A Study Of Iterative-Compression Based Improved AES Key Expansion

Wei Wang [b], Haitao Zhang [a], Haiyan Tan [c], Bin Zheng [d]

Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[a] zxzhanght@sina.com, [b]124838165@qq.com, [c] beauthy@126.com, [d]1530868301@qq.com

Keywords: AES, Iterative-Compression, one-way property, FPGA

Abstract. AES key expansion algorithm doesn't have one-way property, and it can't prevent the sub secret key reverse. So an improved algorithm of AES key expansion based on Iterative-Compression was proposed in this study. The algorithm confused initial key by cyclically shifting, and then the 128-bit key iteration obfuscated compressed into a 32-bit secret key. The first sub key was generated by the key word, and the follow sub keys were extended through recursive computation according to the first sub key. In the end, results of the one-way property analysis and FPGA simulation demonstrate that the one-way property strength of the improved key expansion algorithm is 296. And it meets high efficiency requirement.

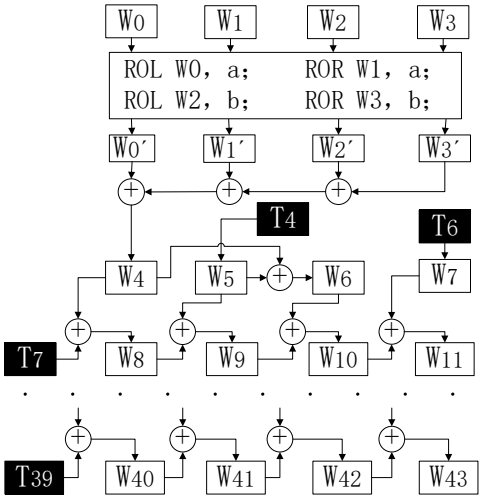
Introduction

Advanced encryption standard (AES) mainly consists of encryption round operation, encryption rounds and key expansion operation [1]. Without of any independence on intermediate variables, the structure of the generation for the word in the key expansion operation is simple and efficient. While there was a great linear among sub words of the AES key expansion algorithm, and the linearity reflected the sub word XOR operation relations. Further, the XOR reflexivity affected the one-way property of the algorithm which is the required number of the former round of the secret key reverse derived the later key [2-4]. It is a great safety threaten for AES key expansion algorithm without one-way property [4]. By taking the AES-128 algorithm as an example, a method based on Iterative-Compression to improve the one-way property in the algorithm of AES key expansion was proposed in this paper. The experiments results show that the algorithm efficiently improved the one-way property to 2^{96}, and the method was efficient.

Improved AES key expansion algorithm

Algorithm description. Compared to the traditional AES, the Iterative-Compression based strategy was used in the first round key expansion to improve the one-way property. A number with the length of k (k<n) will be gotten by iterative calculating the n long number in the strategy. If divided the n long number into j numbers each with the length of k, and compressed the numbers to one in j-1 rounds iterative calculations, the required exhaustive times for the k long number to reverse derive the initial n long number would be $2^{k*(j-1)}$。

In the improved algorithm for the initial key word, the W0 was Left shifted by a bits to get W0′and the W1 was cyclic shifted by a bits to get W1′. And the same calculations were done for W2 and W3 to get W2′and W3′ by b bits under the condition a≠b≠32. Further on, the W4 would be calculated by iteratively calculated W0′、W1′、W2′、W3′ in three rounds. What's more, the W5 would be obtained according to W4 by Subword、 Rotword and Rcon, the W7 would be generated in the same manner of W5 by W6, and the W6 would be gotten by W4 XOR W5. At last, the subsequent operate of key expansion would be similar with the original method.



Step 1:

$W0' \leftarrow ROL\ W0,\ a;$ $W1' \leftarrow ROR\ W1,\ a;$

$W2' \leftarrow ROL\ W2,\ b;$ $W3' \leftarrow ROR\ W3,\ b;$

Step 2:

$W4 = W0' \oplus (W1' \oplus (W2' \oplus W3'));$

$W5 = SubWord(RotWord(W4)) \oplus Rcon(1);$

$W6 = W4 \oplus W5;$

$W7 = SubWord(RotWord(W6)) \oplus Rcon(2);$

Step 3:

For(i = 8; i < 44; i + +)

{AES_Expaneded_Key(Wi);}}

Fig. 1 The diagram of improved AES key expansion algorithm

The Repeating operations of the step 3 would be continued until the 44 words were all generated after step 1 and step 2, and the diagram of the method was shown as Fig.1. The improved algorithm disordered the words of the initial key, so the circumstances that the different initial keys share the same sub key words would be avoided. To some extends, the difficulty increased to reverse derive the initial keys.

One-way property analysis. In the iterative compression strategy, the proportion was 4:1, and the reverse derivation diagram shown as Fig. 2. If the W4 was known, and conjectured the one bit of W0′ as 0 or 1 in two times, so the variant Temp1 could be gotten. In addition, the values of both W0′ and Temp1 could be confirmed in 2^{32} exhaustive operations. In the same way, W1′ and Temp2, and W2′ and W0′ could be confirmed in two 2^{32} exhaustive operations. To conclusion, the 2^{96} exhaustive operations were needed to reversely derivate the first round four words. In order to optimize the algorithm, the same recursive operation mode was employed for the subsequent operations.

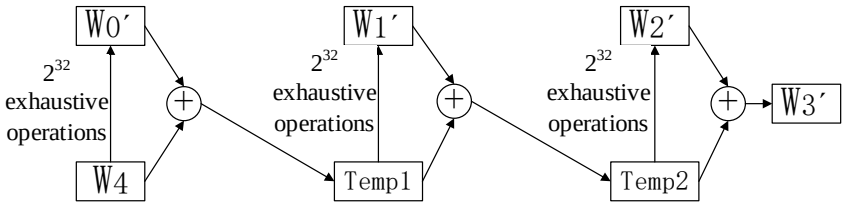


Fig. 2 Logic diagram of iterative compression inverse derivation

The algorithm implementation experiments. The improved algorithm was design in Quartus II, the function and timing simulations were completed in ModelSim, and hardware implementation experiments were conducted in Cyclone IV FPGA [5, 6]. The efficiency of key expansion was taken into consideration, and four parallel 32bits cyclic shift registers were designed in logic aspect. The shift value of the a and the b were 1 and 2, so two clock cycles were consumed in shifting. Taking a data with length of 128bits as an experiment, 128 bit key algorithm based on the improved method was applied. The results were shown as Fig. 3.

key=[3ABCFD60CA6BFDC7C83EBA1B7BDCAF6C]

data_in=[AB4CBCFE3C1BCE6F79FD63BA6CAEBF3C]

encrypt_result128=[CF4BD210B0ADE3BFF16C7EAA7CA8DEA4]

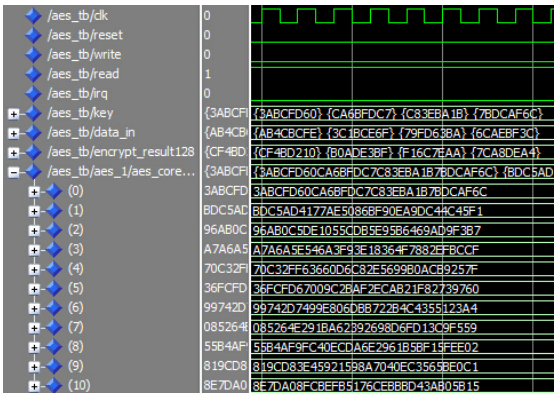


Fig. 3 Simulation results of encryption

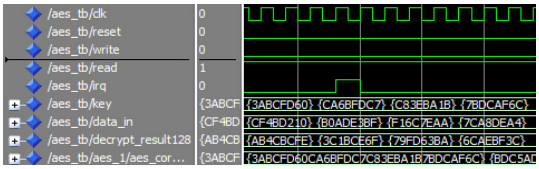


Fig. 4 Simulation results of decryption

In order to test the results, the encryption output value encrypt_result128 was employed as decryption input. The key in encryption was shared in decryption, the results presented in Fig. 4. In the decryption, the sequences of add round key operations were opposite to the sequences in encryption. Add round key operations were executed by the order that beginning at the last round of sub expansion key up to the former.

key=[3ABCFD60CA6BFDC7C83EBA1B7BDCAF6C]
data_in=[CF4BD210B0ADE3BFF16C7EAA7CA8DEA4]
decrypt_result128=[AB4CBCFE3C1BCE6F79FD63BA6CAEBF3C]

The time series consuming of the improved algorithm was analyzed. In the experiments, eight cycles were used up in the first round of key expansion, and five in the second. However, the same operation structure was used in the second round and the later rounds. So, five cycles should be the same time consuming in the later rounds, and the experiments results revealed in Table 1. Original algorithm

Table 1 Time cycles consuming comparison between the two methods

	1	2	3	·····	10	Total
Original Algorithm	5	5	5	·····	5	50
Improved Algorithm	8	5	5	·····	5	53

According to the results, compared to the traditional methods three cycles were increased in the first round expansion for improved method. While, the increased time consuming could be ignored when the system clock with a high frequency as 100MHz, the time consuming difference was too small to find out. The results illustrated that the improved AES key expansion method was efficient and feasible.

Summary

An improved AES key expansion based on iterative compression was proposed in this paper to solve the problems that the AES key expansion one-way strategic was affected by the XOR reflexive and expansion efficiency. The improved method strengthen the key expansion one-way property to 2^{96}, greatly enhanced the security of the initial key in the AES key expansion. Experiments results revealed that the method was safer was safe and efficient.

Acknowledgements

This work was supported by the demonstration project of Chongqing China (No. CSTC2013jcsf10029). Corresponding author: Wei Wang, zxzhanght@sina.com.

References

[1] K. Rahimunnisa, M. Priya Zach, S. Suresh Kumar, J.Jayakumar. Architectural Optimization of AES Transformations Transformations And Key expansion[J]. International Journal on Cryptography and Information Security (IJCIS), 2012,12(3):117-130.

[2] YUAN Wei, ZHANG Yun Ying, HU Liang, LI Hong Tu, WANG Cheng Ming. Structure Cryptanalysis of Rijndael Algorithm [J], Journal of Jilin University (Information Science Edition),2008,26(5):447-493.

[3] Michael Tunstall. Improved "Partial Sums"-based Square Attack on AES [C]// International Confer-ence on Security and Cryptography – SECRYPT. Rome, Italy:INSTICC Press,2012:25-34.

[4] HU Liang ,YUAN Wei ,YU Meng Tao, CHU Jian Feng, LIU Fang. One-way property strategy and improvement of key generation algorithm of Rijndae[J], Journal of Jilin University (Engineering and Technology Edition),2009.39(1):137-142.

[5] D.Rahul Gandh, V.Kamalakannan, R.Balamurugan, S.Tamilselvan. FPGA Implementation of Enhanced Key Expansion Algorithm for Advanced Encryption Standard [C]//International Conference on Contemporary Computing and Informatics (IC3I). Mysore, India: IEEE ,2014: 409-413.

[6] Shun Wen Xiao,Xian Zhi Dai,Huai Bing Qi,Han Kui Liu,Qian Shu Zhang,Yun Xiu Wang. High-Speed Parallel Implementation of AES Key Expansion Algorithm Based on FPGA[J].Applied Mechanics and Materials,2015,3749 (719):712-716.