Cryptography for Resource Constrained Devices: A Survey

Jacob John
Dept. of Computer Engineering
Sinhgad Institute of Technology
Pune, India.
jj31270@yahoo.co.in

***Abstract*— Specifically designed and developed cryptographic algorithms, which are suitable for implementation in resource constrained devices such as RFID systems, smart cards and wireless sensor networks are called light weight cryptographic algorithms. In this paper a survey is done on the selected light weight cryptographic algorithms. The light weight cryptographic algorithms are of two types, block ciphers and stream ciphers. Algorithms under both these categories are presented in this paper. Security features and performances of hardware implementations of these algorithms are also analyzed.**

Keywords -resource constrained devices; cryptography; light weight; block cipher; stream cipher

I. INTRODUCTION

The demand for applications involving resource constraint devices like RFID systems, smart cards and wireless sensor networks are growing. RFID systems are used in identification of objects like products or animals. An RFID system consists of a tag which is attached to the object and reader is used to access data from the tag. The communication between the reader and tag should be protected from unauthorized access. Wireless sensor network consists of a number of sensor nodes, which operate without human intervention for a long period of time with little energy supply. Security should be provided to the data or information being transmitted among the sensor nodes. Due to mass deployment of these devices, there are security concerns about the data being handled by these devices. The traditional cryptographic algorithms cannot be implemented on these devices because resource constraint devices have limited processing power, limited energy supply and limited memory size. So there was a need, to develop specifically designed cryptographic algorithms for these low resource devices. These specifically designed algorithms are called light weight cryptographic algorithms, which are of two types: block ciphers and stream ciphers. The block ciphers are AES [5], DESL [4], HIGHT [3] and PRESENT [2]. The stream ciphers are Trivium [6] and Grain [7]. Humming bird-2 [1] is having the properties of both. In this paper a brief over view of the above mentioned algorithms is given. The paper also analyses the security systems and hardware implementations of these algorithms.

The brief descriptions of the algorithms are given in section 2. The security analysis is discussed in section 3. Section 4 contains analysis of hardware implementations of the light weight algorithms. Then the paper is concluded in section 5.

II. ALGORITHMS

A. AES

Advanced Encryption Standard is symmetric encryption algorithm which was selected by National Institute of Standards Technology (NIST) as a standard cryptographic algorithm in 2001. AES operates on blocks of data called state that have a fixed size of 128 bits. The state is formed as a matrix with four rows and four columns of bytes. The key size is 128 bit. AES encrypts a block by applying the same round function. The round function iterations process the state by applying, non linear, linear and key related transformations. Each transformation converts 128 bit state into a modified 128 bit state. Every byte of the state matrix is processed by applying the following transformations.

Sub Bytes :-substitutes each byte of the state. This operation is non linear. It is also called S-Box operation.

Shift Rows :- rotates each row of the state by the row index. For example first row is not rotated, last row is rotated 3 bytes to the left.

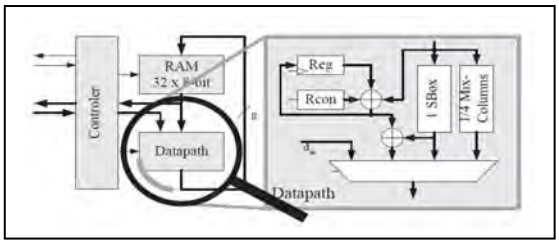

Figure 1 Architecture of AES [5]

Mix Columns :- transforms columns of the state. It is a multiplication by a constant polynomial in an extension field of GF(2^8).

Add Round Key :- combines the 128 bit state with a 128 bit round key by adding corresponding bits mod 2.

The calculation of the 128 bit round keys is done by applying the key schedule function. The architecture of the 8 bit AES module is shown in the figure 1. This is the smallest hardware implementation of the AES algorithm. There are three parts a controller, RAM, and a data path. The controller communicates with other modules to exchange data and controls the ten rounds of AES encryption. It addresses the RAM and generates control signals for the data path. The RAM stores 128 bit state and 128 bit round key.

B. DESL

DESL is the light weight extension of DES. DESL uses single S-Box repeatedly 8 times, which decreases its chip size. DESL resists linear and differential cryptanalysis with the help of S-Box. It encrypts 64 bit plain text in 144 clock cycles. At an operating frequency of 100KHz it consumes current 0.89 µA. The area requirement is 1848 GEs. DESL architecture consists of five main modules : mem_left, mem_right, keyschedule, controller and S-Box.

Controller module manages all control signals and key schedule generates round keys. Key schedule contains an input multiplexer and output multiplexer to select right part of round key. Mem_left consists of eight 4- bit registers. These registers are made using D flip flops. Mem_right consists of eight 4-bit wide registers but it has 6 bit wide output. S-Box module consists of 8 S-Boxes of DES algorithm and an output multiplexor. In DESL there is only one S-Box which is more resistant against differential and linear cryptanalysis. In DESL S-Box is repeated 8 times. Design of DESL is same as DES except IP , IP^{-1} and S-Box.

C. HIGHT

HIGHT(HIGH security and light weight) is a 64 bit block cipher with 128 key size. A group of researchers from Korea University, National Security Research Institute (NSRI) and Korea Information Security Agency (KISA) developed HIGHT in 2006. It uses generalized Fiestel structure with 32 rounds containing simple operations such as XOR, modular addition in the group of 2^8 elements, and bitwise rotation. It requires area of 3048 GEs.

The HIGHT algorithm consists of 32 rounds with initial and final transformation before and after the round function respectively. The plain text P and cipher text C are split into 8 bit blocks P_0 .. P_7 and C_0 .. C_7. The 128 bit master key is split into 8 bit blocks of K_0 .. K_{15}. The initial transformation uses the four whitening key bytes to transform plain text into first round of round function. Output of the final round of the round function is processed with four whitening keys to produce the cipher text.

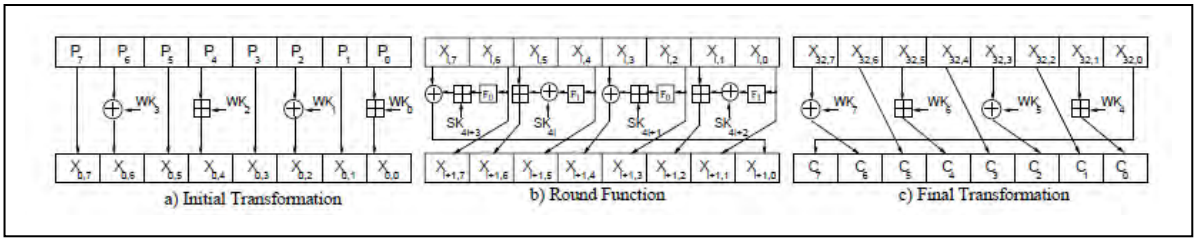

Figure 2 Algorithm HIGHT

The round function uses two auxiliary functions F_0 and F_1. The two functions F_0 and F_1 provide bitwise diffusion which is similar to a transformation in GF. The 7 bit constants are generated by 7 bit LFSR h. Sub keys are generated by processing master key bytes with constants. These sub keys are processed in round function when plain text moves from one transformation to another in the round function rounds.

D. PRESENT

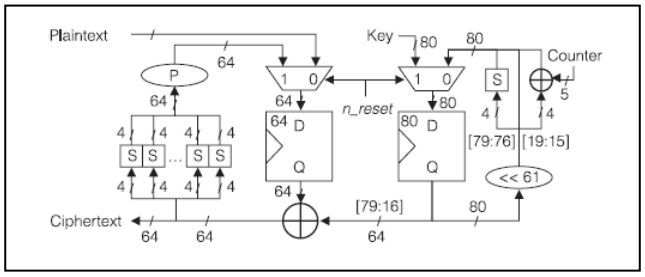

Figure 3 Data path of PRESENT implementation [2]

PRESENT is a 31 round cipher with substitution layer and permutation layer. The block size of PRESENT is 80 bit and key size is 80 bit. It was designed as an ultra light weight block cipher by its authors A. Bogdanov et al. The initial state of PRESENT is $c_{63}\ldots c_0$. The round function transforms its state in three stages addRoundkey, sBoxLayer and pLayer. The addRoundkey operation XORs the current state $c_{i,63}\ldots c_{i,0}$ with the i^{th} round key.

$RK_i = rk_{i,63}..rk_{i,0}$ as follows

$c_{i,j} \leftarrow c_{i,j} \oplus rk_{i,j}$

where $0 <= j <= 63$

The second layer is the non linear sBoxLayer which consists of 4 bit to 4 bit S-Box. The linear bit permutation layer is the third stage, which is shown below.

$c_i \leftarrow c_{4*i}$, $c_{i+16} \leftarrow c_{4*i+1}$

$c_{i+32} \leftarrow c_{4*i+2}$, $c_{i+48} \leftarrow c_{4*i+3}$

where $0 <= i <= 15$.

The round key is the 64 most significant bits of current state of key register K. The round key RK_{i+1} for the next round is generated by shifting the key register to the left by 61 bits and passing the left most 8 bits through two S-Boxes of PRESENT. The 5 bits K_{66} .. K_{63} are XORed with 5 bit round counter.

E. Grain

Grain is a stream cipher designed by Martin Hell, Thomas Johansson and Willi Meier. This stream cipher is designed in such a way that the hardware implementation is very easy and it requires comparatively less chip area. Operations in a stream cipher consist of two phases.

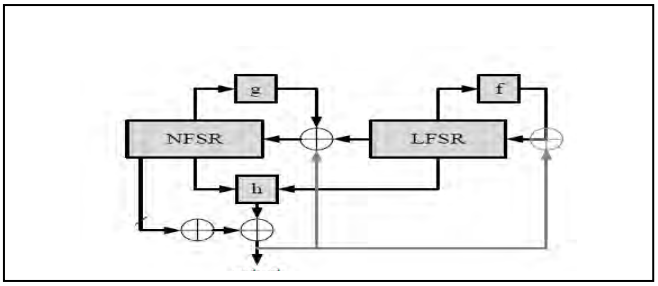

Figure 4 Grain stream cipher

The first phase is initialization of the internal state using the secret key and the IV. After that the state is updated repeatedly and key stream bits are generated. The main components of the stream cipher Grain are two 80 bit shift registers. Out of these one has a linear feedback (LFSR) and the other a non linear feedback (NFSR). The key size is specified with 80 bits and additionally an initial value of 64 bits is required.

The basic structure of the algorithm is shown in the figure 4. The two polynomials of degree 80, f(x) and g(x) are used as feedback function for the feedback shift registers LFSR and NFSR. The output function h(x) uses as input selected bits selected bits from both feedback shift registers. Additionally, seven bits of NFSR are XORed together and the result is added to the function h(x).This output is used during the initialization phase as additional feedback to the LFSR and NFSR. During normal operation this value is used as key stream output.

F. Trivium

Trivium is a hardware oriented synchronous stream cipher developed by Christophe De Canniere and Bart Preneel. This is an example of a stream cipher with simple design without compromising on security. From an 80 bit key and an initial value (IV) of 80 bits, it generates 2^{64} bits of key stream. The state consists of 288 bits which are denoted as $s_0, s_2, \ldots, s_{287}$. The pseudo code is shown below which shows how the algorithm uses 15 specific bits of the state to generate three variables which are used to update the state and which produce one bit

of the output. During the initialization, the key and the IV are loaded to the state and the same update function is applied 1152 times without using the output for the key stream. In the algorithm, N is used for the number of output bits of the stream cipher.

for i=0 to N-1 do

$a_0 = s_{65} + s_{92}$

$a_1 = s_{161} + s_{176}$

$a_2 = s_{242} + s_{287}$

$out_i = a_0 + a_1 + a_2$

$a_0 = a_0 + s_{90}.s_{91} + s_{170}$

$a_1 = a_1 + s_{174}.s_{175} + s_{263}$

$a_2 = a_2 + s_{285}.s_{286} + s_{68}$

$(s_0, s_1,\ldots, s_{92}) = (a_2, s_0, \ldots., s_{91})$

$(s_{93}, s_{94},\ldots, s_{176}) = (a_0, s_{93}, \ldots., s_{175})$

$(s_{177}, s_{178},\ldots, s_{287}) = (a_1, s_{177}, \ldots., s_{286})$

end for

G. Humming bird-2

Humming bird-2 does not come under the category of block cipher or stream cipher, but is having the properties of both. The block size of Humming bird-2 is 16 bit, which is suitable for RFID devices or wireless sensors because it handles only small messages. Humming bird-2 optionally produces an authentication tag for each message. The key size of The Humming bird-2 is 128 bit. Its internal state R, with size 128 bit, is initialized using 64 bit Initialization Vector IV. Accessing of these variables is done as vectors of 16 bit words. The operations in Humming bird-2 are exclusive OR, addition modulo 65536 and non linear mixing function f(x) which are performed on 16 bit words.

The nonlinear mixing function f(y) is computed using the following operations.

$S(y) = S_1(y_0) \mid S_2(y_1) \mid S_3(y_2) \mid S_4(y_3)$

$L(y) = y \oplus (y <<< 6) \oplus (y <<< 10)$

$f(y) = L(S(y))$.

where S(y) indicate computation of four S- Boxes and L(y) is the linear transformation.

A 16-bit keyed permutation WD16 is formed using the following expression

$WD16(y, a, b, c, d) = f(f(f(f(y \oplus a) \oplus b) \oplus c) \oplus d)$

Encryption of a single word of plain text to cipher text is done in four invocations of WD16. After encryption or decryption of each word, follows state updating. Authenticated Encryption with Associated Data is a method in Humming bird -2, which authenticates any associated data that travels with cipher text. Processing of associated data happens only after entire encrypted payload has been processed. Communication of data without message expansion is better for messages with size less than 16 bits. If g is a short message with size 15 bits or less the cipher text message is derived from the n least significant bits of $g \oplus E(0)$. The state is further updated by E (g) for message integrity.

III. SECURITY ANALYSIS

In AES, plain text is processed with non linear, linear and key dependant transformations. These transformations make a good defensive mechanism against various security attacks. The single s-box of DESL is resistant against Differential Cryptanalysis [8] and Linear Cryptanalysis [9].

The constants generated by the linear feedback shift register of HIGHT increases the randomness of the sub keys. It also provides good resistance against slide attack [11]. The final transformation and round function resist the differential and linear cryptanalysis. Related key attacks [10] are resisted by strong key schedule.

The combination of sBoxLayer and pBoxLayer in PRESENT provides good defensive mechanism against differential cryptanalysis and linear cryptanalysis. Linear cryptanalysis of PRESENT requires more than 2^{84} plain text/cipher texts. Such data requirements are impossible to achieve. PRESENT is also resistant to related key attacks and slide attacks. Round – dependant keys are used in PRESENT so that the sub key sets cannot be slid and a non linear operation is used to mix the contents of the key register. The bitwise operations of PRESENT resist structural attacks like integral attacks [12] and bottleneck attacks [13].

Humming bird-2 can resists security attacks like Differential Cryptanalysis and Linear Cryptanalysis. It is found from the security analysis that Humming bird-2 is resistant against linear cryptanalysis up to 12 rounds

of *f*. The resistance against related key attacks is done by four rotations of the initialization phase. The algebraic degree and branch number of S-Boxes thwart different forms of algebraic distinguishing attacks.

IV. HARDWARE IMPLEMENTATION

Light weight version of AES encrypts a 128 bit block of data in 1016 clock cycles at an operating frequency 100 KHz. It has a current consumption of 8.5 μA on a 0.35 μm CMOS process. The required area is 3595 gate equivalents (GEs). DESL requires 144 clock cycles with operating frequency 100 KHz to encrypt 64 bit plain text. The power consumption is 0.89 μA and the throughput is 44.4 Kbps. The required chip size is 7392 transistors (1848 gate equivalents). Hardware implementation of PRESENT requires area 1570 GE and power 5 μA to encrypt a 64 bit plain text with 80 bit key. The throughput of HIGHT is 188.20 Kbps at an operating frequency of 100 KHz, which requires area 3048 GEs. The process used in the hardware implementation of Humming bird-2 is TSMC 0.13 μm which requires 3220 GEs. The encryption of 16 bit word is done in 4 clock cycles. The hardware requirement of Trivium is 2599 GEs and Grain is 1294 GEs. The results of hardware implementations of all the above discussed algorithms are shown in the below given table.

TABLE I. COMPARISON OF PERFORMANCES OF LIGHT WEIGHT CIPHERS

Cipher	***Key size***	***Block size***	***Cycles per block***	***Throughput at 100KHz(Kbps)***	***Logic process***	***Area(GEs)***
Humming bird-2	128	16	4	400.00	0.13 μm	3220
Block Ciphers						
PRESENT	80	64	32	200.00	0.18 μm	1570
HIGHT	128	64	34	188.20	0.25μm	3048
DESL	184	64	144	44.40	0.18 μm	1848
AES	128	128	1016	12.59	0.35 μm	3595
Stream Ciphers						
Trivium	80	1	1	100.00	0.13 μm	2599
Grain	80	1	1	100.00	0.13 μm	1294

V. CONCLUSION

Security features and hardware implementations of some efficient light weight cryptographic algorithms are discussed in this paper. It is found that, compared to stream ciphers block ciphers are more suitable for resource constrained environments. The block ciphers AES, DESL, HIGHT, PRESENT provide good security. The throughput of HIGHT is higher than other block ciphers, which shows its processing speed is faster. The hardware resource requirement of PRESENT is comparatively less, which is only 1570 GEs. Humming bird-2 is having the properties of block cipher and stream cipher. Another advantage of Humming bird-2 is that its power consumption is low and speed of processing is faster.

REFERENCES

[1] Daniel Engels, Marakku-Juhani O. Saarinen, Peter Schweitzer, Eric M. Smith "The Hummingbird-2 Light weight Authenticated Encryption Algorithm" RFID Sec 2011.The 7th workshop on RFID Security and Privacy, Amherst, Massachusets , USA June 2011.

[2] A. Bogdanov et al., "PRESENT: An ultra-lightweight block cipher," in CHES 2007, ser. LNCS, vol. 4727. Springer, 2007, pp. 450–466.

[3] Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S; Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee. "HIGHT: A New Block Cipher Suitable for Low-Resource Device," In L. Goubin and M. Matsui, editors, Proceedings of CHES 2006.

[4] G. Leander, C Paar, A. Poschmann, and K Schramm "A Family of Lightweight Block Ciphers Based on DES Suited for RFID Applications". In A. Biryukov, editor,Proceedings of FSE 2007, LNCS, Springer-Verlag.

[5] M. Feldhofer, S. Dominikus, and J.Wolkerstorfer."Strong Authentication for RFIDSystems Using the AES algorithm". In M. Joye and J.-J. Quisquater, editors, Proceedings of CHES 2004, LNCS, volume 3156, pages 357–370, Springer Verlag, 2004.

[6] C. D. Canniere, B. Preneel TRIVIUM Specifications. eSTREAM, ECRYPT Stream Cipher Project Report 2005/030, April 2005.

[7] M. Hell, T. Johansson, W Meier. Grain – A Stream Cipher for Constrained Environments. eSTREAM, ECRYPT Stream Cipher Project Report 2005/010, 2006 Revised version.

[8] E. Biham, A. Shamir. "Differential Cryptanalysis of the Data Encryption Standard," Springer-Verlag,1993.

[9] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," Advances in Cryptology - EUROCRYPT'93, T. Helleseth, Ed., LNCS 765, Springer-Verlag, pp. 386-397, 1994.

[10] E.Biham. " New Types of Cryptanalytic Attacks Using Related Keys". In T.Helleseth, editor, Proceedings of Eurocrypt '93, LNCS, volume 765, pages 398-409, Springer-Verlag , 1994.

[11] A. Biryukov and D Wagner. "Advanced Slide Attacks". In B . Preneel , editor, Proceedings of Eurocrypt 2000, LNCS, volume 1807, pages 589-606, Springer – Verlag 2000.

[12] L R Kundsen and D Wagner , "Integral Cryptanalysis". In J.Daemen and V.Rijmen, editors, Proceedings of FSE 2002, LNCS , volume 2365, pages 112-137, Springer – Verlag 2002.

[13] H. Gilbert and M. Minier. "A Collision Attack on 7 Rounds of Rijndael". In Proceedings of Third Advanced encryption Standard Conference, National Institutes of Standards and Technology, 230-241, 2000.