Key Management Policy in Wireless Sensor Security

Siddhartha Chatterjee[1], Shirsha Mullick[1]

[1]NSHM Knowledge Campus, Durgapur

Abstract: Bluetooth is a wireless technology used for exchanging data over short distances and personal area networks. The complication in mass adoption of Bluetooth is because of its Security concern, secure simple pairing (SSP) is an important security mechanism for devices. Authentication is one of the main features as well as encryption for Bluetooth communication. Bluetooth uses different types of levels to be defined for devices and the security of Bluetooth. An encryption algorithm which is based on AES is suggested to increase the security of data in Bluetooth communication. A device gets connected using its address and authentication random number so that we can encrypt and decrypt data on the connection. AES is known for its higher efficiency. We use symmetric key cryptography for encryption and decryption of the data.

Keywords: Real time, surveillance, pedestrian detection, Convolution Neural Networks.

1. Introduction

1.1. Bluetooth Security

Bluetooth uses different types of levels for the device. There are too many vulnerabilities in modern wireless technology and to find out those vulnerabilities like Man-in-the-Middle Attack or unauthorized data access, some encryption processes are used in Bluetooth security. Levels of security are:-

- Authorization needed: It is a level where authorization takes place between two unique devices.
- Authentication needed: The unique devices must authenticate with each other before connecting to the application.
- Encryption needed: The process must be secured by using an encryption process.

1.2. Modes of Security

We have four different modes of Bluetooth security and they are –

- Mode 1: It is non-secure, so it bypasses the authentication and encryption process with various devices which connect. It is for a larger data rate from the normal one.
- Mode 2: There is a centralized security manager which mainly manipulates the access to certain devices and certain services.
- Mode 3: In this mode, both the authentication and encryption are done from the devices themselves.
- Mode 4: The physical link is confirmed before the security process and secure pairing begins with the key exchange and link key generation.

1.3. Link-level Security

Below is the link-level security of Bluetooth. In the figure, the claimant and verifier try to connect.

Four parts are described which are involved in the process in Figure 1.

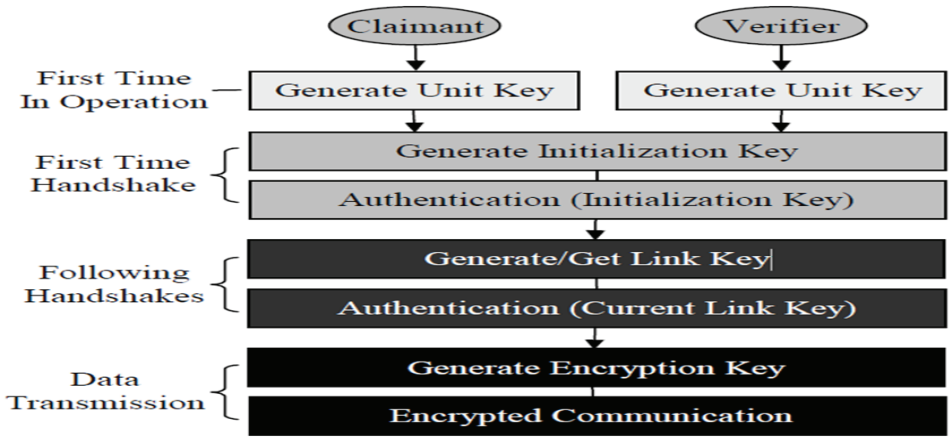

Figure 1. Link-level Security

1.4. Previous work

In Bluetooth, a challenge-response scheme for the authentication is used. There are two units; unit A and unit B. Unit A acts as the verifier and unit B as the claimant. Unit A and unit B share a secret key K which will be the current link key. Unit A authenticates unit B by sending authentication random number A and Bluetooth device address A. After the authentication is done by unit A, unit B starts the authentication process with unit A by sending its authentication random number B and Bluetooth device address B along with a link key. The SRES and SRES' are derived from the new authentication random number B. The value of ACO is retained when authentication achieves success.

2. Proposed Algorithm & General Discussion

2.1. Key Management Scheme

Generating, Storing and Distribution of keys are done using key management scheme. The security scheme is of symmetric key (private key) which is also known as a link key, shared by both the claimant and verifier. Link key can be of four types based on the purpose of authentication lifetimes. Temporary key, only used for a current session, Semi-permanent key, used after the session is over or terminated. For the initialization process Initialization key is used and at last Encryption key can be used for the encryption, to ensure security. Temporary key generates a master key which is used for broadcast message and semi-permanent keys generate both unit and combination key which is used for authentication and initialization hence completing the scheme.

Bluetooth uses custom algorithm e21 and e22 and authentication as MAC called (E1)

2.1.1. E1 ALGORITHM

E1 is used to perform authentication. Inputs :

1. A random word "Authentication Random Number A
2. Link key
3. Bluetooth device address (48 bits long).

2.1.2. ALGORITHM

Step 1:- E1 contains A+ and AF+

Step 2:- Link key is supplied first for the input of A+

Step 3:- Transformation is being done by adding and XOR-ing its bytes with some constants.

Step 4:- Link key is supplied

Step 5:- For the Bluetooth device address.

2.2 Key Management

The link key handles all security transactions between the parties between two or more than that. After completing the current session for authenticating Bluetooth units that have shared it, a semi-permanent key is used to carry out this process. On the other hand, temporary key is used to connect the point to multipoint connections and cannot be reused. It waits only before the current session is terminated.

When the device is installed, the unit key is generated. The master key is a temporary key and can replace the current link key, which is used to transmit information to more than one recipient when it is needed by the master unit.

A fixed code can be implemented in an application for the willing device to connect with. The main point is that the PIN must be the same and to be entered on both the devices during the initialization process.

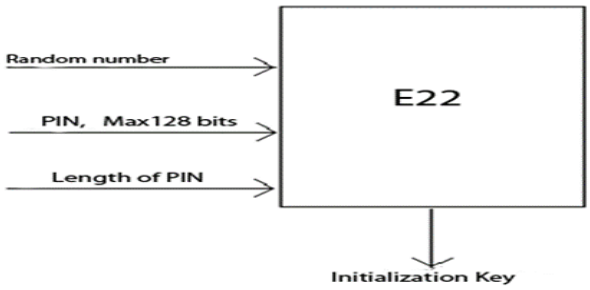

Figure 2: Master and initialization keys using E22

The PIN is entered into both devices during the initialization process. To communicate between two devices an initialization key is needed, which is generated by the E22 algorithm. For the very first time when Bluetooth is installed in a device, a unit key is generated with the help of the E21 generating algorithm. After the generation of a unit key, the key is stored in a volatile memory and is rarely changed.

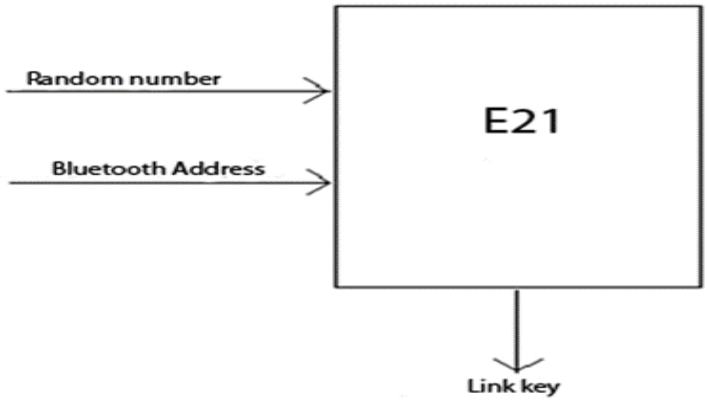

Figure 3: Link Key using E21

If the devices installed have decided to use one link key as an operation key, then, the combination key is generated during the initialization process and is created at the same time interval. First of all, they generate a random number, using the E21 generating algorithm, which generates a key, which connects the Bluetooth devices address with the random number created. Again using the E22 generating algorithm a master key is generated which is the only temporary key of that of the link key. After that, the transmission of the third random number is done to the slave key with the help of a generating algorithm by which the link key is executed by both the master and the slave units.

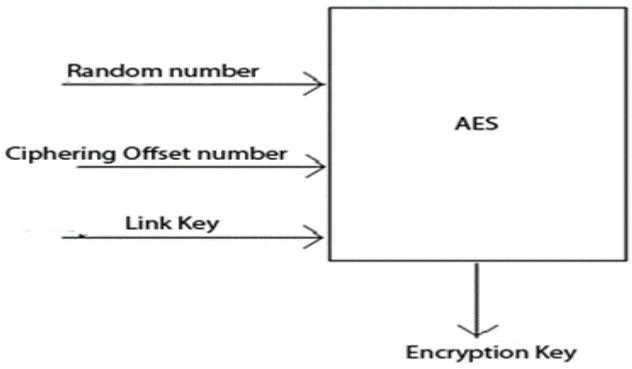

Figure 4: Key generating algorithm AES for the encryption key

An encryption key is then generated from the newly current link key. During the authentication process, a Ciphering Offset Number (COF) is generated based on Authenticated Ciphering Offset (ACO). The encryption key is automatically changed when the Bluetooth device enters the encryption mode. The

generation of encryption key is done when the link manager (LM) activates the encryption process during setting up the device.

2.3. Bluetooth Encryption Process

At first, a 4-digit pin is used using E21 which generates a link key which helps in the authentication process and is completed by the AES algorithm for the encryption. After the connection is established the encrypted data can be sent which is secure because of the encryption using AES.

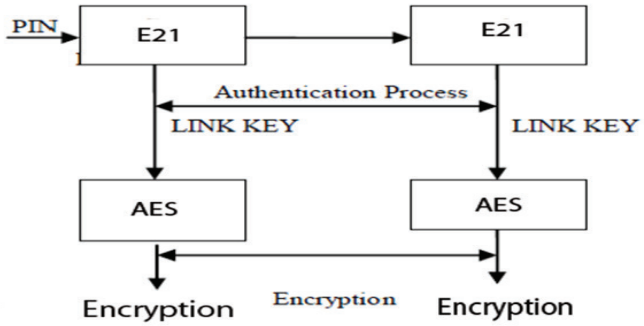

Figure 5. Bluetooth Encryption

Steps involved in Bluetooth encryption process are:

1. E21 create a link key using a 4-digit pin and after that encryption key is created using AES algorithm.
2. Plaintext is converted to cipher text using encryption.
3. Both devices need to match the link key for the authentication process using the E1 algorithm.

2.3.1. AES ALGORITHM

AES is a symmetric key encryption algorithm, it stands for Advanced Encryption Standard and is the most widely used encryption algorithm in the world. For the encryption, we give a message or data with a key. It then scrambles the outputs and converts the original message or data into an unreadable or unrecognizable format.

For the decryption, the same process is used i.e. providing the same key with unreadable message or data, which is then converted into readable format after the decryption. It is made up of a couple of initialization steps, encryption is performed using the expanded key in a series of rounds.

3. Implementation

Implementation of the proposed algorithm has been done using a GUI Based on the device connection as on the computer. JAVA language has been used for the program and Java Swing for GUI. NetBeans IDE platform has been used for the development of the program and encryption/decryption is done using AES Algorithm.

Assumption made while Coding:

- Device 1 acts as the sender and Device 2 as the receiver.
- Device 1 initiates the connection to Device 2.
- Both Device 1 and Device 2 are in the same place as they both know the password used in Encryption/ Decryption.

GUI (Graphical User Interface):

4. Test Result and Analysis

While running the paper from NetBeans IDE, a GUI is created and represented from where the device can connect and encrypt files. At first, both the devices create a connection by giving Device 1 a name and a link key and after pressing the connect button, both the devices get connected. The Connect button takes input from both the Device and Link Key label and generates a random authentication number. After a Successful connection, we select a file from a file browser. Now we use the Encrypt and Send button to encrypt and send the data to the other device. For Encryption, we give a password so that it can be decrypted by this key. After the successful encryption, we get a file with .aes extension which is not accessible, means it is fully secure and we can access it only after the decryption by giving the correct secret key. For decrypting the file, we again have to go through the file browser and select the encrypted file. Now we will use Decrypt button to decrypt the file by providing the secret password or key used during the Encryption. After giving the right secret key, we achieve the decrypted file. Hence, the two devices can connect and can send or receive the file with full security. Full security is achieved by encrypting the file using AES algorithm.

5. Conclusion

It is very well known to us that the Bluetooth device replaces data and information from one to another using wireless signals. Newer versions support greater range and safer security. Giving it better security coincides with better attacks too. During transmission, the chances of attacks can be increased which is

secure by an AES algorithm. This algorithm is highly secure and is less attacked. In future, the attacks may increase though the transmission of data is more replicable. Providing better security needs better algorithm techniques which leads to better technological immigration around the globe.

References

1. Juha T. Vainio, "Bluetooth Security", 2000-05-25.
2. Amoroso E., "Fundamentals of Computer Security Technology", Prentice Hall, 1994.
3. John Padgette, John Bahr, Mayank Batra, Marcel Holtmann, Rhonda Smithbey, Lily Chen, Karen Scarfone, "Guide to Bluetooth Security", NIST Special Publication, 2017.
4. Andreas Becker, "Bluetooth Security & Hacks", 2007.
5. Trishna Panse, Vivek Kapoor, "A Review on Security Mechanism of Bluetooth
6. Communication", International Journal of Computer Science and Information Technologies, 2012.
7. Xiaojiang Du, Yang Xiao, Mohsen Guizani, Hsiao-Hwa Chen, "An effective key management scheme for heterogeneous sensor networks", Ad Hoc Networks, 2007.
8. Renu Dalal,Yudhvir Singh, Manju Khari1, "A Review on Key Management Schemes in MANET", International Journal of Distributed and Parallel Systems, 2012.
9. Zhen Yu, Yong Guan, "A Key Management Scheme Using Deployment Knowledge for Wireless Sensor Networks", IEEE, 2008.
10. Chu-Hsing Lin, "Dynamic key management schemes for access control in a hierarchy", Computer Communications, 1997.
11. Yan Sun, W. Trappe, K.J. Ray Liu, "A scalable multicast key management scheme for heterogeneous wireless networks", IEEE, 2004.
12. Mohammad Sadegh Yousefpoor, Hamid Barati, "Dynamic key management algorithms in wireless sensor networks: A survey", Computer Communications, 2019.
13. Wuling Ren, Zhiqian Miao, "A Hybrid Encryption Algorithm Based on DES and RSA in Bluetooth Communication", IEEE, 2010.
14. MA Albahar, O Olawumi, K Haataja, P Toivanen, "Novel hybrid encryption algorithm based on AES, RSA, and Twofish for Bluetooth encryption", erepo.uef.fi, 2018.
15. K Rege, N Goenka, P Bhutada, S Mane, "Bluetooth Communication using Hybrid Encryption Algorithm based on AES and RSA", International Journal of Computer Applications, 2013.
16. Lin Zou, Ming Ni, Yiting Huang, Wenfeng Shi, Xiaoxia Li, "Hybrid Encryption Algorithm Based on AES and RSA in File Encryption", Frontier Computing, 2020.
17. Khet Khet Khaing Oo, Yan Naung Soe, "Encryption Data Measurement and Data Security of Hybrid AES and RSA Algorithm", International Journal of Trend in Scientific Research and Development, 2019.

18. M Harini, K Pushpa Gowri, C Pavithra, M Pradhiba Selvarani, "A novel security mechanism using hybrid cryptography algorithms", IEEE, 2017.

19. Pekka Aho, Nadja Menz, Tomi Räty, Ina Schieferdecker, "Automated Java GUI Modeling for Model-Based Testing Purposes", IEEE, 2011.

20. Yanhong Sun, Edward L. Jones, "Specification-driven automated testing of GUI-based Java programs", ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference, 2004.

21. Johannes Weber, Andreas Rehkopf, "A Java-based remote GUI concept for distributed automation systems", IEEE, 2009.