synthesis tool of Synopsys Design Compiler. The implementation results of various crypto engines are displayed in Table 4. As presented, the proposed AES-CBC achieves the highest throughput and area efficiency. However, owing to the deep pipelining, the area of the proposed design is also the largest.

For comparison purpose, the ASIC implementation of 3DES-CBC is based on the TSMC 130 nm and 65 nm standard cell libraries and the synthesis tool of Synopsys Design Compiler. The implementation results of various crypto engines are displayed in Table 5. The authors [27] combined the multiple iterative rounds in one clock cycle. Besides, a set of XOR gates was removed from the critical path to reduce the critical path delay. The highest throughput [27] is “16 rounds in 1 cycle” and can reach 1.94 Gbps. The work [28] removed two XOR gates from the critical path and reused the single-round hardware, and can reach a throughput of 1.69 Gbps. As presented in Table 5, the proposed 8-stage pipelined architecture has the highest throughput of 44.75 Gbps, while the proposed 2-stage pipelined architecture has the high-

<table>
<thead>
<tr>
<th>Tech.</th>
<th>Architecture</th>
<th>Freq. (Mhz)</th>
<th>Throughput (Gbps)</th>
<th>Area</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>[20] NanGate 45nm</td>
<td>AES-128, Enc/ Dec</td>
<td>694.4</td>
<td>0.889/0.808</td>
<td>17368 GE</td>
<td>51.18/46.52 Kbps/GE</td>
</tr>
<tr>
<td>[21] High-metal gate CMOS 45nm</td>
<td>AES-128, Enc/ Dec</td>
<td>2100</td>
<td>2.65</td>
<td>0.15 mm²</td>
<td>17.67 Kbps/μm²</td>
</tr>
<tr>
<td>Our Work TSMC 45nm</td>
<td>AES-128, Enc/ Dec</td>
<td>1075</td>
<td>137.8</td>
<td>75832 μm²/111517 GE</td>
<td>1817 Kbps/μm² 1235 Kbps/GE</td>
</tr>
</tbody>
</table>

Figure 8. Single-round circuit of the i-th round of 3DES-CBC.

Figure 9. The architecture of Pipelined 3DES-CBC.

Table 4. Comparison among different AES-CBC ASIC designs.
est area efficiency of 85.37 Kbps/μm². Moreover, the throughput of the proposed design implemented using 130 nm standard cell library still outperforms the works [27,28] in literature implemented using 65 nm standard cell library, which validates the efficiency of the proposed architecture.

We compare state-of-the-art works [30,31] to show the robustness of the implemented AES-CBC design through FPGA implementation in Table 6. As displayed, under the same throughput, the proposed design can achieve the best area efficiency. This justifies the advantages of the proposed architecture.

### Table 5. Comparison among different 3DES-CBC ASIC designs.

<table>
<thead>
<tr>
<th>Tech.</th>
<th>Architecture</th>
<th>Freq. (Mhz)</th>
<th>Throughput (Gbps)</th>
<th>Area (μm²)</th>
<th>Efficiency (Kbps/μm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[27] SMIC 130nm</td>
<td>CBC, 4 rounds in 1</td>
<td>275.6</td>
<td>1.47</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>[27] SMIC 130nm</td>
<td>CBC, 16 rounds in 1</td>
<td>90.93</td>
<td>1.94</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>[27] SMIC 65nm</td>
<td>CBC, 16 rounds in 1</td>
<td>156.09</td>
<td>3.33</td>
<td>164599</td>
<td>20.23</td>
</tr>
<tr>
<td>[28] SMIC 130nm</td>
<td>CBC</td>
<td>N/A</td>
<td>1.69</td>
<td>38688.8</td>
<td>43.68</td>
</tr>
<tr>
<td>[28] SMIC 65nm</td>
<td>CBC</td>
<td>2130</td>
<td>2.84</td>
<td>12852</td>
<td>220.9</td>
</tr>
<tr>
<td>Our Work</td>
<td>TSMC 130nm CBC, 2-stage pipelined</td>
<td>574.052</td>
<td>36.74</td>
<td>430338</td>
<td>85.37</td>
</tr>
<tr>
<td>Our Work</td>
<td>TSMC 130nm CBC, 3-stage pipelined</td>
<td>638.978</td>
<td>40.89</td>
<td>536974</td>
<td>76.14</td>
</tr>
<tr>
<td>Our Work</td>
<td>TSMC 130nm CBC, 4-stage pipelined</td>
<td>671.59</td>
<td>42.98</td>
<td>625685</td>
<td>68.69</td>
</tr>
<tr>
<td>Our Work</td>
<td>TSMC 130nm CBC, 8-stage pipelined</td>
<td>699.3</td>
<td>44.75</td>
<td>1048556</td>
<td>42.68</td>
</tr>
</tbody>
</table>

### Table 6. Comparison among different AES-CBC FPGA designs.

<table>
<thead>
<tr>
<th>Arch.</th>
<th>Devices</th>
<th>Slices</th>
<th>Freq. (Mhz)</th>
<th>Throughput (Gbps)</th>
<th>Efficiency (Mbps/Slice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[30] AES-128, Enc</td>
<td>Virtex-6 xc6vlx240t</td>
<td>4830</td>
<td>617.6</td>
<td>79</td>
<td>16.36</td>
</tr>
<tr>
<td>Our Work</td>
<td>Virtex-6 xc6vlx240t</td>
<td>3140</td>
<td>631.2</td>
<td>80.8</td>
<td>25.37</td>
</tr>
<tr>
<td>[31] AES-128, Enc/Dec</td>
<td>Virtex-5 xc5vfx70t</td>
<td>9756</td>
<td>460</td>
<td>60</td>
<td>6.15</td>
</tr>
<tr>
<td>Our Work</td>
<td>Virtex-5 xc5vfx70t</td>
<td>5277</td>
<td>473.2</td>
<td>60.6</td>
<td>11.47</td>
</tr>
</tbody>
</table>

### 5. Conclusions and future works

In this paper, we presented a folded architecture for encrypting the packet data of different network channels at the same time. The resources of the pipelined crypto engine can be fully utilized without any waste. Only one copy of pipelined circuit is required to maintain the same throughput of the parallel architecture. There typically exists a tradeoff between throughput and the area of digital circuits. In addition to enhancing the throughput, adopting the proposed technique can also enhance the area efficiency (throughput/area).

Future works can implement the layout of the proposed design. Another research direction can realize the buffer space required to store the plaintexts of each channel.

### Nomenclature

- AddRoundKey: add round key
- AES: advanced encryption standard
- CBC: cipher-block chaining
- CFA: composite field arithmetic
- ECB: electronic codebook
- InvMixColumns: inverse mix columns
- InvShiftRows: inverse shift rows
- InvSubBytes: inverse substitute bytes
- LUTs: look-up tables
- MixColumns: mix columns
- NIST: National Institute of Standard and Technology
- ShiftRows: shift rows
- SubBytes: substitute bytes
- TDEA or 3DES: triple data encryption standard
- XOR: exclusively-OR
Author Contributions

Kai-Chun Chang: Design and implementation of 3DES.
You-Tun Teng: Design and implementation of AES.
Wen-Long Chin: Supervision.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research received no external funding.

References

composite field arithmetic for IoT applications.


Appendix A

Synthesis Constraints

- create_clock -period 1.565 [get_ports clk]
- set_ideal_network -no_propagate [get_clocks clk]
- set_ideal_network [get_ports rst]
- set_dont_touch_network [get_ports {clk rst}]
- set_clock_latency 0.5 [get_clocks clk]
- set_input_delay 0.1 -clock clk [remove_from_collection [all_inputs]/
  [get_ports {clk rst}]]
- set_output_delay 0.1 -clock clk [all_outputs]
- set_fix_multiple_port_nets -all -buffer_constants
- set_load 0.05 [all_outputs]
- set_drive 0 [get_ports rst]
- set_driving_cell -lib_cell INVXL -no_design_rule [remove_from_collection/ [all_inputs] [get_ports {clk rst}]]

- set_max_area 0
- set_max_fanout 4 TDESCBC_intra_pipeline
- set_operating_conditions -min_library fast -min_fast -max_library slow/ -max slow
- set_wire_load_model -name tsmc13_wl10 -library slow
- set_wire_load_mode top
- remove_unconnected_ports -blast_buses [get_cells -hierarchical *]

Appendix B

FPGA Implementation Setup in ISE 14.7

- Family: Virtex4
- Device: XC4VLX25
- Package: FF676
- Speed: -10
- Compile Option: default