International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Efficient Hardware Encryption Using Lightweight Process

Deeksha Bhardwaj[1], Rohit Pawar[2], Aman Jain[3]

[1, 2, 3]Department of Computer Engineering, SavitribaiPhule Pune University, G.H.R.I.E.T., Wagholi, Pune, India

Abstract: ***Providing security by software encryption has many vulnerabilities. Hence, hardware cryptography is becoming more prominent. Hence, we propose a novel approach in lightweight hardware cryptography which provides higher throughput, faster execution, lower memory consumption and reliable security. We are using the hybrid encryption system which is a combination of PRESENT, HUMMING-BIRD, TWOFISH algorithm and GRP. In this paper we enhanced the existing cryptosystem with variable length key size and also keep the sender and receiver unaware of the generated key used for encryption, hence providing maximum security.***

Keywords: Lightweight Cryptography, Block- cipher, Encryption, PRESENT, HUMMINGBIRD, TWOFISH.

1. Introduction

In recent years, information security through encryption has not only become an essential feature but an integral component of any application especially in mobile computing and embedded applications, which are easy to hack, given their limited hardware resources to support the heavy requirements of software security applications. Hence, hardware-based security models are being rapidly embraced by the industry due to their lightweight and robust architecture and low-level implementation. Thus, there is an enormous scope inthe field of hardware cryptography with many popular techniques like DES, AES, CLEFIA, Hummingbird, PRESENT,TWOFISH etc. being used today. Each have their pros and cons.

In our system, we use components of three prominent techniques, PRESENT algorithm, TWOFISH algorithm, HUMMINGBIRD and bit permutation instruction GRP (Group operation) which is widely studied and researched. All modern block ciphers consist of networks called SP Networks. These consist of two instructions:
Substitution: A binary word is replaced by some other binary word. We call their circuits S-boxes.

Permutation: A binary word has its bits reordered(Permuted). We call their circuits P-boxes. In our system, we take the data as plaintext and encrypt it using the S-box of TWOFISH and HUMMINGBIRD algorithm. We also have a pattern generator which generates a variable sized key based on an algebraic series. The key is used by the receiver to decrypt the cipher text to retrieve the original document.

2. Literature Survey

Hardware cryptography has seen a lot of resurgence in the recent years. This has led to many techniques being used. Some of the popular algorithms are: DES, AES, Hummingbird, CLEFIA, PRESENT, etc. Most of these hardware-based block ciphers use universal Gates to implement their algorithm. The main criterion for a lightweight cipher is to consume less memory and hence use less number of Gate Equivalents (GEs) for an efficient hardware implementation without compromising the requirements of strong security properties.

Ciphers like DES and AES have a large number of GEs. Hence, they are not feasible for small scale real time applications. The following table gives a comparison of various algorithms on the basis of their GEs.

Table 1: Comparison of various Lightweight Ciphers

Lightweight Algorithm	Block Size	Key length	Gate Equivalents
HIGHT	64	128	3048
mCrypton	64	64 96 128	2420 2681 3758
SEA	96	96	3758
TEA	64	128	2355
ICEBERG	64	128	7732
CLEFIA	128	128	2488
PRESENT	64	128	1884

In addition to having large GEs, these algorithms also pose a risk of their keys being predictable due their fixed size. This makes them open to Brute force attacks as the fixed size key reduces the domain of searches.

3. Proposed System

Our system attempts to resolve most of the drawbacks of the afformentioned existing systems. This is done by our approach in fusing the best components of these algorithms and integrating them into our system. In our system, the P-box we use for permutation of the given text is the same P-box we use in PRESENT algorithm. The S-box we use is a combination of the Hummingbird and Twofish algorithms. These algorithms are discussed in more detail in the Algorithm Description topic in this paper.

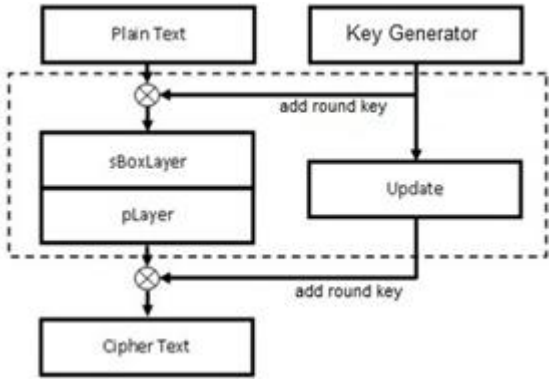


Figure 1: Block Diagram of the Proposed System

In our system, the user first registers itself to the interface. Then logs in to input data as plaintext. The motive here from the user is to act as a sender to send data in plaintext securely using our system. Also, to encrypt and decrypt data, the user must enter the size of the key which will be used by the receiver to decrypt the data. The key can be of any size which is a factor of the size of the plaintext. Hence, our system allows variable key sizes. Most of the algorithms described in the literature survey use fixed sized keys. Variable key size adds to the confusion property of the cipher.

As shown in Fig. 1 here, the key pattern will be generated by a Key Generator based on a pattern applied to the plaintext and the text will be encrypted using the S-box and P-box of the algorithms mentioned above.

4. Algorithm Description

a. PRESENT Algorithm

PRESENT cipher is a hardware-optimized ultra-lightweight block cipher that has been designed with area and power constraints. PRESENT is an example of SPN structure. It has 64 bits block size, 80 or 128 bit key size with 31 rounds. Here S-box is 4 bits and used 16 times in one round.
Each round consists of the following 3 steps:

1) AddRoundKey: Key XORed with cipher.
2) Substitution: Used 4 bits S-box.
3) Permutation: Used P-layer.

b. Hummingbird

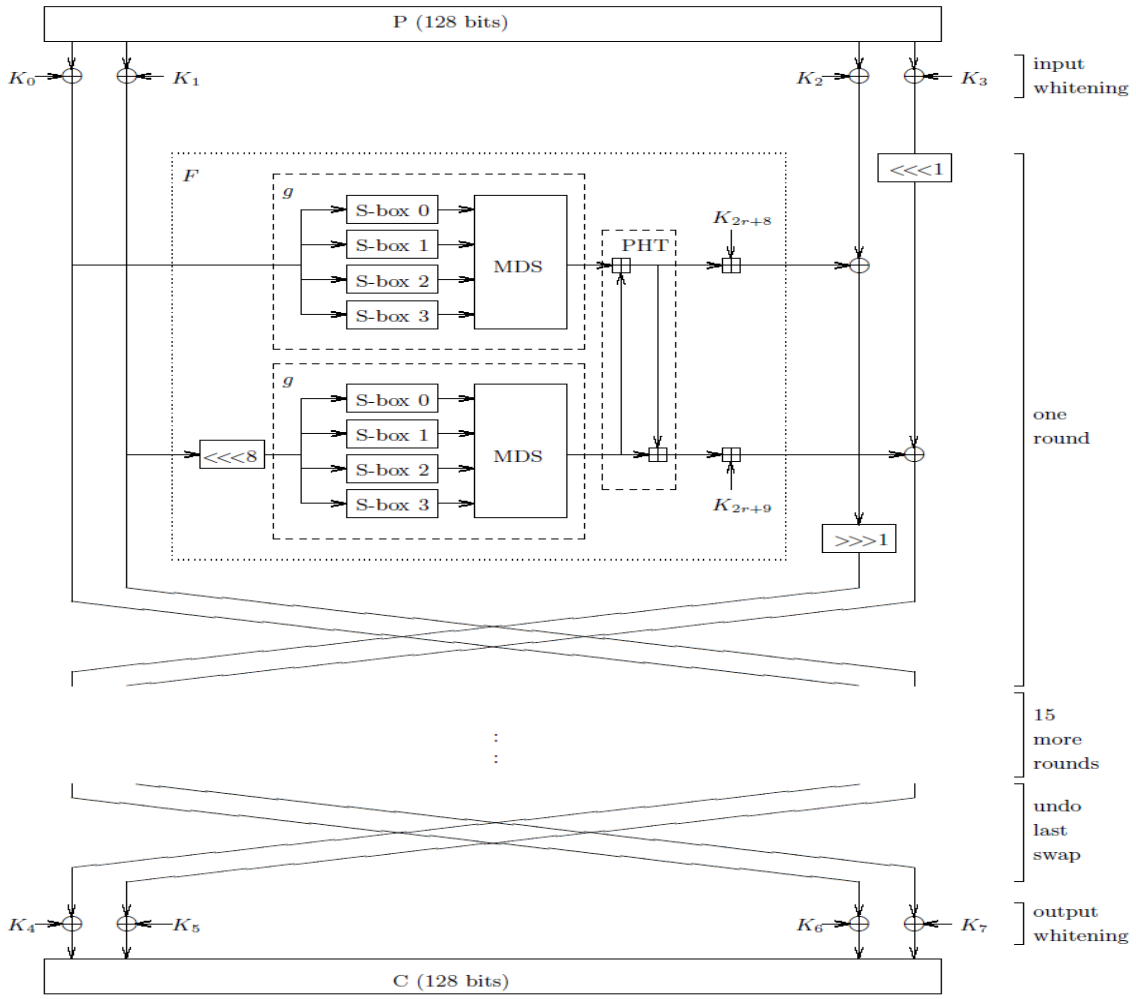
In order to overcome these security issues a new algorithm called Hummingbird has been designed as a mutual authentication algorithm. Which is a combination of both block cipher and stream cipher. This is designed with a small block size and expected to meet stringent response time and power consumption requirements. This consists of 16-bit block size, 256-bit key size and 80-bit internal state where the key size provides security for various RFID applications.

c. TwoFish Algorithm

Twofish is a 128-bit block cipher that accepts a variable-length key up to 256 bits. The cipher is a 16-round Feistel network with additional whitening of the input and output.

The plaintext is split into four 32-bit words in the input whitening step, these are xored with four key words.
In each round:

1) The two words on the left are used as input to the g-function.The g function consists of four byte-wide key-dependent S-boxes, followed by a linear mixing step based on an MDS matrix.
2) The resultsof the two g functions are combined using a Pseudo-HadamardTransform (PHT), and two keywords are added.
3) These two results are then xored into the words on the right (one of which is rotated left by 1 bit _rst, the other is rotated right afterwards).
4) The left and right halves are then swapped for the next round.
5) After all the rounds, the swap of the last round is reversed, and the four words are xored with four more key words to produce the ciphertext.



Function F:

- The function F is a key-dependent permutation on 64-bit values.
- It takes three arguments, two input words R0 and R1, and the round number r used to select the appropriate subkeys.
- R0 is passed through the g function, which yields T0. R1 is rotated left by 8 bits and then passed through the g function to yield T1.
- The results T0 and T1 are then combined in a PHT and two words of the expanded key are added.

T0 = g(R0)
T1 = g(ROL(R1; 8))
F0 = (T0 + T1mod 232 +K2r+8
F1 = (T0 + 2T1mod 232 +K2r+9
where (F0; F1) is the result of F.

Function g:

- The function g forms the heart of Twofish.
- The input word X is split into four bytes.
- Each byte is run through its own key-dependent S-box.
- Each Sbox is bijective, takes 8 bits of input, and produces

8 bits of output.
four results are interpreted as a vector of length 4 over GF(28), and multiplied by the 4*4 MDS matrix (using the field GF(28) for the computations). The resulting vector is interpreted

- as a 32-bit word which is the result of g.

$$
\begin{aligned}
x_i &= \lfloor X/2^{8i} \rfloor \bmod 2^8 \qquad i = 0, \ldots, 3 \\
y_i &= s_i[x_i] \qquad i = 0, \ldots, 3 \\
\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} &= \begin{pmatrix} \cdot & \cdots & \cdot \\ \vdots & \text{MDS} & \vdots \\ \cdot & \cdots & \cdot \end{pmatrix} \cdot \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \\
Z &= \sum_{i=0}^{3} z_i \cdot 2^{8i}
\end{aligned}
$$

wheresi are the key-dependent S-boxes and Z is the result of g.

Pseudo-HadamardTransforms:-
A pseudo-Hadamard transform (PHT) is a bitwise rotations operation, and a carefully designed key schedule. Given two inputs a and b, the 32-bit PHT is defined
as:
a0 = a + b mod 232
b0 = a + 2b mod 232

Twofish uses a 32-bit PHT to mix the outputs from its two parallel 32-bit g functions. This PHT can be executed in two

opcodes on most modern microprocessors, including the Pentium family.

3. Conclusion

This paper proposes a novel approach by introducing a hybrid system in term of memory requirements that is best suited for lightweight cryptographic design. In the future, we would like to design the system with even more confusion properties to make it even harder to decrypt without authorization. We would also like to increase the scope of the project by including more type of files as input with unlimited size. We can also increase the robustness and scalability of the algorithm by making the pattern generator use different patterns for the same input. The extensibility of our system is a key in making these enhancements possible. In conclusion, we would like to say that we have proposed a cipher which is not only unbreachable but robust, scalable, lightweight and can prove revolutionary in hardware encryption, setting new standards in cryptography.

4. Acknowledgment

We are working on this project under continuous and insightful guidance of Mrs. DeekshaBhardwaj, Assistant Professor (HOD Computer Department) at G. H. Raisoni Institute of Engineering & Technology, Wagholi, Pune. We would also like to thank the B.E. Projects coordinator Mr. Mahesh Bhandari, Asst. Professor, Computer Department, G.H. Raisoni Institute of Engineering and Technology, Wagholi, Pune, for his constant support and constructive criticism.

References

[1] "The 128 bit block cipher CLEFIA: Algorithm specification." On-line document, 2007. Sony Corporation.
[2] National Institute of Standards and Technology, "Data Encryption Standard (DES)," *FIPS 46-3*. Available via http://csrc.nist.gov, October1999.
[3] X. Yang and R. B. Lee, "Fast subword permutation instructions using omega and flip network stages," In *Proceedings of the International Conference on Computer Design*, pages 15-22, September 2000.
[4] C.E. Shannon, "Communication Theory of Secrecy Systems," Bell System Technical Journal. A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultralightweight block cipher," In *CHES*, Vol. 4727 of *LNCS*, pages 450-466. Springer, 2007.
[5] B. Preneel, V. Rijmen, A. Bosselaers, \Recent Developments in the Design of Conventional Cryptographic Algorithms," *State of the Art and Evolution of Computer Security and Industrial Cryptography, Lecture Notes in Computer Science*, B. Preneel, R. Govaerts,J. Vandewalle, Eds., Springer-Verlag.
[6] Shun-Lung Su, Lih-ChyauWuu, and Jhih-Wei Jhang, "A New 256-bits Block Cipher –Twofish 256", Computer Engineering & Systems, International Conference *in IEEE, 2010*.