A design of high-Speed SMS4 cipher circuit

Chenjie Wu, Ying Tang* and Yi Wei
College of Optical and Electronic Technology, China Jiliang University, Hangzhou, Zhejiang, 310000, China
*Corresponding author’s e-mail: tangy@cjlu.edu.cn

Abstract. SMS4 cipher algorithm is a commercial cipher independently developed by China. This paper proposes a circuit design of high-speed SMS4 cipher algorithm, which adopts two-dimensional expansion and key path optimization technology. The round key generation module runs synchronously with the encryption module, and the running round is constant, which can effectively resist the side channel attack. Using Xilinx artix-7 for synthesis, the results show that the design function is correct, and achieves the balance of computing speed and resource occupation.

1. Introduction
With the rapid development of computer network technology, information security has also become a key issue that must be faced in network communication technology, and the application of cryptography is a key factor in safeguarding data integrity and confidentiality. There are three main types of cryptographic algorithms: asymmetric cryptographic algorithms, hash algorithms and symmetric algorithms. Among them, symmetric algorithms are widely used in big data encryption, which can achieve high performance in encrypting data and have reliable security.

SMS4 encryption algorithm is one of the encryption specifications for wireless LAN products announced by the State Cryptography Administration, which is the first time that China announces its own commercial cryptographic algorithm, marking that China's commercial cryptographic management is more scientific and in line with international standards. Initially used in the field of wireless LAN, it is now widely used in the fields of Internet of Things and smart cards due to its high security and easy implementation.

Currently, there are two main ways to implement the SMS4 algorithm: software and hardware. The software implementation method executes the program written to run on the CPU, which causes a considerable waste of resources due to the use of a general-purpose processor, making it difficult to increase the speed of the algorithm, and greatly restricting power consumption and security. In contrast, the hardware implementation, which uses a dedicated processor to execute the algorithm, can realize the miniaturization of the device, reduce power consumption and improve security. On this basis, many scholars have made great efforts in structure optimization in order to further reduce power consumption and speed up computing. In the literature[3], the SMS4 module is designed in the form of an IP core, which is interconnected with the CPU through the bus, and the cryptographic algorithm is completed by the hardware and software in cooperation, requiring a total of 128 cycles to complete the operation. The literature[5] proposes an architecture that uses the host computer for round key generation and the hardware to perform the cryptographic operations, and uses a double cascade structure for the cryptographic operations, reducing the number of iterations to half. The literature[9]
designed a lightweight circuit with a serialized design approach, synthesized using FPGAs, and used only 242 ALMs, but the encryption required longer clock cycles and lower throughput. In order to achieve faster processing speed with less hardware overhead, this design proposes a cyclic design using a two-degree unfolding structure, which optimizes the key road strength in the circuit to achieve a faster synthesis frequency. The encryption process is fully processed using hardware, and the total number of encryption rounds is constant, which can effectively resist side channel attacks. The synthesis is completed on a Xilinx Artix-7 to verify the correctness of the design, and the results show that the design has significant improvement in resource usage and processing speed compared with the previous design.

2. Algorithm Analysis

SMS4 encryption algorithm is a grouping algorithm that groups the plaintext information to be encrypted according to a fixed length[1], the encryption key length is equal to the plaintext length, and the grouped plaintext and the key are changed in multiple rounds to generate the ciphertext. The length of the plaintext and the key is 128 bits, and both the encryption algorithm and the key expansion algorithm are generated by 32 rounds of nonlinear iteration.

2.1. Round Key Generation

The round key generation algorithm is shown in Figure 1 and is generated from the given encryption key by the key extension algorithm.

The input key \( MK = (MK_0, MK_1, MK_2, MK_3) \) and the round key is generated by equations (1) and (2).

\[
\begin{align*}
(K_0, K_1, K_2, K_3) &= (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3) \quad (1) \\
k_{i+4} &= K_i \oplus T' (K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i) \quad i = 0, 1, 2 \ldots 31 \quad (2)
\end{align*}
\]

FK and CK are constants. The synthetic permutation \( T' \) consists of a nonlinear operation \( \tau \)-transform and a linear operation \( L \)-transform. \( \tau \)-transform is obtained from a given 8-bit input, 8-bit output S-box through a lookup table and plays a confusing role. \( L \)-transform is generated through equation (3) and consists of a shift-isoperative operation, \( <<<i \) meaning a 32-bit cyclic shift of \( i \) bits.

\[
L(x) = x \oplus (x <<< 13) \oplus (x <<< 23) \quad (3)
\]

Figure 1. Round Key Generation Algorithm
2.2. Encryption algorithms

The encryption algorithm is shown in Figure 2 and is obtained from the inverse order transformation of the wheel function.

The round function is similar to the key expansion algorithm, where the sender takes 128 bits of information as plaintext input noted as \((X_0, X_1, X_2, X_3)\). Each round of encryption transformation is

\[
X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+4}, r_{ki}) = X \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus r_{ki}), \quad i = 0, 1, 2 \ldots 3
\]

(4)

\(r_{ki}\) is the wheel key generated in Section 1.1.

\[
L(x) = x \land (x <<< 2) \land (x <<< 10) \land (x <<< 18) \land (x <<< 24)
\]

(5)

The synthetic replacement \(T\) is basically the same as the \(T^\prime\) operation in the round key generation algorithm, the difference lies in the \(L\)-transform shift operation, as in equation (4).

The final output ciphertext is noted as \((Y_0, Y_1, Y_2, Y_3)\), which is obtained by the inverse order transformation of the wheel function \((X_32, X_33, X_34, X_35)\)

\[
(Y_0, Y_1, Y_2, Y_3) = R(X_{32}, X_{33}, X_{34}, X_{35}) = (X_{35}, X_{34}, X_{33}, X_{32})
\]

(6)

Figure 2. Wheel function generation algorithm

3. Hardware implementation

3.1. Overall

Figure 3. Block Diagram
The design mainly contains the above four modules: wheel key generation module, memory module, data selector module, and encryption/decryption module. In the encryption operation, the output of the round key generation module is directly sent to the encryption module, and the two are executed in parallel without waiting for the encryption operation. And the whole encryption process is completed with a fixed constant period, which can effectively resist the side channel attack during encryption. In the case of encryption operation, the order of wheel key usage is inverted and requires RAM for temporary storage, and the decryption operation requires more time for data storage than the encryption operation.

3.2. unfolding optimization and critical path optimization

Hardware implementations for key rotation, encryption and decryption in SMS4 algorithm usually have cyclic architecture and pipeline architecture. The round-robin architecture is resource-efficient, but requires 32 rounds to produce results, resulting in a large time loss for communication device applications. The pipeline architecture has the highest throughput, however, the large bit-width and multi-round nature of the encryption algorithm will consume unbearable hardware resources. The literature[9] used a pipeline architecture for synthesis, using 7736 FFs as well as 3213 LUTs, which is no longer possible on some low- to mid-range FPGAs.

This design uses a two-degree unfolding loop structure to shorten the 32 iterations in the algorithm to 16, which will significantly improve the encryption efficiency and achieve a balance between resource utilization and performance.

According to the encryption algorithm above, put substitute(2) into(1) to obtain the flow chart described in the following block diagram

By completing two iterative operations in one clock, the total number of operation rounds will be halved, but this will subsequently lead to an increase in latency and consequently a decrease in the synthesis frequency. Therefore it is necessary to minimize the number of layers in the path.

First of all, for Xi input, set it to , and both then perform the heterodyne operation, which will shorten the delay path by one level, and temp2 will also be used when generating the second level of iteration, and 32-bit heterodyne gate multiplexing will reduce the resource usage and decrease the delay. Also on the output operation of Xi+4, Xi+4 is generated in parallel with the input of the second SBOX, although this brings some additional use of resources, it can significantly reduce the number of delay gate levels and achieve the effect of reducing the delay. Similarly, for the final output at Xi+5, a similar operation is performed.
Figure 5. critical path optimization

The algorithm of the round function generation module is the same as that of the encryption/decryption module, except for the shift operation, so the above optimization is also applied to the round function generation module to achieve the effect of reducing clock cycles and shortening the number of gate stages.

4. Simulation
To verify the correctness of the SMS4 hardware design, simulations were performed using modelsim, and the simulation results are shown in Figure 6. Style and spacing

Plaintext: 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10
Key: 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10
Ciphertext: 68 1e df 34 d2 06 96 5e 86 b3 e9 4f 53 6e 42 46

The correctness of this circuit was verified by comparing the examples in the official documentation. And a total of 17 cycles were used for the operation, verifying the fast nature of this design.

Figure 6. Cryptographic verification

5. Synthesis
FPGA is very helpful for the design verification and synthesis evaluation of digital circuits, this circuit is based on Xilinx Artix-7 xc7a200 for design verification, EDA platform is vivado 2015.4.

Table 1. Design resources, performance comparison

| Design | Device   | LUT    | FF    | Data Throughput |
|--------|----------|--------|-------|-----------------|
| [7]    | Straitx IV | 3213   | 7736  | 52.1Gbps        |
| [8]    | Zynq 7020 | 3703   | 4520  | 800Mbps         |
| [9]    | EP2      | 242 ALMs |     | 142.7Mbps       |
| This design | Artix-7 | 1102   | 267   | 1.3Gbps         |
This design mainly uses one set of two-degree unfolding round key generation circuit, one set of two-degree unfolding encryption circuit, and the S-box design adopts the design method of LUT, consuming a total of 1102 LUTs and 267 FFs, with a throughput of 1.3Gbps. Comparing with similar designs, this design has a greater advantage with similar designs in terms of throughput, and at the same time, due to the adoption of two-degree unfolding mode, the resource occupation, although higher than single-step iteration, but it is still controlled within a reasonable range.

6. Conclusion
In this paper, we propose a circuit design for high-speed SMS4 encryption algorithm, which adopts the second-degree expansion and key road strength optimization techniques to shorten the number of iteration rounds from the previous 32 rounds to 16 rounds, which greatly improves the processing throughput with reasonable resource utilization. The round key generation module runs synchronously with the encryption module and the number of running rounds is constant, which can effectively resist side channel attacks. This design will be applied as an accelerated core on 32-bit CPUs, and in today's world where big data is in every corner of our lives, the need for encryption of the design is more urgent than ever. Therefore this design has the prospect of wide application.

References
[1] Office of State Commercial Cipher Administration .SMS4 cipher for WLAN products[EB].
[2] M. Babu; G. A. Sathish Kumar . In Depth Survey on SMS4 Architecture [C]. 2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW)
[3] Jun Zhao ,Zhichuan Guo,Xuewen Zeng . High Throughput Implementation of SMS4 on FPGA [J] IEEE Access 17 June 2019 88836-88844
[4] Z. Guan, Y. Li, T. Shang, J. Liu, M. Sun and Y. Li, "Implementation of SM4 on FPGA: Trade-off analysis between area and speed", Proc. IEEE Int. Conf. Intell. Saf. Robot. (ISR), pp. 192-197, Aug. 2018.
[5] X. Gao, E. Lu, L. Xian and H. Chen, "FPGA implementation of the SMS4 block cipher in the Chinese WAPI standard", Proc. Int. Conf. Embedded Softw. Syst. Symposia, pp. 104-106, Jul. 2008.
[6] X. Gao, E. Lu, L. Li and K. Lang, "LUT-based FPGA implementation of SMS4/AES/Camellia", Proc. 5th IEEE Int. Symp. Embedded Comput., pp. 73-76, Oct. 2008.
[7] HE Shiyang , Li Hui, LI Fenghua . Optimization and implementation of the SM4 on FPGA. JOURNAL of XiDian University.2021.6
[8] Zhang Lihua,Implementation and Online Verification of SMS4 Algorithm Based on FPGA.Journal of East China Jiaotong University.2018.10 111-116
[9] Zheng Zhaoxia , Zi Yichun, Serialized design of SMS4 and lightweight implement.J. Huazhong Univ. of Sci & Tech.(Nature Science Edition) . 2016, 2, 61-64