

A New Secure Scheme Purposed For Recognition And Authentication Protocol In Bluetooth Environment

Seyed Hassan Sadeghzadeh*, Seyed Javad Mirabedini Shirazani**, Mohammad Mosleh*

**Faculty of Higher Education, Islamic Azad University, Dezful Branch, Dezful, Iran*

***Islamic Azad University Central Tehran Branch, Iran*

Sadeghzadeh1@gmail.com, jvd2205@yahoo.com, Mosleh@iaud.ac.ir

Abstract— Since its creation, Bluetooth has transformed itself from a cable replacement technology to a wireless technology that connects people and machines. Bluetooth has been widely adapted on mobile phones and PDAs. Many other vendors in other industries are integrating Bluetooth into their products. Although vendors are adapting to the technology, Bluetooth hasn't been a big hit among users. Security remains a major concern. Poor implementation of the Bluetooth architecture on mobile devices has led to some high profile Bluetooth hacks. Weak security protocol designs expose the Bluetooth system to some devastating protocol attacks [1].

Because some of these devices often contain sensitive personal or corporate information, security is always an issue when using Bluetooth with these devices. One issue with wireless technology is that signals might leak to outside of desired boundaries. These leaked signals can be used for various attacks ranging from eavesdropping to injecting traffic to fully impersonating another device. Over the years, several vulnerabilities have been discovered in the Bluetooth protocol design as well as the implementation of Bluetooth technology in applications. In this paper I will begin with briefly describing the Bluetooth protocol, explaining security measures in Bluetooth, and go into detail on specific analyze and discuss the vulnerabilities, and finally present an improvement pairing protocol that works with Encrypted Key Exchange Protocol.[24]

Keywords—Bluetooth, Mac, station-to-station, SPEKE

I. INTRODUCTION

Bluetooth data transmit on the unlicensed 2.4GHz ISM band. ISM band stands for industrial, scientific, and medical radio bands. It defines and reserves radio frequency for industrial, scientific, and medical purposes [2]. Bluetooth uses a frequency-hopping scheme in order to minimize the interferences with other technologies and applications such as 802.11, microwave ovens, cordless phones, etc. The connection range of off-the-shelf Bluetooth devices vary from 10 meters to 100 meters. Each Bluetooth device has a globally unique 48bit MAC address. The first 24 bits of the

Bluetooth address is vendor specific. For example 0:14:9A:C9:20:10 .

II. BLUETOOTH SECURITY

The primary emphasis of the Bluetooth security architecture involves link and network service protection. To do this, Bluetooth security mechanisms enforce authentication based on shared secret exchange, separable authorization for trusted versus untrusted connecting devices, and communication encryption. In this chapter the procedures that allow devices to exchange link keys, authenticate, establish secure sessions, and gain authorization to access network services are briefly summarized. A short description of the Bluetooth security algorithms is given at the end.

A. Security Modes

Three security modes are described in the Bluetooth specification. They are defined as follows:

- Mode 1: No security. A device will not initiate any security at all.
- Mode 2: Service-level security. A channel on L2CAP level is first established without any security procedures. Depending on which applications are running, different security requirements can be set for each one of the applications.
- Mode 3: Link-level security. Before establishing a channel on L2CAP the device initiates security procedures for a secure connection. This is the built-in security mechanism and it is not aware of service- or application-layer security.
- For security mode 2, settings per device basis and per service basis are made. Two databases are used, one to hold device information and one to hold service information. The service database contains the security configuration as provided by the application software. The device database holds information regarding past sessions with other devices [3]

- Two trust levels are specified: trusted and untrusted. Trusted devices are paired and that are marked as trusted in the device database. This level has unrestricted 15 accesses to all services. Untrusted devices are unknown or are paired but not marked in the database as trusted. This level has restricted access to services. Untrusted devices must be authorized to access a service if authorization is required (see below). A new device is always regarded as untrusted [4]. Three levels of service security are defined so that the requirements for authorization, authentication, and encryption can be set independently. For example, services can be set to require authorization and authentication, require authentication only, and set to be open to all devices. The three attributes that determine the level of security are as follows [5].
- Authorization required: Access to the application is granted automatically to trusted devices and only after an authorization procedure granted to untrusted devices. Authorization always includes authentication.
- Authentication required: The remote device must be authenticated before being allowed to connect to the local service.
- Encryption required: Access to the service is possible only after the link has entered encrypted mode.

B. Key management

The Bluetooth security architecture is based on symmetric key cryptography where two Bluetooth devices share a common link key for authentication and encryption. Figure 1 shows the Bluetooth key structure and his description in table1

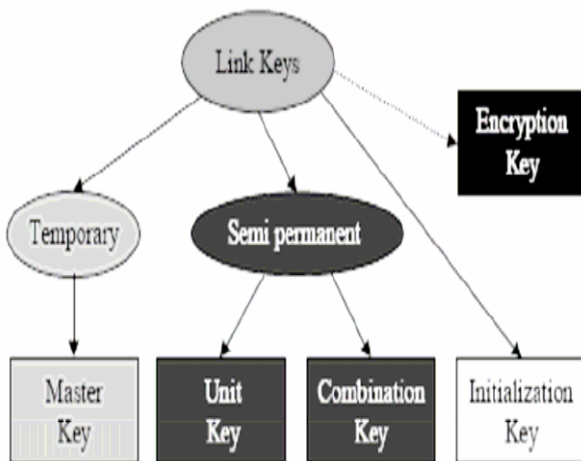


Figure1. Bluetooth Key Structure

Table 1. Bluetooth Keys

Key name	descriptions
Initialization Key	The initialization key (Kinit) is the first key being generated in the pairing process. It is used to derive combination or unit keys later on in the pairing process. Once a combination or a unit key is derived, the initialization key will be discarded. Note that the strength of this key relies solely on a 4 to 16bytes PIN
Combination or unit keys	Combination keys (Kab) and unit keys (Ka) are semi-permanent in the sense that devices store them permanently unless they are being updated through the link key update procedures or the broadcast encryption scheme. These keys can be reused in multiple sessions by the devices that share them.
Master key	The Bluetooth specification defines shared master keys to allow piconet masters to encrypt broadcast traffic.
Encryption Key	Encryption keys (Kc) are derived from the current link keys and are automatically updated each time the devices enter encryption mode. KC is used to generate a cipher stream KCipher that in turn will be XORed with payloads.

Four hash functions are used in pairing, authentication, and encryption. The heart of all four functions is the SAFER+ block cipher. Table2. shows the hash functions.

Table 2. Bluetooth security function

function name	descriptions
E22	The Bluetooth design uses E22 to generate initialization keys (Kinit)
E21	Bluetooth design uses E21 to generate unit keys
E1	The Bluetooth design uses E1 to generate authentication responses
E3	to generate the ciphering key Kc ,which is used by system E0 to generate key streams for encrypting message payloads in encryption

C. Pairing

Before two Bluetooth devices can establish a connection and send data to each other, they have to go through a pairing procedure, which is essentially a process for creating a common key for authentication and encryption between two Bluetooth devices. Figure 2 [14] show a simplified pairing .

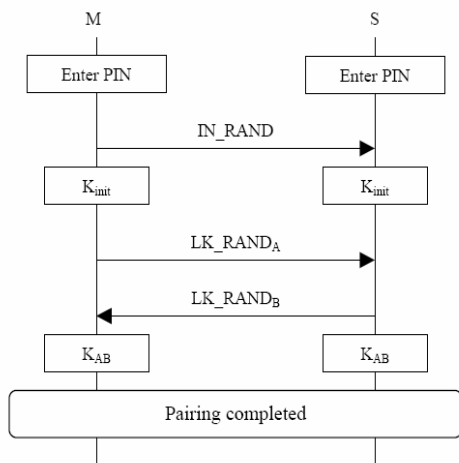


Figure2. A simplified Bluetooth pairing protocol

D. Authentication

Bluetooth security architecture uses a challenge-response authentication scheme. Figure 3 show a simplified authentication.[14]

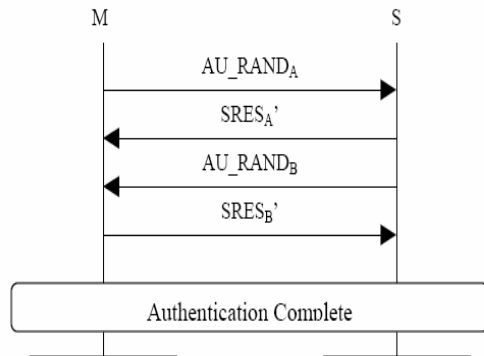


Figure3. A simplified Bluetooth authentication protocol

III. ATTACKS ON BLUETOOTH PROTOCOL

By studying the Bluetooth protocol specification, several security issues were discovered.

In addition, by experimenting with Bluetooth devices, a number of weaknesses were found in the usage and implementation of Bluetooth technology. This paper focuses on abuses in the pairing and authentication procedures due to poor protocol design or due to flaws or pitfalls in the implementation of Bluetooth applications and devices. An overview of the abuses is given in a vulnerability table (table 3) followed by more detailed descriptions of the individual security issues in conjunction with proposed solutions. Attacks on the encryption cipher are out of the scope of this paper.

Table 3. detailed descriptions of the individual security issues in conjunction with proposed solutions[13]

Attack name	Attack Description	Flaw
Unit Key Attack	Unit key devices are vulnerable to impersonation and eavesdropping. Same key is used for all devices, which makes it possible for any device to masquerade as another device.	Bluetooth pairing and Authentication protocol
Offline PIN Cracking	A third party can capture messages being sent during the pairing and authentication procedures between two devices, and then brute force the PIN offline, as well as calculate the link key.	Bluetooth pairing and Authentication protocol
Message Relay Attack	A third party can relay authentication messages sent between two devices and get authenticated on both sides without knowing the shared secret. Devices are not mutually authenticated. Man-in-the-middle does not know the shared secret, and can eavesdrop on unencrypted communication.	Bluetooth pairing and Authentication protocol
Device Impersonation	A third party can impersonate another device, and get authenticated without knowing the shared secret by relaying the authentication request to the real trusted device and relaying back the real response.	Bluetooth pairing and Authentication protocol
Denial-of-Service Attack on Authentication	A third party can cause DOS on the authentication procedure by repeatedly requesting authentication on a faulty link key using the identity of a previously paired device, which will lead to the previously paired device not being able to get authenticated. The repeated attempt protection scheme [6, vol-3, p.799] is abused to perform the DOS.	Bluetooth pairing and Authentication protocol
DoS Attack on Pairing and Authentication	A third party can cause DOS on the pairing and authentication procedure by repeatedly sending bogus pairing messages forcing heavy calculations on the receiving device.	Bluetooth pairing and Authentication protocol
Online PIN Cracking	A third party can brute force the PIN of fixed devices by repeatedly connecting and trying a PIN. To bypass the repeated attempt protection [6, vol-3, p.799] the BD ADDR of the third-party device is changed for every attempt.	Bluetooth pairing and Authentication protocol
Online-Offline PIN Cracking	A third party can first initiate pairing with a fixed PIN device and then brute force the PIN offline. The pairing and authentication process is abused, and information about the PIN is leaked from the fixed PIN device.	Bluetooth pairing and Authentication protocol

IV. ENHANCEMENTS

In this section, an enhancement to the Bluetooth security protocol will be proposed in an attempt to defence against the attacks described in the previous section. All proposed enhancements are on the protocol level. In other words, all lowerlevel cryptographic features and functions, such as the SAFER+ block cipher and various hash functions, will not be discussed.

V. IMPROVED SCHEME

The goal of the password-based EKE protocol is to exchange a common key between two parties over an insecure channel, such as a wireless interface, using a shared weak PIN number, such as a 4 digit PIN. That is exactly what the pairing process in the Bluetooth design tries to accomplish. The design of SPEKE incorporates the usage of both symmetric and asymmetric systems. Figure 4 shows the modified pairing using password-based EKE.[13]

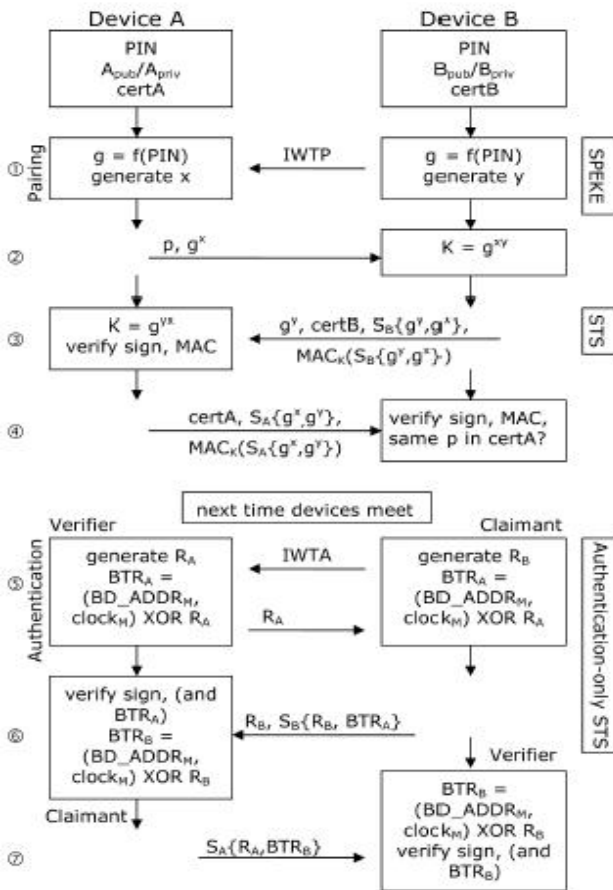


Figure4. the modified pairing and Authentication protocol

Steps 1 through 4 are equivalent to the pairing procedure in current Bluetooth. During these steps the authenticated certificate exchange takes place. Steps 1 and 2 are based on SPEKE. In step 1, device B initiates the pairing procedure by

sending an IWTP packet to device A. Both devices then calculate g using the function f with PIN as the input. Since the same PIN is entered on both sides, the same g is calculated. A then generates a random number x while device B generates a random number y . Then in step 2, A sends p and g^x to B, which calculates the strong key K . The key K is used only to create MACs. Steps 3 and 4 are based on STS. In step 3, B sends g^y , its certificate (certB), a hash of the concatenation of g^y and g^x , signed with B's private key, and a MAC of the signed hash using the key K . A calculates the key K , and uses B's public key found in B's certificate to verify the signature of the hash of the concatenation of g^y and g^x . If this is valid, B received A's g^x value successfully, and has the private key associated with the public key included in the provided certificate. A then verifies the MAC using the calculated key K to verify that B knows the key K . This is followed by step 4, where A sends its certificate, a hash of the concatenation of g^x and g^y , signed with A's private key, and a MAC of the signed hash using the key K . B verifies that the signature is correct using A's public key included in the provided certificate, and that the MAC is correct using K . Last, B verifies that the provided p in step 2 is the same as the p included in A's certificate. An attacker might replace the p in step 2 with a small value in order to launch a subgroup confinement attack or a discrete log attack [12]. Therefore, p must be included in the certificate, and this last check is very necessary.[13]

Figure 4 shows the modified authentication protocol using only STS. As you see, Steps 5 through 7 are equivalent to the authentication procedure in current Bluetooth.

The authentication procedure is based on Authentication-Only STS, a mutual authentication challenge-response protocol [25]. B initiates the authentication procedure in step 5 by sending an IWTA packet to A. Upon reception of the IWTA packet, A (verifier) generates a challenge R_A , which it sends back to B. B also generates a challenge R_B . On both sides R_A is XORed with the concatenation of the master's BD ADDR and the master's clock to render BTR_A . Using these piconetspecific values prevents relay attacks as suggested by [24]. In step 6, B sends its challenge R_B , a hash of the concatenation of R_B and BTR_A , signed with B's private key. A then verifies the signature, and implicitly verifies that BTR_A was calculated correctly by B, that is the devices are in the same piconet. On both sides R_B is XORed with the concatenation of the master's BD ADDR and the master's clock to generate BTR_B . Finally, in step 7, A sends a hash of the concatenation of R_A and BTR_B , signed with A's private key. B verifies the signature, and implicitly verifies that BTR_B was calculated correctly by A, that is the devices are in the same piconet. Both devices are now successfully authenticated. The use of random values (R_A and R_B) is to prevent replay attacks of authentication replies.

VI.IMPROVED SCHEME

Even if an attacker captures the messages sent in steps 2 and 3 (mainly p , g^x , and g^y), calculating the key K is hard without knowing either x or y because of the discrete logarithm problem [24]. Furthermore, using only p , g^x and g^y to bruteforce the PIN is hard because of the fact that x and y are random. Even if an educated guess on the PIN is done and g is calculated, verification is hardly possible because of the random numbers involved. The device sending the IWTP and IWTA packets is protected from doing the heavy calculations first. An attacker can try to send the IWTP and IWTA packets to force the receiving device to perform heavy calculations. If an attacker sends the IWTP packet, the receiving device will generate a random number and use the function f to calculate g . This involves some calculation. In the following step, the receiving device calculates the key K and verifies the signature and MAC of the incoming message. This also involves some calculation. However, if the attacker sends something random, the signature and MAC will fail. As soon as verification fails the receiving device will terminate the connection. If an attacker sends the IWTA packet, the receiving device will generate a random number and send it back. In the following step, the receiving device verifies the signature of the incoming message. If it fails the connection is terminated.

I write his program and attacks program and view that result, that can view in table 4

```
Node1: PIN=123456
Node2: PIN=123456

*** Pairing between Node1 and Node2 ***
Node2: g=123456
Node2: y=53735942743
Node1: x=56930505086
Node1: p=65240177
Node1: gx=24499805
Node2: p=65240177
Node2: gx=24499805
Node2: gy=3927680
Node2: K=48752461
Node1: gy=3927680
Node1: K=48752461
Node1: *** Pairing Accepted ***
Node2: *** Pairing Accepted ***

*** Authentication between Node1 and Node2 ***
Node2: rb=-32924264961
Node1: ra=-33822605964
Node1: btra=-33822605954
Node2: btra=-33822605954
Node2: sb=-66746870915
Node1: btrb=-32924264971
Node1: sa=-66746870935
Node2: btrb=-32924264971
```

Figure 5. the successful pairing and Authentication between two device

Table4. the result experiment

Attacks	Current Bluetooth Protocol		proposed Bluetooth Protocol	
	Effective	Non-effective	Effective	Non-effective
Unit Key Attack	×			×
Offline PIN Cracking	×			×
Message Relay Attack	×			×
Device Impersonation	×			×
Denial-of-Service Attack on Authentication	×			×
DOS Attack on Pairing and Authentication	×			×
Online PIN Cracking	×			×
DOS Attack on Authentication	×		×	

The result show the new scheme be able to stop many attacks again Bluetooth.

The focus in this paper has been on pairing vulnerabilities in Bluetooth. Issues associated directly with integrity and confidentiality is out of the scope of this paper. This paper can be extended to fully develop a secure protocol or pairing and in Bluetooth. Further enhancements would be including per-packet integrity checks, as well as improving the encryption procedure in Bluetooth.

REFERENCES

- [1] Wikipedia.org. Bluetooth. Available: <http://en.wikipedia.org/wiki/Bluetooth> [Accessed 8 november 2007]
- [2] Definition of ISM band. Available: http://en.wikipedia.org/wiki/ISM_band
- [3] McCune, Perrig, and Reiter. "Seeing is believing: Using Camera Phones for Human-Verifiable Authentication". Carnegie Mellon University, 2005
- [4] BluetoothSIG. Bluetooth Core Specification. <http://www.bluetooth.org/foundation/adopters/document/Corev2.0EDR/en/1/Corev2.0EDR.zip>. [Accessed 8 September 2005].
- [5] T. Muller. Bluetooth Security Architecture.
- [6] <http://www.waterwood.com.cn/technology/bluetooth/documents/Security.pdf>, 1999. [Accessed 14 December 2007].
- [7] Yaniv Shaked and Avishai Wool, Cracking the Bluetooth PIN. Available: <http://www.eng.tau.ac.il/~yash/shaked-wool-mobisys05/>
- [8] http://en.wikipedia.org/wiki/Simple_Password_Exponential_Key_Exchange [Accessed 8 september 2007]
- [9] http://en.wikipedia.org/wiki/Station-to-Station_protocol [Accessed 8 september 2007]
- [10] Bellare, M.; Canetti, R.; Krawczyk, H. (1998), *A modular approach to the design and analysis of authentication and key exchange protocols*, Proceedings of the 3th Annual Symposium on the Theory of Computing
- [11] RFC 2412, "The OAKLEY Key Determination Protocol"
- [12] ISO/IEC 117703, "Mechanisms Using Asymmetric Techniques", (1999).
- [13] D.P. Jablon. Strong Password-Only Authenticated Key Exchange. Available: <http://www.jablon.org/jab96.pdf>. [Accessed 6 Januari 2007].
- [14] Bluetooth Pairing and Authentication Vulnerabilities, Protocol Design, Implementation, Dennis K. Nilsson, Department of Computer Engineering, Chalmers University of Technology, 2007
- [15] Bluetooth Security Protocol Analysis, The Faculty of the Department of Computer Science San Jose State University, Chi Shing Lee, May 2007
- [16] Tom Karygiannis and Les Owens. "Wireless Network Security 802.11, Bluetooth, and Handheld Devices". National Institute of Standards and Technology, Special Publication 800-48 Nov 2002
- [17] Bluetooth, <http://www.autoit.com/page.php?id=198>, 2008 -1-10

- [18] Bluetooth security, Thomas peyrin C.P.E lyon -E.T.I, Diploma project 20th September 2005
- [19] bart Prenee, Dave singelee, Security overview of Bluetooth ,1 june 2004
- [20] security Weaknesses in Bluetooth ,Luncent technologies-Bell lab
- [21] Informatin security Research center Murray Hill ,Nj 07974 USA
- [22] The Hidden Risks of Bluetooth, Aruba Joshua Wright ,2006
- [23] Brno University of Technology ,ON Security of Bluetooth system, 2006
- [24] ghermann Christian, Persson Joakim, Smeet Ben, Bluetooth Security, Artech House 2004
- [25] Karl E Persson and D. Manivanan. *Secure Connections in Bluetooth Scatternets*. Proceedings of the 36 th Hawaii International Conference on System Sciences, 2003.
- [26] Levi, Cetinatas, Aydos, et al. *Relay Attacks on Bluetooth Authentication and Solution*. Springer-Verlag Berlin Heidelberg, 2004.
- [27] Keijo M.J. Haataja. *Detailed descriptions of new proof-of-concept Bluetooth seuciry analysis tools and new secruti attacks*. Dept of Computer Science, University of Kuopio, 2005.