Ms.K.Durgadevi, Ms. Selvanathiya, Mr.M.Sivasubramanian / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 1, Issue 3, pp.507-512

IMPLEMENTATION OF SECURE MASTER USING MODIFIED TWOFISH ALGORITHM IN FPGA DEVICES

Ms.K.Durgadevi [1]
Assistant Professor

Ms. Selvanathiya [2]
Assistant Professor

Mr.M.Sivasubramanian [3]
Assistant Professor

VELTECH MULTITECH Dr RANGARAJAN Dr SAKUNTHALA ENGINEERING COLLEGE, AVADI, CHENNAI

Abstract

In this paper a novel VLSI architecture of the modified Twofish block cipher is presented. Twofish is one of the most secure cryptographic algorithms. The characteristic features of the Twofish Algorithm are good security margin, fast encryption/decryption in software, moderately fast in hardware and moderate flexibility. Designed Twofish cryptographic algorithm improved the existing MDS block with a MDS-M2 block that improved a process speed, and decreased complexity and power consumption. MDS-M2 block of the designed Two fish cryptosystem that compares to MDS block of a Two fish cryptographic system used in wire communication is a little processed the number of operation and system structure is simple, so a complexity duty is very low. Various applications of this algorithm are Low-end smart cards, Wireless, ATMs, and in Satellite communication.

Keywords: Block cipher, Cryptography, Two fish, MDS matrix, MDS-M2

1. INTRODUCTION

As electronic communications spread to every area of modern life, cryptography has become an essential component to control the authenticity, integrity, confidentiality and non-repudiability of private data that flows through public networks. With increasing performance, requirements and improvements in technology, better-adapted ciphers are making their apparition to replace aging algorithms that have proven to be too weak or too slow for the current applications.

In this project the ALTERA FPGA device was used to implement a promising cipher of this new generation: Twofish. Twofish is a 128-bit cipher that supports keys of length of 128, 192 or 256-bits. For simplicity only a 128-bit key length version was implemented. Software implementations of the algorithm were used for interoperability testing. All the features including sub-key generation, encryption and decryption were implemented.

The DES cryptographic algorithm that was used as a cryptographic standard on establishment in 1977 lost a function as a cryptographic standard because of the development of a platform and the diversification of the communication environment. Therefore, NIST (National Institute of Standards and Technology) conspired AES (Advanced Encryption Standard) and adopted Rijndael cryptographic algorithm as the final cryptographic standard in 2000. However, the reason that Rijndael cryptographic algorithm was adopted as standard is because it compares to other candidate algorithm is a process speed is fast. Rijndael cryptographic algorithm can't be used actively yet, because it cannot carry out an encryption /decryption at the same time and has a lot of trouble on implementation.

What was admitted among five algorithms (Twofish, Rijndael, MARS, RC6, and Serpent) in a safety and performance of cryptosystem is Twofish cryptographic algorithm. It has extensively been used because of the merit that easiness on implementation and a encryption /decryption are able to have been carried out at the same time. Therefore, we improved a process speed of this Twofish cryptographic algorithm in this paper and designed suitable in wireless communication environment.

2. CRYPTOGRAPHY

Cryptography is the technique of sending the data from the sender to the receiver. This is the process of conversion of data into a secret code for the transmission over the public network. The original text is converted into the coded equivalent called “cipher text”.

There are two processes that take place in the cryptographic technology. They are: Encryption, Decryption.

Encryption is the process of obscuring the information to make it unreadable without the special knowledge. In other words it is also described as the conversion of a human readable message, known as either the plain text or the clear text, into the cipher text which unauthorized people are unable to easily understand. This is usually done for the secrecy, and typically for confidential information and the

communications. Encryption is also used for authentication.

The two types of encryption algorithms are 1) Symmetric Encryption 2) Asymmetric Encryption

The symmetric encryption techniques are used for the secured data transformation on the public networks. The traditional symmetric cryptographic systems are based on the idea of a shared secret. In such a system, two parties that want to communicate securely first agree in advance on a single "secret key" that allows each party to both encrypt and decrypt messages.

In an asymmetric algorithm there are two keys namely the public key and the private or secret key.

Decryption is the process of obscuring the information from the unreadable format to the readable format is known as decryption. In other words it is the conversion of cipher text to the plain text.

The key is a binary number that is typically form 40 to 256 bits in length. The greater the number of bits in the key, the more possible key combinations and the longer it would take to break the code. The data is locked or the encrypted by combining the bits mathematically with the data bits. At the receiving end, the key is used to unlock the code and restore the original data.

3. HARDWARE OVERVIEW

Field Programmable Gate Arrays (FPGAs) can be used to implement just about any hardware design. One common use is to prototype a lump of hardware that will eventually find its way into an ASIC. However, there or not it does will depend on the relative weights of development cost and production cost for a particular project. (It costs significantly more to develop an ASIC, but the cost per chip may be lower in the long run. The cost tradeoff involves expected number of chips to be produced and the expected likelihood of hardware bugs and /or changes. This makes for a rather complicated cost analysis, to say the least).

The development of the FPGA was distinct from the PLD/CPLD evolution just described. This is apparent when you look at the structures inside. There are three key parts of its structure: logic blocks, interconnect, and I/O blocks. The I/O blocks form a ring around the outer edge of the part. Each of these provides individually selectable input, output, or bi – directional to one of the general – purpose I/O pins on the exterior of the FPGA package. Inside the ring of I/O blocks lies a rectangular array of logic blocks. And connecting logic blocks to logic blocks and I/O blocks to logic blocks is the programmable interconnect wiring.

The logic blocks within an FPGA can be as small and simple as the macro cells in a PLD (a so – called fine grained architecture) or larger and more complex (coarse grained). However, they are never as large as an entire PLD, as the logic blocks of a CPLD are. Remember that the logic blocks of a CPLD contain multiple macro cells. But the logic blocks in an FPGA are generally nothing more than a couple of logic gates or a look – up table and a flip – flop.

Because of all the extra flip- flops, the architecture of an FPGA is much more flexible than that of a CPLD. This makes FPGAs better in register – heavy and pipelined applications. They are also often used in place of a processor plus software solution, particularly where the processing of input data streams must be performed at a very fast pace. In addition, FPGAs are usually denser (more gates in a given area) and cost less than their CPLD cousins, so they are the de facto choice for larger logic designs.

4. TWOFISH CRYPTOGRAPHIC ALGORITHM

Twofish cryptographic algorithm is a 128 bits block cipher that accepts a variable-length key up to 256 bits. Figure1 shows basic structure of Twofish cryptographic algorithm. Twofish cryptographic algorithm includes a whitening process of input / output and is similar to Feistel network structure of 16 rounds except that output of a f- function is circulated through 1 bit.

128 bits plaintext is divided into four words of each 32 bits using little-endian convention. And, each word passes through input whitening process that is exclusive ORed four units of 32 bits subkey and 128 bits plaintext. Two words of the left side is used as inputs of two g- function inside the F- function in each round. One input word is inputted passes through 8 bits left circulation. A g- function is composed MDS matrix multiplier and S-box to be subordinate to four 8-by-8 bit key-value. Outputs of two g- function combined to use PHT (Pseudo-Hadamard Transform), and two subkeys are added by modulo-2 addition. Two outputs of the F function exchange a position for the following round. The results of the last round exchange a position again after 16^{th} Round. And, it creates a cypertext of 128 bits with the little-endian convention which passes through output of whitening and applied to 128 bits plaintext. A decryption process carry out inversely the process that an order of subkeys applied in an encryption process and an output of a F- function become exclusive-ORed with right two words.

4-by-4 byte MDS matrix multiplication in g-function is used as an important spread mechanism of the Twofish cryptographic algorithm, and a primitive polynomial is same as the equation (1).

$$g(x) = x^8 + x^6 + x^5 + x^3 + x + 1 \qquad (1)$$

The key generation module is used to generate the forty K-subkeys needed by the algorithm. It takes for input even and odd numbers ranging from 0 to 39 and is dependent upon the global user supplied key. As one might notice, it uses the same building blocks as the F-function mentioned earlier. This is important as it allows the use of the same function for both key generation and encryption, thus reducing the hardware required to implement this algorithm.

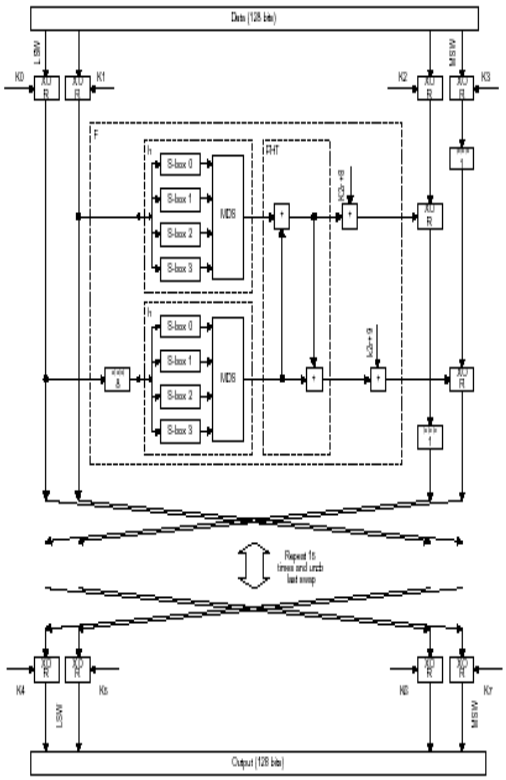

Figure: 1 Overall Structure of Twofish

There is another set of subkeys used in the Algorithm: the S-subkeys. These are computed by means of multiplying an appropriate part of the global key with an RS matrix described in the following section. An important issue regarding the latter is that they are computed once for a particular global key, and stay fixed during the entire encryption and decryption process.

5. DESIGN AND IMPLEMENTATION

A. Design Decisions

The Modified Twofish structure offers a great deal of flexibility in terms of space versus speed tradeoffs. We decided to go for a minimum hardware implementation of the algorithm with hopes of fitting the circuit on an Altera FPGA.

In order to do so, some design decisions had to be made. It was decided that only one h-function, instead of two, would be used for computing the K-subkeys and the encrypted data.

The second design decision regarded the computation of the K-subkeys. The two choices available were to either pre-compute the K-subkeys or store them in a RAM memory (full keying) or to compute the K subkeys on the fly as needed (zero keying). It was decided that zero keying best suited our first design choice as the RAM needed would consume too much space on the FPGA we aimed at using.

B. Building Blocks

1) Q-Permutations

The Q-Permutation is at the core of the design of Twofish. The permutation is executed on a byte of input, which is split in two before being rotated and modified along different paths. The most important operations of the permutation are executed with four lookup tables labeled t0, t1, t2 and t3.

Each lookup table takes 4 bits of input and produces a 4-bit value. Each lookup table thus has 16 entries of 4 bits. There are four lookup tables per q–permutation and two different q-permutations, q0 and q1, each with its set of lookup table. The lookup table could have been programmed in to ROM.

2) S-Boxes

The S-Box operates on a 32-bit word. Each byte of the word passes through three Q-Permutations. The output of each bank of q- permutations is then recombined into a word and XOR-ed with a 32-bit value. These two values are derived from the key material and Twofish's s-boxes are thus referred to as "key-dependent" s-boxes.

3) Maximum Distance Separable Matrix

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 01 & EF & 5B & 5B \\ 5B & EF & EF & 01 \\ EF & 5B & 01 & EF \\ EF & 01 & EF & 5B \end{pmatrix} \bullet \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

The Maximum Distance Separable (MDS) Matrix is a 4x4 matrix of bytes that multiplies a vector of four bytes. Multiplications are carried out in the

Galois Field GF (2^8) with the primitive polynomial $x^8 + x^6 + x^5 + x^3 + 1$.

Each byte is converted into a polynomial in which each power *p* of x is present only if the *p*-th bit is 1. A multiplication in GF amounts to a multiplication of polynomials followed by a division by the primitive polynomial. The result is converted back to a bit vector by setting a bit to 0 if the corresponding power of x has an odd coefficient and 1 otherwise (modulo 2 division).

In this case the computations are fairly straightforward since there are only three coefficients: 0x01, 0xEF and 0x5B. The result of a multiplication can be reduced to a series of XOR's for each bit of the output. For example, multiplying '*a*' by 5B results in byte '*b*':

b0 = a2 xor a0
b1 = a3 xor a1 xor a0
…
b7 = a7 xor a1

4) Reed-Solomon Matrix

$$RS = \begin{pmatrix} 01 & A4 & 55 & 87 & 5A & 58 & DB & 9E \\ A4 & 56 & 82 & F3 & 1E & C6 & 68 & E5 \\ 02 & A1 & FC & C1 & 47 & AE & 3D & 19 \\ A4 & 55 & 87 & 5A & 58 & DB & 9E & 03 \end{pmatrix}$$

The Reed-Solomon (RS) matrix is similar to the MDS matrix. In this case the multiplication is executed between an 8x4 matrix and a vector of 8 bytes. The computations are done in GF (2^8) with a different prime polynomial: $x^8 + x^6 + x^3 + x^2 + 1$.

Unlike the MDS matrix, the RS matrix has a large number of different multiplicands. In order to minimize the resources used, a series of XOR's was derived for each of the 23 multiplicands to give equations similar to those seen in the MDS matrix.

These equations were not given and had to be derived. This was done by first computing a general equation for the multiplication of two polynomials in GF (2^8). The equation was then used to compute each 23 cases. Once well understood the process was used with a minimum risk of error by inserting the constants into the derived equations using a text editor.

5) Operation Selector

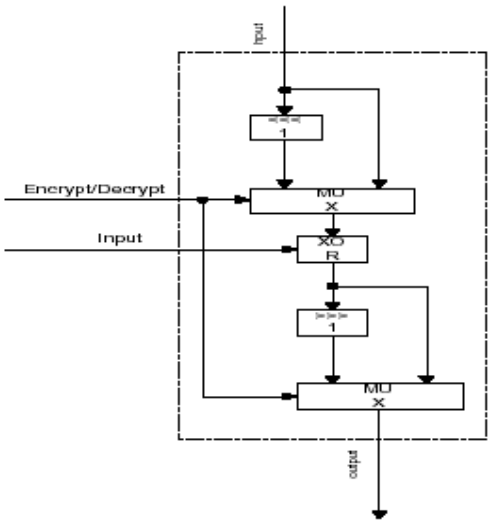

Figure 2: Building Block for Operation Selection

Twofish is a very symmetric algorithm. Encryption and decryption can be executed with almost all the same pieces of hardware. The first difference is that the sub-keys must be used in reverse order. The second difference is at the output stage of the round.

The output stage consists of a rotation and an XOR. The output stages are mirrors of each other and can thus be constructed by implementing each path

6. MDS-M2 FOR PROCESS SPEED DEVELOPMENT

When MDS block is implemented in Twofish cryptographic algorithm, it takes the most a lot of process time. MDS block is existing in a F- function and must carry out multiplication operation like the equation (2). According to bottleneck phenomenon occurs owing to the multiplication operation in a MDS block which is important block in Twofish cryptographic algorithm, it takes a lot of process time.

We used MDS-M2 (Maximum distance separable modulo-2) in order to control an increase of process time by this multiplication operation in this paper and reduced data process time.

We transformed MDS algorithm of the equation (2) into the equation (3).

$$\begin{aligned} Y_0 &= 01 \cdot X_0 + EF \cdot X_1 + 5B \cdot X_2 + 5B \cdot X_3 \\ Y_1 &= 5B \cdot X_0 + EF \cdot X_1 + EF \cdot X_2 + 01 \cdot X_3 \\ Y_2 &= EF \cdot X_0 + 5B \cdot X_1 + 01 \cdot X_2 + EF \cdot X_3 \\ Y_3 &= EF \cdot X_0 + 01 \cdot X_1 + EF \cdot X_2 + 5B \cdot X_3 \end{aligned} \qquad (3)$$

Because a result value of operation becomes self-itself, we omitted operation in equation (3). So, it is same as the equation (4).

$$
\begin{aligned}
Y_0 &= EF \cdot X_1 + 5B \cdot X_2 + 5B \cdot X_3 \\
Y_1 &= 5B \cdot X_0 + EF \cdot X_1 + EF \cdot X_2 \\
Y_2 &= EF \cdot X_0 + 5B \cdot X_1 + EF \cdot X_3 \\
Y_3 &= EF \cdot X_0 + EF \cdot X_2 + 5B \cdot X_3
\end{aligned} \qquad (4)
$$

If we unfold to use modulo-2 for Ox5B and OxEF of MDS, it can express like equation (5) and (6).

$$
\begin{aligned}
y_7 &= x_7 \oplus x_1 \\
y_6 &= x_6 \oplus x_0 \\
y_5 &= x_7 \oplus x_5 \oplus x_1 \\
y_4 &= x_6 \oplus x_4 \oplus x_1 \oplus x_0 \\
y_3 &= x_5 \oplus x_3 \oplus x_0 \\
y_2 &= x_4 \oplus x_2 \oplus x_1 \\
y_1 &= x_3 \oplus x_1 \oplus x_0 \\
y_0 &= x_2 \oplus x_0
\end{aligned} \qquad (5)
$$

$$
\begin{aligned}
y_7 &= x_7 \oplus x_1 \\
y_6 &= x_7 \oplus x_6 \\
y_5 &= x_7 \oplus x_6 \oplus x_5 \oplus x_1 \\
y_4 &= x_6 \oplus x_5 \oplus x_4 \oplus x_1 \\
y_3 &= x_5 \oplus x_4 \oplus x_3 \oplus x_0 \\
y_2 &= x_4 \oplus x_3 \oplus x_2 \oplus x_1 \\
y_1 &= x_3 \oplus x_2 \oplus x_1 \oplus x_0 \\
y_0 &= x_2 \oplus x_1 \oplus x_0
\end{aligned} \qquad (6)
$$

A multiplication operation using module-2 is easily solved like the equation (5) and (6). If it combines equation (5) and (6) with one equation, it is same as the equation (7). The ‘a’ means Ox5B term and ‘b’ means OxEF term.

$$
\begin{aligned}
y_{7a} &= y_{7b} = x_7 \oplus x_1 \\
y_{6a} &= x_6 \oplus x_0 \\
y_{5a} &= x_{7a} \oplus x_5 \\
y_{4a} &= x_{6a} \oplus x_4 \oplus x_1 \\
y_{3a} &= x_5 \oplus x_3 \oplus x_0 \\
y_{2a} &= x_4 \oplus x_2 \oplus x_1 \\
y_{1a} &= x_3 \oplus x_1 \oplus x_0 \\
y_{0a} &= x_2 \oplus x_0
\end{aligned}
\qquad
\begin{aligned}
y_{6b} &= x_7 \oplus x_6 \\
y_{5b} &= y_{6b} \oplus x_5 \oplus x_1 \\
y_{4b} &= y_{4a} \oplus x_5 \\
y_{3b} &= y_{3a} \oplus x_4 \\
y_{2b} &= x_4 \oplus y_{1b} \\
y_{1b} &= x_5 \oplus y_{0b} \\
y_{0b} &= y_{0a} \oplus x_1
\end{aligned} \qquad (7)
$$

If the equation (7) transforms into the equation to correspond to least term, it is same as the equation (8).

$$MDS(x_i, y_{ia}, y_{ib}) \quad i = 0, \ldots, 7 \qquad (8)$$

i has range between 0 and 7 in equation (8). The equation (8) is the final function for multiplication operation about MDS. If it fixes with look-up table, it has decreased stage operation and decreases total process time than the existing MDS.

7. CONCLUSION

A standard of the cryptosystem that is suitable for wireless communication environment is a process speed. Therefore, the Rijndael cryptographic algorithm that is the most suitable cryptographic algorithm in AES standard, but it is hard to popularize because of disadvantages that can't carry out encryption / decryption concurrently and has other weak.

If Twofish cryptosystem compares to Rijndael cryptosystem, its speed drops a bit than Rijndael. Therefore, we designed MDS-M2 which can carry out encryption/ decryption concurrently and can improve a speed of the Twofish cryptosystem that implementation was easy. Twofish Cryptosystem algorithm by the simulation results confirmed a performance improvement of 11% than the existing Twofish cryptographic algorithm.

Therefore, we considered the security system that the Twofish cryptographic system that used an improved MDS-M2 block is very suitable for wireless communication environment.

REFERENCES

1. Bruce Schneier, Applied Cryptography, 2nd Edition, John Wiley & Sons, 1996.

2. Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson. Twofish: A 128-Bit Block Cipher, 1998.

3. Pawel Chodowice, Kris Gaj, "Implementation of the Twofish Cipher Using FPGA Devices", Technical Report, George Mason University, 1999.

4. Yeong-Kang Lai, Liang-Gee Chen, Jian-Yi Lai, “VLSI architecture design and implementation for twofish block cipher” IEEE International Symposium on Circuits And Systems, Vol.2, pp 356-359, 2002

5. William Stallings, “Cryptography and Network Security: Principles and Practice” Published by Prentice Hall, Edition 2006

6. Swinder Kaur, Prof. Renu Vig “Efficient implementation of AES algorithm in FPGA device” International conference on computational intelligence and multimedia applications, pp 179-187, May 2007

7. Chi-Wu Huang, Chi- Jeng Chang, Mao- Yuan Lin, Hung-Yung Tai "Compact FPGA implementation of 32 bits AES algorithms using block RAM " IEEE Conference on computer society, Vol. 10, pp 1-4, October 2007

8. Chi-Wu Hung, Chi-Jeng Chang, Mao-Yuan Lin, Hung-Yun Tai "FPGA implementation of 128 bits AES algorithm based on four 32 bits parallel operations" The First International Symposium on Data and Privacy, pp 462-464, November 2007

9. Ahmed Rady, Ehab EL Sehely, A.M.EL Hennawy,"Design and Implementation of area optimized AES algorithm on reconfigurable FPGA" IEEE ICM pp 35-38, December 2007

10. National Institute Standards & Technology, http://csrc.nist.gov /encryption/aes/

11. J. Daeman, V. Rijnmen, "AES" proposal: Rijndael, http: //csrc.nist.gov/encryption/aes