A Lightweight Paradigm for Security in Bluetooth

Mrs.Sandhya S
Research Scholar,
Visvesvaraya Technological University,
MCA Department Research Center, RVCE,
Bangalore, India

Dr.Sumithra Devi K A
Prof. & Director,
Department of MCA,
RVCE,Bangalore,India

Abstract— **In traditional Bluetooth communication, 128-bit of symmetric stream based encryption is used. In an earlier paper, we had done the comparison of AES-Blake Algorithm and the hybrid encryption method (Triple DES-Tiger, by Patheja, Akhilesh and Sudir [18]). Bluetooth technology is being adopted fast and there are many tiny devices which have come in the market now. It may not be possible to implement the encryption method which used AES as these devices may not have the necessary hardware configurations to support the AES approach. So, it is important to see if there could be an alternative approach which does not use AES and which is also lighter compared to AES. In this paper, we evaluate the feasibility of using PRESENT algorithm in place of AES for encryption in these tiny devices.**

Index Terms— **Bluetooth , Encryption, Decryption, PRESENT, AES and Tiger**

I INTRODUCTION

Bluetooth is a wireless communication technology for short range communications. Blue tooth was designed for low power consumption and data transfer in moderate rate over short ranges. The system operates in the 2.4 GHz ISM Band. This frequency band is 2400 – 2483.5 MHz [1].

Bluetooth wireless technology provides peer-to-peer communications over short distances. In order to provide protection and confidentiality, the system provides security measures at the application layer as well as the link layer. Four different entities are used to maintain security at the link layer. They are a Bluetooth device address, two secret keys and a pseudo random number that will be regenerated for every new transaction [1].

There have been tremendous growth on the number of Bluetooth enabled devices in the recent years. With the release of Low Energy support in version 4.0, Bluetooth can now be a part of many devices which were hitherto not thought about. With the devices getting tinier as time progresses, there is a need to see if all those devices can have the processing power to use AES encryption method. This paper makes an attempt to use if any of the light weight cryptographic algorithms can be used in place of AES in the encryption process.

The remainder of this paper is devoted to look at the studies done in the cogitation of E_0 cipher algorithm. Section 2 and 3 gives an overview of Bluetooth security and the entities that are used to maintain security in the link layer. Section 4 details how the encryption algorithm works. Section 5 reviews the recent security studies done on the analysis of E_0 cipher algorithm and the issues with the same. Section 6 talks about the alternate encryption approach to solve the security risks in Bluetooth data transmission

II BLUETOOTH SECURITY OVERVIEW

Bluetooth security is maintained using four different entities. The device address is a 48 bit address. Every Bluetooth device has a unique address. It can be obtained automatically or through a inquiry by another device. The secret keys are derived during initialization and are never disclosed. The encryption key is derived from the authentication key during the authentication process.

For the authentication algorithm, the size of the key used is always 128 bits. For the encryption algorithm, the key size may vary between 1 and 16 octets (8 - 128 bits). The size of the encryption key is configurable for two reasons. The first reason is the different requirements imposed on cryptographic algorithms in different countries both with respect to export regulations and official attitudes towards privacy in general. The second reason is to facilitate a future upgrade path for the security without the need of a costly redesign of the algorithms and encryption hardware; increasing the effective key size is the simplest way to combat increased computing power at the opponent side [1].

The encryption key is entirely different from the authentication key (even though the latter is used when creating the former. Each time encryption is activated, a new encryption key shall be generated. Thus, the lifetime of the encryption key does not necessarily correspond to the lifetime of the authentication key. It is anticipated that the authentication key will be more static in its nature than the encryption key once established, the particular application running on the device decides when, or if, to change it. This is also referred as link key. The RAND is a pseudo-random number which can be derived from a random or pseudo-random process in the device. This is not a static parameter and will change frequently.

III AUTHENTICATION AND ENCRYPTION

Authentication is referred to as the process in which somebody verified that the devices are the one they claim to be. In Bluetooth the authentication process uses a secret number called the PIN and the device addresses [2]. In the context of [3], this can also be called as an agreement protocol.

The authentication process involves the following steps:

1. Generating Initialization Key
2. Generating a Link Key (Authentication Key)
3. Authentication

Authentication uses a challenge response scheme. The verifier sends a message to the claimant that consists of a 128 bit random number. The claimant responds with a message which consists of SRES. The verifier calculates the SRES and if it matches the one received from the claimant, then authentication is successful [2].

Encryption is a process in which the original message is transformed into a different one using an encryption key. The encryption process starts after the authentication process is successfully completed. The encryption key length can vary from 8 to 128 bits. Only the Bluetooth payload is encrypted and not the header and access code [1]. The encryption of the payload is done by a stream cipher called E_0 which will be re-synchronized during every payload. The Figure 1 below shows the overall principle.

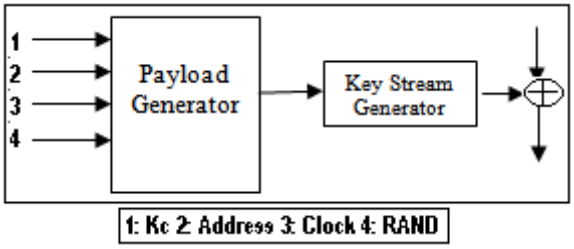

Figure 1. Stream Ciphering for Bluetooth with E_0

The stream cipher E_0 consists of three parts [1]

1. The first part performs initialization (generation of payload key). The payload key generator shall combine the input bits in an appropriate order and shall shift them into the four LFSRs used in the key stream generator
2. The second part generates the key stream bits and shall use a method derived from the summation stream cipher attributable to Massey and Rueppel. This is the main part as it is used for initialization.
3. The third part performs encryption and decryption.

The Massey and Rueppel method has been thoroughly investigated and there exist good estimated of its strength with respect to presently known methods for cryptanalysis. Although the summation generator has weaknesses that can be used in correlation attacks, the high re-synchronization frequency will disrupt such attacks.

Encryption Negotiation: The initiator device sends an encryption mode request message to the peer device. The encryption mode can be either enable encryption or not. If encryption is requested, then the negotiation of the encryption key size is done. The negotiation can go on multiple times till an acceptable key size by both sides is arrived at. The final phase includes sending of a random number by the initiator device in order for both the devices to calculate the encryption key. Encryption is enabled after this key is calculated [2]. This process is vulnerable to Man in the Middle attack if the negotiated key value is weak.

IV ENCRYPTION ALGORITHM

The system uses linear feedback shift registers (LFSRs) whose output is combined by a finite state machine called the summation combiner with 16 states. The output of this state machine is the key stream sequence. The algorithm uses an encryption key K_c, a 48-bit Bluetooth address, the master clock bits and a 128-bit random value. There are four LFSRs of length $L_1 = 25$, $L_2 = 31$, $L_3 = 33$, $L4 = 39$. The total length of the registers is 128. The feedback polynomials are all primitive.

The Hamming weight of all the feedback polynomials is chosen as five – a reasonable trade-off between reducing the number of XOR gates in the hardware implementation and obtaining good statistical properties of the generated sequences [1]. The operational mode of this algorithm is very peculiar. The stream cipher is initialized on every new packet to be encrypted with the following data [2].

1. The encryption key
2. The master device address
3. 26 bits of the master clock

V CRYPTOGRAPHIC ALGORITHM METRICS

Norman and Landgrave [23] deal with the typical characteristics of an Encryption Algorithm that were considered for the development of metrics in their paper. The following are the characteristics as per [23]:

Sl.	Parameter Name	Purpose
1	Type	Whether the encryption is **symmetric or asymmetric**
2	Key Size	The key size that is used in the algorithm
3	Rounds	Number of Rounds
4	Complexity	Complexity for encryption, decryption and key setup
5	Attack	Best known methods of attacks such as brute force and linear cryptanalysis
6	Strength	Assessment of the strength of the algorithm based on key length, complexity and the modes of attack

Table 1. Characteristics of Encryption Algorithm

VI AES-BLAKE ENCRYPTION APPROACH

The AES (Rijndael) method is a block cipher algorithm designed by Joan Daemen and Vincent Rijmen. It can operate over variable length blocks using variable length keys.
The key length can be any of 128, 192 or 256. Most of the attacks on AES have been on the larger key versions like 192 and 258. AES-128 provides a good amount of protection and security

Blake is a hashing algorithm which is an improvement over some of the already existing hash methods and is one of the five finalists for the SHA-3 contest announced by NIST. The compression algorithm on BLAKE is a modified version of a stream cipher called Cha-Cha, whose security and performance has been intensively analyzed [19]. As of December 2010, the best attack on the (reduced) BLAKE hash functions is a pre image attack on 2.5 rounds [20] with complexity 2209 for BLAKE-256 and 2481 for BLAKE-512. A high-complexity distinguisher for 7 middle rounds of the compression function of BLAKE-256 has also been reported.

The third-party ECRYPT benchmarking project compared the five SHA-3 finalists on a large number of computer systems, across a wide range of message sizes. Their results are released into the public domain and are available at the project's website [21]. The ECRYPT benchmarks also serve as an informative comparison between the five SHA-3 finalists: BLAKE, JH, Skein, Keccak, and Grøstl. In general, BLAKE was typically one of the fastest algorithms (probably due to its small number of rounds), comparable with Skein [22].

The plain text to be encrypted is converted into cipher text using AES with a 128 bit key. The session key is encrypted using BLAKE. The combination message is then sent to receiver [27].

The message is divided into two parts one from the BLAKE encryption and the other from AES encryption. The receiver will decrypt the cipher text by their own private key, receive the key that belongs to AES, and then decrypt the cipher text to original [27]

VII . PRESENT ENCYPTION APPROACH

PRESENT is a Block CIPHER which consists of 31 rounds. The block length is 64 bits and two key lengths of 80 and 128 bits are supported. Given the applications we have in mind, we recommend the version with 80-bit keys. This is more than adequate security for the low-security applications typically required in tag-based deployments, but just as importantly, this matches the design goals of hardware-oriented stream ciphers in the eSTREAM project and allows us to make a fairer comparison [23].

Structural Attacks: Integral attacks [25] and bottleneck attacks [26] are mostly suited for AES like ciphers. These attacks are suited for typical bytes bases approach. PRESENT uses an exclusive bitwise feature and hence is not very susceptible to these attacks:

Encryption Process: The plain text to be encrypted is converted into cipher text using PRESENT with a 128 bit key. The session key is encrypted using BLAKE. The combination message is then sent to receiver. Decryption Process: The message is divided into two parts one from the BLAKE encryption and the other from PRESENT encryption. The receiver will decrypt the cipher text by their own private key, receive the key that belongs to PRESENT, and then decrypt the cipher text to original.

The characteristics comparison of AES and PRESENT is given in Table 2

Characteristic	**AES (128)**	**PRESENT (80)**
Type	Symmetric	Symmetric
Key Size (Bits)	128	80
Block Size (Bits)	128	64
Attacks	Published but not computationally feasible	Linear Attack used to analyze up to 26 rounds
Rounds	10	31

Table 2. Comparison of AES and PRESENT

VIII. RESULTS FROM THE TEST ENVIRONMENT

The tests were conducted using the Java Net Beans environment. The AES-Blake approach and the PRESENT-Blake approach were implemented using Java. This piece of code was used to send and receive data of varying sizes like 20, 40 and 60 kilobytes. The results of these tests are compared on the following parameters are:

- Time taken
- Throughput at 100 KHz.
- Gate Equivalents

The summary of the performance of PRESENT-Blake algorithm is given as a table below:

Data Size (kb)	**Time Taken in milliseconds**		
	Whole Process	**Encryption Time**	**Decryption Time**
20	**15000**	**1500**	**1600**
40	**15850**	**2000**	**3250**
60	**16700**	**2500**	**3500**

Table 2. PRESENT-Blake algorithm Time Vs. Data Size

Characteristic	**AES (128)**	**PRESENT (80)**
Throughput at 100KHz	12.4	200
Gate Equivalents	3400	1570
Time (ms)	22600	15000

Table 3. Comparison of AES and PRESENT

The table above (table 3) gives the comparison of AES and PRESENT for various parameters like Throughput, Gate Equivalents and Total time in milliseconds.

PRESENT gives 15 times better throughput when compared to AES because of its light weight nature. This has been validated in the experiment done by us. PRESENT consumes half the gate equivalents when compared to AES. This has been validated in the experiment done by us. The time taken by the light weight algorithm is less than AES as expected.

The results from these tests approximately matches with the comparison of the light weight ciphers done by the team that created PRESENT [23] is given below in Figure 2

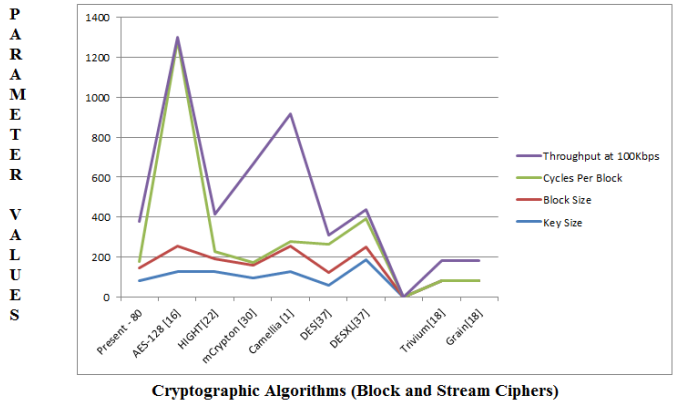

Figure 2. Comparison of Light weight cipher implementations

As interpreted from the table above, PRESENT performs well when compared to the other light weight cipher implementations. However, this needs to be subjected to continuos analysis as new light weight implementation start coming out.

IX. Conclusion

The PRESENT-Blake algorithm performs better than the AES-Blake algorithm based on the tests conducted in the net beans environment. The test results are based on the environment setup for studying the usability of this approach for Bluetooth with varying data sizes. The focus will now shift to doing a very detailed analysis of this new approach and come up with recommendations. The framework should allow both the algorithms and the manufacturer of the device should be the one deciding the encryption approach that will be available in that Bluetooth device based on its capabilities.

(1) References

[1] Bluetooth, S. I. G. (2010). Bluetooth Core Specification v4.0. 30 June 2010 Available online at https:// www. bluetooth.org /Technical /Specifications/adopted.htm

[2] Nikos Mavrogiannopoulos (2005). On Bluetooth. security

[3] Lowe. A hierarchy of authentication speci fications. In PCSFW: Proceedings of The 10th Computer Security Foundations Workshop. IEEE Computer Society Press, 1997.

[4] Y. Lu and S. Vaudenay. Cryptanalysis of the bluetooth keystream generator two-level e0. In Advances in Cryptology - Asiacrypt 2004. Springer-Verlag, 2004. Available from http://www.iris.re.kr/ac04/ data/Asiacrypt2004/11SymmetricKeyCryptanalysis/04 YiLu.pdf.

[5] Y. Lu, W. Meier, and S. Vaudenay. The conditional correlation attack: A practical attack on bluetooth encryption. In Advances in Cryptology- Crypto 2005. Springer-Verlag, 2005. Available from http://www.iacr. org/conferences/crypto2005/p/16.pdf.

[6] O. Levy and A. Wool. A uniform framework for cryptanalysis of the bluetooth e0 cipher. Cryptology ePrint Archive, Report 2005/107, 2005. Available from http://eprint.iacr.org/2005/107.pdf.

[7] D. Bleichenbacher. Personal communication in [8].

[8] M. Jakobsson and S. Wetzel, Security weaknesses in Bluetooth, . in Proc. RSA Security Conf. – Cryptographer's Track, LNCS 2020, pp. 176.191, Springer-Verlag, 2001.

[9] S. R. Fluhrer and S. Lucks, .Analysis of the E0 encryption system, . in Proc. 8th Workshop on Selected Areas in Cryptography, LNCS 2259, Springer-Verlag, 2001.

[10] F. Armknecht, .A linearization attack on the Bluetooth key stream generator.. Cryptology ePrint Archive, report 2002/191, available from http://eprint.iacr.org/2002/191/ 2002.

[11] M. Krause, .BDD-based cryptanalysis of keystream generators, . in Advances in Cryptology – EUROCRYPT'02, LNCS 1462 (L. Knudsen, ed.), pp. 222.237, Springer-Verlag, 2002.

[12] M. Saarinen, Re: Bluetooth und E0. Posting to sci.crypt.research, September 2000.

[13] P. Ekdahl and T. Johansson, Some results on correlation in Bluetooth stream cipher, in Proceedings of the 10th Joint Conference on Communication and Coding, Obertauern, Austria, March 2000.

[14] M. Hermelin and K. Nyberg, Correlation properties of the Bluetooth combiner generator, in Information Security and Cryptology, LNCS 1787, pp. 17-29, Springer-Verlag, 1999.

[15] J. Goli´c, V. Bagini, and G. Morgani, .Linear cryptanalysis of Bluetooth stream cipher,. in *Proceedings of Eurocrypt 2002*, Springer, 2002.

[16] Y. Lu and S. Vaudenay, Faster correlation attack on Bluetooth keystream generator E0, in Advances in Cryptology – CRYPTO'04, LNCS 3152, pp. 407-425, Springer-Verlag, 2004.

[17] Y. Shaked and A. Wool, .Cracking the Bluetooth PIN,. In *Proc. 3rd USENIX/ACM Conf. Mobile Systems, Applications, and Services (MobiSys)*, (Seattle, WA), June 2005.

[18] Patheja, Akhilesh and Sudir, A Hybrid Encryption Technique to Secure Bluetooth Communication in Proceedings by International journal of Computer Applications, International Conference on Computer Communication and Networks CSI- COMNET-2011

[19] J.-P. Aumasson, L. Henzen, W. Meier, and R. Phan, "SHA-3 proposal BLAKE," December 2010.

[20] Li Ji and Xu Liangyu. Attacks on round-reduced BLAKE.

ePrint report 2009/238, 2009.

[21] D. J. Bernstein and T. Lange, "List of SHA-3 candidates measured, indexed by machine," 2011, http://bench.cr.yp.to/results-sha3.html.

[22] Ryan Toukatly, "SHA-3: The BLAKE Hash Function", Rochester Institute of Technology.

[23] A. Bogdanov1, L.R. Knudsen etal "PRESENT: An Ultra-Lightweight Block Cipher", 2007

[24] Norman D. Jorstad and Landgrave "Cryptographic Algorithm Metrics", 1997

[25] L.R. Knudsen and D.Wagner. Integral Cryptanalysis. In J. Daemen and V. Rijmen, editors, Proceedings of FSE 2002, LNCS, volume 2365, pages 112–127, Springer-Verlag, 2002

[26] H. Gilbert and M. Minier. A Collision Attack on 7 Rounds of Rijndael. In Proceedings of Third Advanced Encryption Standard Conference, National Institute of Standards and Technology, 230–241, 2000.

[27] Sandhya S, Sumithra Devi K A,Performance Evaluation of Crypt Analytical Approaches in Bluetooth Networks, International Journal of Application or Innovation in Engineering & Management (IJAIEM),Volume 2, Issue 7, July 2013,ISSN 2319 - 4847

