ISSN 2395-1621

Cryptographic Techniques and Lightweight Encryption Design

[#1]Shalini, [#2]Vidya Lokhande, [#3]Ravinder, [#4]Ashwini Veer
[#5]Prof. Ms. Nivedita Kadam

[1]shaliniaug941@gmail.com
[2]vidyalokhande23@gmail.com
[3]decentravi94@gmail.com
[4]veer.ashwini8@gmail.com

[#1234]Student, Department of Computer
[#5]Assistant Professor, Department of Computer

Savitribai Phule Pune University
G.H. Raisoni College of Engineering and Management, Pune, India.

ABSTRACT

Lightweight Cryptography is a relatively young scientific sub-field that is located at the intersection of cryptography with computer science and concentration on new designs, changes or applications of cryptographic primitives and procedures in efficient ways. Due to harsh cost limitations and a very strong assailant model especially noteworthy is the possibility of physical attacks there is an snowballing need for lightweight security explanations. Every designer working on lightweight cryptography has to cope with the trade-off between security, costs, and presentation. The focus of lightweight cryptography is on studying new algorithms to overcome the problems that occur mostly in standard cryptographic algorithms as these can be too big, too slow or too energy-consuming. Operation of lightweight cryptography should be virtually light as a feather and must hit the perfect balance in providing security, low power ingesting, higher throughput and compactness. In this project, it's absolutely based on symmetric key encryption, the generic security of lightweight structures, deliberations on block cipher, block size and key size.

Keywords: **lightweight cryptography, encryption, decryption, block cipher, block size, key size, S-box**

ARTICLE INFO

Article History

Received: 13th May 2016

Received in revised form :

13th May 2016

Accepted: 16th May 2016

Published online :

17th May 2016

I. INTRODUCTION

Currently, the questions about security have been increased due to use of prevalent and common devices in the field of digital electronics. After all, information is an organization's most critical treasure and no framework security controls are 100% effective. As the priority and the value of traded data over the Internet or other media types are booming, the hunt for the best solution to provide the necessary protection against the sensitive and important data, thieves' attacks along with furnishing these services under timely manner is one of the utmost active subjects in the security related communities. In a layered security model, it is essential to focus one final prevention control wrapped around sensitive material that is encryption. Efforts to protect the systems and networks attempt to achieve three outcomes: data obtainability, integrity, and discretion. In embedded submissions, implementing a fully developed cryptographic environment would not be empirical because of the restrictions like power dissipation, area and cost. Due to these limitations, the focus is on using lightweight cryptography that needs as minimum memory space as possible. The major facet of lightweight cryptography is to undertake the security-efficiency trade-offs deep-rooted in applications of cryptographic procedures.

Cryptography is the art of turning approximately readable into something untidy, and then being able to undo it.that is, it is a method of accruing and data in a particular form so that only those for whom it is planned can read and process it.it employs complex mathematics and logic to plan strong encryption systems. Cryptography is important because it prohibits criminals from larceny of the information online. When we see a website with HTTPS protocol permitted, that is cryptography at labor. It is also at exertion when you log onto a Wi-Fi hotspot or encrypt a file.

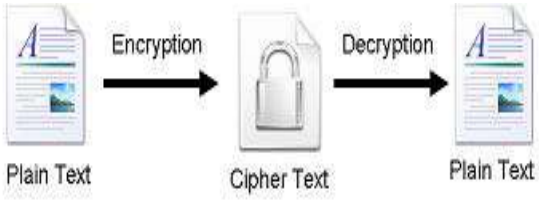

Fig 1. Encryption and decryption process

The main criterion for the lightweight cipher is to have less memory space and that which would result into a less Gate Equivalent (GEs) count for an efficient hardware implementation without compromising the requirement of strong security properties. An ISO/IEC standard on lightweight cryptography requires that the design be made with 1000-2000 gate equivalents (GEs). For security applications, total GEs available would be approx2000-3000. Block ciphers should be limited to less GEs in command to fit in lightweight submissions.

II. SOFTWARE BASED AND HARDWARE BASED ENCRYPTION

Software-Based Encryption

Software encryption packages can help to protect records and provide a good major line of security but they are vulnerable to a number of decryption attacks. It shares computers assets to encrypt data with extra packages on the computer and Uses the user's password as the encryption key that creates confusion and makes it difficult to decrypt the data. It can require software updates. Receptive to brute force attacks, computer tries to bound the number of decryption attempt but hackers can admittance the computer's memory and reset the attempt counter. Software based encryption can be economical and profitable in small application environments and it can be implemented on all types of media.

Hardware-Based Encryption

Uses a committed processor physically located on the encrypted drive and the Processor contains a random numeral generator to produce an encryption key, which the user's password will unlock, it shield keys and critical security parameters within crypto-hardware. Here the Authentication takes place on the hardware circuit. It is profitable in average and bigger application environments and easily adjustable. Here the Protection against the most common attacks, such as cold boot attacks, petty code and brute force attacks, Hardware-based encryption offers a stronger protection in contradiction of the threat replicas, and is now accessible on a changed generation of transportable data security and verification devices.

III. TECHNIQUES

DES (data encryption standard)

For more than three decagon, the Data Encryption Standard (DES) was one the most Broadly used cryptographic procedures [3][4]. It is still the domineering block cipher for banking applications. It was first published in 1977.it is a symmetric block cipher. Its block size is 64 bit and key length is 56-bits.it was mainly developed for government communication. DES was originally practical only in hardware implementations.it uses fiestal network that is it divides the block into two halves earlier working over the encryption steps. Maximum amount of data that can be shift with a single encryption key is 32 GB.

DES I challenge- it took 85 days for the attackers to crack the message encrypted using DES I.
DES II challenge – it took 3 days for the attackers to crack the message encrypted using DES II.
DES III challenge- it took 22 hours and 15 minutes for the attackers to crack the message encrypted using DES III.

AES (advanced encryption standard)

It was published in year 2001 and is more mathematically well-organized and well-designed cryptographic algorithm [2]. Its main strength rests in the option for various key lengths. It allows choosing 128-bit, 192-bit, or 256-bit key. It is exponentially stronger than the 56-bit key.it uses Permutation-Substitution that is, it involves a series of substitution and permutation phases to form the encrypted block. Substitution is simply a mapping of one value to another and permutation is the re-ordering of the bit positions for each of the inputs. There can be 2^128, 2^192, 2^256 combinations of the key. Maximum amount of data that can be transferred with a single encryption key is 256 billion gigabytes.

RSA

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman announced asymmetric cryptographic algorithm. Asymmetric is nothing but there are two different keys. This is also termed as public key cryptography, because of these public key can be gives to everybody. The additional key must be kept private. by using RSA procedure, we can use the private key of the source to sign the plaintext, public key of the receiver to encrypt. For the receiver, he can use private key to decrypt and the public key of the source to verify the sign. RSA algorithm is extra superior to the DES algorithm. Because the RSA algorithm can allocate encryption key, it is also very calm to update the encryption keys, and for the dissimilar communication objects, just have the decryption keys secret.

Hummingbird Algorithm

Hummingbird is a brand-new acutely light cryptographic algorithm intended for resource required plans such as Radio Frequency Identification (RFID) tags, smart cards and wireless sensor nodes.it is a hunt algorithm used by Google. AES algorithm takes more time as compared to hummingbird algorithm. So hummingbird algorithm encrypts and decrypts the bits faster.

Name of the cryptographic algorithm	Key size	No of bits applied	Total no of clock cycles Required to encrypt and decrypt are
AES	128	128	1616
16 BIT HUMMINGBIRD	64	16	4
256 BIT HUMMINGBIRD	1024	256	16

CLEFIA

CLEFIA is a elixir block cipher algorithm. Its name is taken from the French word clef, meaning "key". The chunk size is 128 bits and the key size can be 128 bit, 192 bit or 256 bit. It is advised to be used in DRM systems. It is one of the cryptographic techniques referred as candidate for government use by the japanese.

Key size	Block size	Structure	Rounds
128,192,256	128	Fiestal network	18,22,26

PRESENT

PRESENT [1][6][8]is an engineered cipher whose S-box is the most compact substitution box among all the light variants and has good linear and differential properties. PRESENT's S-box results in a very compact implementation that consumes merely 21 GEs for a single 4 bit S-box. RAM and Flash memory requirements for PRESENT-GRP implementation results in very less bytes as compared to other lightweight algorithms and even with PRESENT individually. The theorem which shows the effect of differential cryptanalysis on S-box of PRSENT is that Any 5 differential features of PRESENT has a minimum number of 10 active S-boxes" and results from papers [5] shows that PRESENT has very good and compact S-box. There are 16 S boxes of PRESENT which are divided into four collections. From papers, the characteristics of S-box are outlined below:

1. Input bit to an S-box comes from 4 well defined-boxes of the same group.

2. The input bits to a collection of four S-boxes come from 16 dissimilar S-boxes.

3. The four output bits from a specific S-box arrive into four well defined S-boxes, each of them belongs to a distinct collection of S-boxes in the following round.

4. The output bits of S-boxes in distinct groups will be fed to distinct S-boxes.

IV. RELATED WORK

There are many new symmetric ciphers. For example, Hight, Clefia, DESXL, and Present with special implementation properties proposed. High was designed with good hardware performance in mind. Provide hardware statistics for a one round implementation, that is, one round is done in one cycle and they arrange that Hight is well seemly for ubiquitous work out plans such as wireless sensor nodes and RFID tags. Hight requires slightly the same chip size as the Advanced Encryption Standard AES algorithm but is much faster. However, figures for applications with a smaller footmark in hardware are not yet accessible. Clefia was designed with a broader request range in mind that is to perform well in both hardware and software applications.

GRP is most difficult bit permutation instructions that make it an obvious choice to be used in cryptographic environment. GRP performs n bit permutation with log2(n) steps while other instructions take O(n) steps [7]. Research in this field and papers [10] have shown the increased strength of cipher RC5 by introducing GRP instructions. GRP scales very efficiently to 2n bits on n bit system by using instruction Shift right pair instruction (SHRP) in PA, RISC and in IA-64 processors. Table look up is the second option to bit permutation instructions, but it is slower as it takes 16 cycles on a superscalar processor for the scheduling of permutation instructions, while GRP does it in only 8 cycles. By loading control bits, GRP requires 13 numbers of instructions while a table lookup needs 31 numbers of instructions. Bit permutation instructions are difficult sub word permutation that makes them best fit in make safe an situation. The use of bit permutation instructions like GRP and OMFLIP is beneficial to design effectual and secure ciphers. Ciphers such as DES, and TWO-FISH uses bit Permutation instruction in its action. Bit permutation instructions are very actual in gaining diffusion operation.

V. PROPOSED SYSTEM MECHANISM

Cyber-attacks are continuously emerging, so security specialists stay busy in the lab concocting certain new schemes to have them at bay. Whether it's protecting our email communications or kept data, some kind of encryption must be included in the lineup of security tools. Successful attacks on victims show that no algorithm 100 percent bulletproof, but without it, we would be offering up convenient access to our data. Because embedded systems have limited computing assets and strict power requirements, script software for embedded devices is a very specific field that requires knowledge of both hardware components and programming. Embedded system security is the discount of vulnerabilities and protection against threats in software successively on embedded devices. Here, we propose a system whose aim is to provide high level confidence to critically important emails that is to textual data by means of applying lightweight encryption methodologies. System will use hardware based encryption for the embedded security to overcome the problems that occur in most conventional software based encryption techniques.

Our focus is on maintaining the time and space complexity of the algorithm and most importantly to reduce the gate equivalents (GE) to make encryption process as efficient and fast as possible. Here we are going to introduce dynamic key generation system which is missing in most of the lightweight encryption algorithms. Proposed system uses symmetric key cryptosystem. That is, the key will be generated from the data itself and it will be a dynamic activity. Basically, we are going to design a circuit using logic gates(mainly EX-OR gate) that will be responsible for calling the key generator and combining the key with the plain text to generate the cipher text(encrypted data).the circuit will then be simulated into a software based circuit using the same logic gates but this time using a programming language.

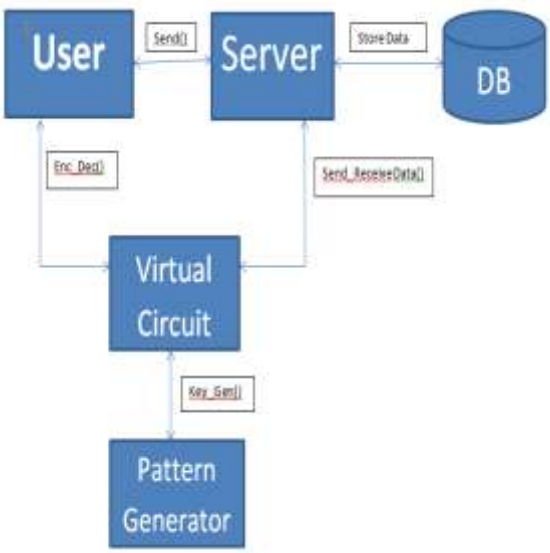

Fig 2. Functional architecture

Block ciphers is to be used in the proposed system. GRP (group operation) implementation and use of s-box (required for the substitution in the bits) for creating the obscurity in the data is being employed in the proposed system so that the potential attacker gets confused during the cryptanalysis and fails to recognize the secret key and decrypt the encrypted data back to the plain-text. Also, the proposed system employs Bit permutation instructions which increases strength of a block cipher by allowing them to perform any arbitrary permutations efficiently with 'log(n)' steps as compared to 'n'that performs fast bit permutation.

Working of proposed system:

Working of the proposed system starts with the signing up of user account on the web application named SMT. SMT stands for Secure Mail Transfer used for sending and receiving e-mails. Once registration gets completed, users can perform operations according to their need. Web application allows the users to type the message and send it after encrypting it and also receive the encrypted message and decrypt it accordingly by using the virtual circuit. The virtual circuit is designed using the EX-OR logic gates. It takes the help of pattern generator to generate a fixed pattern by using mathematical functions and generates the secret key from the data to be encrypted dynamically. Then the circuit takes the responsibility to combine the original message with the secret key to produce the cipher text. The encrypted message is then sent to the server and then the server sends that message to receiver side. After that the circuit running on the receiver side will again generate the key and decrypt the message. Here we are using symmetric key cryptography, that is same key is used to encrypt and decrypt the message. Following are the step by step snapshots of working of our system.

We are transferring input data from 10 layers of circuit blocks. When input data pass from all 10 layers after that we get a secure cipher text. The processing of data is in sequential manner. All this computing is done in background process.

VI. ALGORITHM

Input: A 16-bit data block m = (m0;m1; · · ·,m15) and a 64-bit subkey ki such that
subkey ki = K(i)1‖K(i)2‖K(i)3‖K(i)4
Output: A 16-bit date block m′ = (m′0;m′1; · · · ;m′15)
1: for j = 1 to 4 do
2: m ← m ⊕ K(i)
j [key mixing step]
3: A = m0‖m1‖m2‖m3; B = m4‖m5‖m6‖m7
C = m8‖m9‖m10‖m11;D =m12‖m13‖m14‖m15
4: m ← S1(A)‖S2(B)‖S3(C)‖S4(D)
[substitution layer]
5: m ← m ⊕ (m ≪ 6) ⊕ (m ≪ 10)
[permutation layer]
6: end for
7: m ← m ⊕ K(i)1 ⊕ K(i)3
8: A = m0‖m1‖m2‖m3; B = m4‖m5‖m6‖m7
C = m8‖m9‖m10‖m11;D = m12‖m13‖m14‖m15
9: m ← S1(A)‖S2(B)‖S3(C)‖S4(D)
10: m′ ← m ⊕ K(i)2⊕ K(i)4
11: return m′ = (m′0;m′1; · · · ;m′15)

VII. RESULT

Above given snapshot is the overview of our GUI which is nothing but SMT i.e. Secure Mail Transfer. This is the login page of our website from where user can login into his account by entering User Id and Password.

This is our GUI for registration of user. If user has already registered then he can directly login into his account by simply entering User Id and Password. After that he can send and receive messages. If user is not registered then here he can register also.

This is our sender view in which we can send messages by selecting multiple users.

This is the receiver view in which user will receive multiple messages at a time and decrypt it by clicking on decrypt button. After decrypting user will get original message in the dialog box which sender has been send.

This is the contact page of our GUI where users can also contact to the service provider if they will face any problem regarding to website. Here user can also give its feedback to the service provider and share his views.

VIII. CONCLUSION

This has given a broad view of Lightweight Cryptography. The paper gives the survey on various lightweight data encryption techniques and deep explanation of how we have implemented the encryption technique through system architecture we have given above. This paper tells how to add cryptographic strength to the cipher and also reduce the memory requirements and the power and how we can provide stronger and reliable security to the highly confidential and private information being transmitted between various applications. Like, bases of military, other government communications, email communication etc. This paper tried to present a fair comparison between the most common and used algorithms in the data encryption field. Since our main concern here is the performance of these algorithms under different settings, the presented comparison takes into consideration the behaviour and the performance of the algorithm when different data loads are used.

IX. ACKNOWLEDGMENT

We would like to thank our guide and various technological experts who researches about lightweight encryption techniques and improve the result by implementing new methods. We would also like to thank Google for providing details on different issues on cryptography and about other related areas and detailed information of encryption and decryption.

REFRENCES

[1]"Implementation of a New Lightweight Encryption Design for Embedded Security", Gaurav Bansod, Nischal Raval, Narayan Pisharoty, 2014.

[2] NIST (National Institute of Standards and Technology), "Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication 197, November 2000.

[3] National Bureau of Standards (NBS), "Data Encryption Standard (DES)," Federal Information Processing Standards Publication 46-2, December 1993.

[4] National Institute of Standards and Technology, "Data Encryption Standard (DES)," FIPS 46-3. Available via http://csrc.nist.gov, October 1999.

[5] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A Survey of Lightweight Cryptography Implementations," IEEE Design & Test of Computers – Special Issue on Secure ICs for Secure Embedded Computing, 24(6): 522-533, November/December 2007.

[6] A. Bogdanov, G. Leander, L.R. Knudsen, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT - An Ultra-Lightweight Block Cipher," In P. Paillier and I. Verbauwhede, editors, Cryptographic Hardware and Embedded Systems — CHES 2007,
number 4727 in Lecture Notes in Computer Science, pages 450-466. Springer-Verlag, 2007.

[7] Z. Shi and R. B. Lee, "Bit permutation instructions for accelerating software cryptography," In Proceedings of the IEEE International Conference on Application Specific Systems, Architectures and Processors (ASAP 2000), pages 138-148, July 2000.

[8] A. Bogdanov, G. Leander, L.R. Knudsen, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT - An Ultra-Lightweight Block Cipher," In P. Paillier and I. Verbauwhede, editors, Cryptographic Hardware and Embedded Systems-CHES 2007, number 4727 in Lecture Notes in Computer Science, pages 450-466.
Springer-Verlag, 2007.

[9] 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.

[10] Zhijie Jerry Shi, "Bit Permutation Instructions: Architecture, Implementation and Cryptographic Properties", Princeton, June 2004.

