International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com
Volume 2, Issue 7, July 2013 ISSN 2319 - 4847

Performance Evaluation of Crypt Analytical Approaches in Bluetooth Networks

Mrs. Sandhya S[1], Dr. Sumithra Devi K A[2]

[1]Asst Prof, MCA Department, RVCE, Bangalore

[2]Prof. & Director, MCA Department, RVCE, Bangalore

ABSTRACT

The evolution of Bluetooth technology has made wireless communication easier. In traditional Bluetooth communication, 128-bit of symmetric stream based encryption is used. A new approach to encryption using AES-Blake algorithm is being suggested. In this paper, AES-Blake Algorithm and the hybrid encryption method (Triple DES-Tiger) are evaluated based on multiple factors such as total time taken for the communication process, encryption and decryption time. From the results of the tests, it is observed that the suggested AES-Blake algorithm works better than the Triple DES-Tiger algorithm.

Keywords: Encryption, Decryption, Blake, AES and Tiger, Bluetooth

1. 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 with a frequency band of 2400 – 2483.5 MHz. The technology allows the formation of adhoc networks called piconets between two or more wireless devices, which when connected can communicate on the same physical channel with a common clock and hopping sequence. A number of independent piconets may exist in close proximity, enabling each piconet to have a different master device and an independent timing and hopping sequence. A Bluetooth device may participate in two or more piconets at the same time that is achieved using a process called time-division multiplexing, but., a blue tooth device cannot be a master of more than one piconet, though it can be a slave in many independent piconets [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 using four different entities to maintain security at the link layer which are : a Bluetooth device address, two secret keys and a pseudo random number that will be regenerated for every new transaction [1].

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 existing attacks on E_0 cipher algorithm and security studies done on the analysis of E_0 cipher algorithm and the issues with the same. Section 6 talks about the TRILE DES-TIGER approach. Section 7 talks about the AES-BLAKE approach suggested. Section 8 and 9 discusses about the test environment and results.

2. BLUETOOTH SECURITY OVERVIEW

Bluetooth devices have 48 bit 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 and 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, namely:

i. Different requirements imposed on cryptographic algorithms in different countries both with respect to export regulations and official attitudes towards privacy in general.
ii. 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.

3. AUTHENTICATION AND ENCRYPTION

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:

i. Generating Initialization Key
ii. Generating a Link Key (Authentication Key)
iii. Authentication

Authentication uses a challenge response scheme. The verifier sends a message to the claimant that consists of a 128 bit random number, to which the claimant responds with a message which consists of SRES(Signed Response). 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 and 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 stream cipher E_0 consists of three parts [1]:

a. 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
b. 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.
c. 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.

4. 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 to generate a key stream sequence. The algorithm uses an encryption key Kc, a 48-bit Bluetooth address, the master clock bits and a 128-bit random value. There are four LFSRs of length L1 = 25, L2 = 31, L3 = 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].

i. The encryption key
ii. The master device address
iii. 26 bits of the master clock

5. ATTACKS ON E_0 CIPHER

Several crypt-analytical results regarding E_0 ([7], [8], [9] and [10]) have appeared over past years. There are two types of attacks “short key” and “long key” attacks as described by Levy and Wool [6]. Short key attacks are those that need at most 240 known key stream bits and are applicable in the Bluetooth setting. Long key attacks are applicable only if E_0 cipher is used outside the Bluetooth mode of operation. There have been many alternate encryption approaches suggested to overcome the issue with the CIPHER based encryption algorithm in bluetooth. The paper compares the TRIPLE DES-TIGER approach by Patheja, Akhilesh and Sudir [18] and the AES-BLAKE approach suggested in this paper.

6. TRIPLE DES-TIGER ENCRYPTION APPROACH

This approach for encrypting Bluetooth communication was suggested by Patheja, Akhilesh and Sudir [18]. In this approach, during the process of sending encrypted information, the random number generator uses 64-bit triple DES session key only once, it encrypts the plaintext to produce cipher text. On the other hand, the sender get debit's public key from public key management center, and then using Tiger to encrypt session key. Finally, the combination of the session key from Tiger encryption and the cipher text from triple DES encryption are sent out [18].

The decryption of hybrid encryption algorithm is as follows. The first, the receiver divide received cipher text into two parts, one is cipher text from the Tiger algorithm encryption, the other is cipher text from the triple DES algorithm encryption. The second, the receiver decrypt cipher text by their own private key, receive the key which belongs triple DES algorithm, then decrypt the cipher text to the original text by key [18].

7. AES-BLAKE ENCRYPTION: PROPOSED 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].

ENCRYPTION PROCESS

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. The process is given in Figure 1.

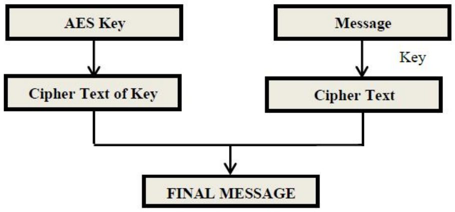

Figure 1. Encryption Process of AES–BLAKE Algorithm

DECRYPTION PROCESS

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. The decryption process is given in Figure 2.

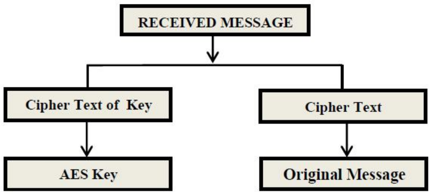

Figure 2. Decryption Process of AES–BLAKE Algorithm

8. Test Environment

The tests were conducted using the .NET environment and the 32feet.net extension for .NET. The code was written in VB.Net. A Bluetooth dongle connected to the windows 7 system and three mobile phones made up the test environment.

The setup is shown in figure 3. The dongle initiates the discovery process and then other discoverable Bluetooth devices in the range are identified and listed.

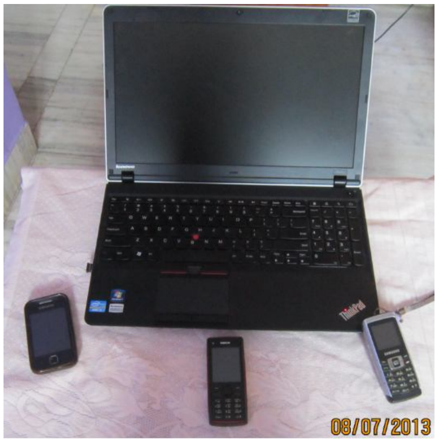

Figure 3. Test Environment Setup

The list of devices which are discoverable were listed in the list box. If the “Interact” option is used, then you can find out the device address, the device class and whether it has already been authenticated or not. You can initiate pairing to the device selected by using the “Pair” option. The screen shot is shown in Figure 4.

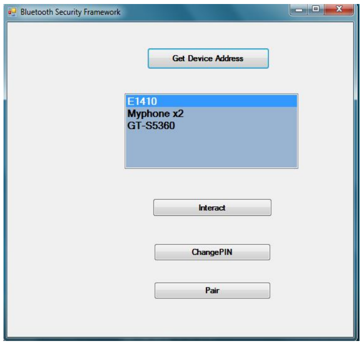

Figure 4. Test Environment screen shot

9. Results Obtained

The tests were conducted using the .Net environment and 32feet addon. The Triple DES-Tiger approach and the AES-Blake approach were implemented using the .Net environment. 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:

a. Time taken for the Whole process

b. Time taken for Encryption

c. Time taken for Decryption

The summary of the performance of AES-Blake algorithm is given as a table below.

Data Size (kb)	Time Taken in milliseconds		
	Whole Process	Encryption Time	Decryption Time
20	22600	1500	2100
40	2300	3000	4500
60	22000	4200	4600

Table 1. AES-Blake algorithm performance summary

The following graph (Figure 6) depicts the time taken for the whole process in micro seconds with respect to varying data packet sizes in kb:

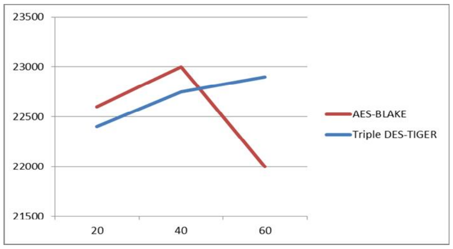

Figure 5. Time Taken Vs Data Size

From the above figure, Triple DES-Tiger performs better at low data sizes like 20 and 40 kb. AES-BLAKE approach works better for a data size of 60 kb. There is a huge drop in the total time taken in case of 60 kb when compared to the time taken in case 40 kb. So, the test was rerun for 60 kb to revalidate the overall time taken.

The following graph (figure 7) depicts the time taken for the encryption in with respect to varying data packet sizes in kb:

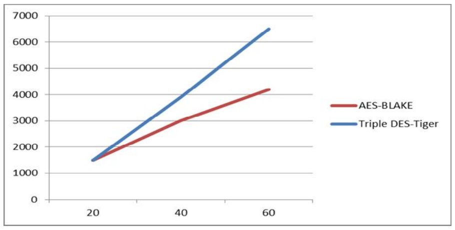

Figure 6. Time Taken for Encryption Vs Data Size

From the above figure, it can be interpreted that AES-BLAKE approach works better from the data size of 40 kb. Moreover, the time between both the algorithms seems to be increasing as the data size increases. AES-BLAKE seems to be performing much better than the other approach as the data size increases.

The following graph (figure 8) depicts the time taken for the decryption in with respect to varying data packet sizes in kb:

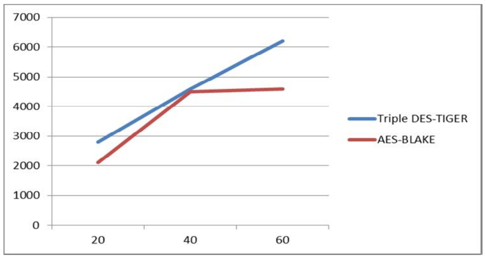

Figure 7. Time Taken for Decryption Vs Data Size

From the above figure, it can be interpreted that AES-BLAKE approach works better for 20 kb and 60 kb. For 40 kb, the decryption times are the same. It is important to note that the time taken by AES-BLAKE for 60 kb seems to be almost same as the value got for 40 kb. So, the test was rerun for 60 kb to revalidate the overall time taken for decryption.

10. CONCLUSION

For the given data set, the AES-Blake algorithm performs slightly better in terms of time efficiency than the Triple DES-Tiger algorithm based on the tests conducted. The algorithms can also be applied for a huge data set in future and can be analyzed in a controlled environment. The algorithms should also be applied to different data patterns to realize the impact of each of these algorithms.

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,
[22.] indexed by machine," 2011, http://bench.cr.yp.to/results-sha3.html.
[23.] Ryan Toukatly, "SHA-3: The BLAKE Hash Function", Rochester Institute of Technology.