International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 7, July - 2013

New Proposed Secure Algorithm For Cryptography In FPGA

Revini S Shende
Dept. Of Electronics and Telecommunication
Smt. Kashibai Navale College of Engg, Pune, India

Mrs. Anagha Deshpande
Asst. Prof., Dept. Of Electronics and Telecommunication
Smt. Kashibai Navale College of Engg, Pune, India

Abstract

Lightweight cryptography (LWC) is an emerging research area which has to deal with the trade-off among security, cost, and performance. In this paper we present the idea and list some types of LWC algorithms. Hummingbird is a novel ultra lightweight cryptographic algorithm targeted for devices like RFID tags, smart cards and wireless sensor nodes. The hybrid model of Hummingbird is explained keeping the constraint devices in mind and thus resulting in an easier software implementation. The paper presents the algorithms for the encryption as well as decryption process and shows some simulation results performed on Xilinx.

1. Introduction

Low-cost smart devices like RFID tags and smart cards are rapidly becoming pervasive in our daily life. Well known applications include electronic passports, contactless payments, product tracking, access control and supply chain management just to name a few. But the small programmable chips that passively respond to every reader have raised concerns among researchers about privacy and security breaches. A considerable body of research has been focused on providing RFID tags with cryptographic functionality, while scarce computational and storage capabilities of low cost RFID tags make the problem challenging. This emerging research area is usually referred as LWC which has to deal with the trade-off among security, cost, and performance.[5]

LWC is a branch of modern cryptography which covers cryptographic algorithms intended for use in devices with low or extremely low resources. LWC does not determine strict criteria for classifying a cryptographic algorithms as lightweight, but the common features of lightweight algorithms are extremely low requirements to essential resources of target devices.[1]

Hummingbird is a recently proposed ultra LWC targeted for low-cost smart devices. It has a hybrid structure of block cipher and stream cipher and is developed with both lightweight software and lightweight hardware implementations for constrained devices in mind. The hybrid model can provide the designed security with a small block size and is therefore expected to meet the stringent response time and power consumption requirements for a variety of embedded applications.[4]

Hummingbird is resistant to the most common attacks to block ciphers and stream ciphers including birthday attack, differential and linear cryptanalysis , structure attacks, algebraic attacks, cube attacks, etc

2. Hummingbird Cryptographic Algorithm

Hummingbird is neither a block cipher nor a stream cipher, but a rotor machine equipped with novel rotor-stepping rules. The design of Hummingbird is based on an elegant combination of block cipher and stream cipher with 16-bit block size, 256-bit key size, and 80-bit internal state. The size of the key and the internal state of Hummingbird provides a security level which is adequate for many embedded Applications.

A top-level structure of the Hummingbird cryptographic algorithm is shown in Figure 1.

IJERT

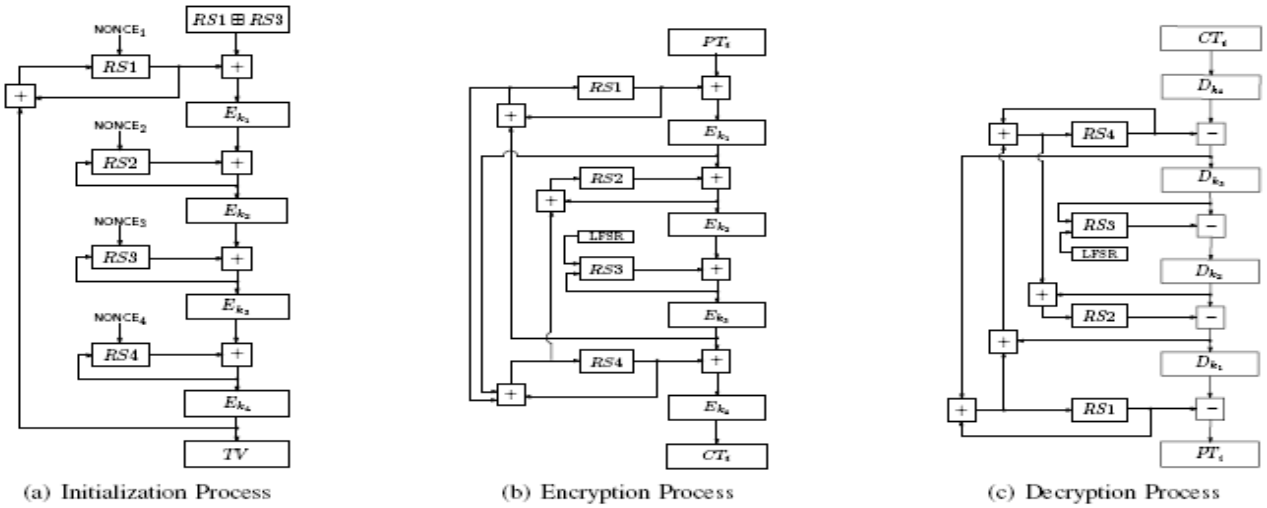


Fig 1. A Top-Level Description of the Hummingbird Cryptographic Algorithm

which consists of four 16-bit block ciphers E_{ki} or D_{ki} (i = 1; 2; 3; 4), four 16-bit internal state registers RSi (i = 1; 2; 3; 4), and a 16-stage Linear Shift Feedback Register (LFSR). Moreover, the 256-bit secret key K is divided into four 64-bit subkeys k_1; k_2; k_3 and k_4 which are used in the four block ciphers, respectively.

The overall structure of the Hummingbird initialization algorithm is shown in Figure 1(a). When using Hummingbird in practice, four 16-bit random nonces $NONCE_i$ are first chosen to initialize the four internal state registers RSi (i = 1; 2; 3; 4), respectively, followed by four consecutive encryptions on the message RS1$\boxplus$ RS3 by Hummingbird running in initialization mode (see Figure 1(a)). The final 16-bit ciphertext TV is used to initialize the LFSR. Moreover, the 13th bit of the LFSR is always set to prevent a zero register. The LFSR is also stepped once before it is used to update the internal state register *RS*3.

The overall structure of the Hummingbird encryption algorithm is depicted in Fig. 1(b). After a system initialization process, a 16-bit plaintext block PT_i is encrypted by first executing a modulo 2^{16} addition of PT_i and the content of the first internal state register *RS*1. The result of the addition is then encrypted by the first block cipher E_{k1} .

The overall structure of the Hummingbird decryption algorithm is illustrated in Figure 1(c). Hummingbird employs four identical block ciphers $E_{ki}(\cdot)$(i = 1; 2; 3; 4) in a consecutive manner, each of which is a typical substitution-permutation (SP) network with 16-bit block size and 64-bit key as shown in the figure 2.

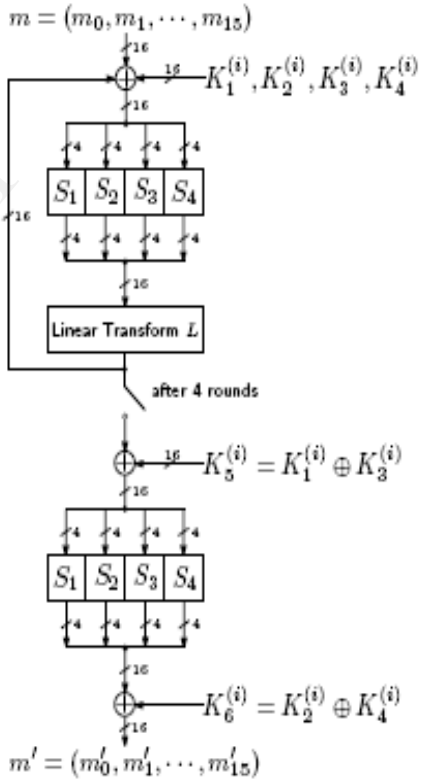


Fig. 2 The structure of block cipher in the Hummingbird cryptography algorithm.

While each regular round comprises of a key mixing step, a substitution layer, and a permutation layer, the final round only includes the key mixing and the S-box substitution steps. The key mixing step is implemented using a simple exclusive-OR operation, whereas the substitution

IJERT

layer is composed of four S-boxes with 4-bit inputs and 4-bit outputs as shown in Table No.2

Table No. 2 S-Boxes Used

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_1(x)$	8	6	5	F	1	C	A	9	E	B	2	4	7	0	D	3
$S_2(x)$	0	7	E	1	5	B	8	2	3	A	D	6	F	C	4	9
$S_3(x)$	2	E	F	5	C	1	9	A	B	4	6	8	0	7	3	D
$S_4(x)$	0	7	3	4	C	1	A	F	D	E	6	B	2	8	9	5

The selected four S-boxes, denoted by $S_i(x) : F_4^2 \rightarrow F_4^2$; $i = 1; 2; 3; 4$, are Serpent-type S-boxes ensure that the 16-bit block cipher is resistant to linear and differential attacks as well as interpolation attack. The permutation layer in the 16-bit block cipher is given by the following linear transform L: $\{0,1\}^{16} \rightarrow \{0,1\}^{16}$ defined as follows:

$$L(m) = m \oplus (m \lll 6) \oplus (m \lll 10),$$

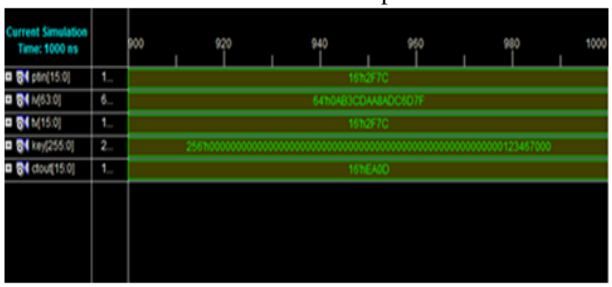
where $m = (m_0;m_1;\ldots\ldots;m_{15})$ is a 16-bit data block.

The block cipher consists of four regular rounds and a final round. The 64-bit subkey k_i is split into four 16-bit round keys $K_1^{(i)}$, $K_2^{(i)}$, $K_3^{(i)}$ and $K_4^{(i)}$ that are used in the four regular rounds, respectively.Moreover, the final round utilizes two keys $K_5^{(i)}$ and $K_6^{(i)}$ directly derived from the four round keys.

This procedure is repeated in a similar manner for another three times and the output of E_{k4} is the corresponding ciphertext CT_i. Furthermore, the states of the four internal state registers will also be updated in an unpredictable way based on their current states, the outputs of the first three block ciphers, and the state of the LFSR.

3. Results

The encryption algorithm was simulated in Modelsim simulator considering 256-bit initialization vector 0ABECDAA8ADC6D7F. For the plaintext of 16-bit 2F7C, cipher text obtained is 16-bit EA0D. The simulation is kept for 1000ns.



4. Conclusion

This paper details about lightweight cryptography and its types and discusses the implementation of ultra lightweight cryptographic algorithm Hummingbird. The security and performance factor is very precisely achieved by the algorithm due to its prominent internal structure.

Compared to other lightweight FPGA implementations of block ciphers XTEA, ICEBERG, SEA, AES, Hummingbird can achieve larger throughput with the smaller area requirement. Consequently, Hummingbird can be considered as an ideal cryptographic primitive for resource constrained environment.

The efficient FPGA implementation of Hummingbird is possible using the given software algorithms so that it can achieve larger throughput with smaller area requirement. Also, Hummingbird can be used in high-security required devices as it is resistant to most cryptographic attacks.

REFERENCES

[1] Sergey Panasenko and Sergey Smagin, "Lightweight Cryptography: Underlying Principles and Approaches", *International Journal of Computer Theory and Engineering, Vol. 3, No. 4, August 2011.*

[2] X. Fan, G. Gong, K. Lauffenburger, and T. Hicks, "FPGA Implementations of the Hummingbird Cryptographic Algorithm", IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2010.

[3] K. Lauffenburger, X. Fan, G. Gong, T. Hicks, "Design Space Exploration of Hummingbird Implementations on FPGA's" Centre for Applied Cryptographic Research (CACR)Technical Reports, CACR-2010-27. http://www.cacr.math.uwaterloo.ca/techreports/2010/cacr2010-27.pdf

[4] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Ultra Lightweight Cryptography Resource- Constrained Devices", 14th International Conference on Financial Cryptography and Data Security- FC 2010

[5] Ismail San, Nuray At, "Enhanced FPGA Implementation of Hummingbird Cryptographic Algorithm", 14th International Conference on Financial Cryptography and Data Security - FC 2010.

[6] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Ultra Lightweight Cryptography for Low- Cost RFID Tags: Hummingbird Algorithm and Protocol," *Centre for Applied Cryptographic Research (CACR) Technical Reports*, CACR 2009

[7] T.Eisenbarth, S. Kumar, C. Paar, A.Poschman, and L. Uhsadel, " A Survey of Lightweight-Cryptography Implementations", IEEE Design & Test of Computers, vol. 24, no.6,pp.522-533,2007

[8] Xilinx Inc., "Spartan-3 FPGA Family Data Sheet", DS099, December 4, 2009, available at http://www.xilinx.com/support/documentation/datasheet/ds099.pdf

IJERT