World Applied Sciences Journal 13 (4): 769-780, 2011
ISSN 1818-4952
© IDOSI Publications, 2011

Secure Public Key Exchange Against Man-in-the-Middle Attacks During Secure Simple Pairing (SSP) in Bluetooth

[1]*Iman ALMomani*, [1]*Mohammed Al-Saruri and* [2]*Mousa AL-Akhras*

[1]Computer Science Department, [2]Computer Information Systems Department
[1+2]King Abdullah II School for Information Technology, The University of Jordan

Abstract: Bluetooth version "Bluetooth 2.1+EDR" adopts a Secure Simple Pairing (SSP) procedure which is a secure method for establishing a Bluetooth connection that uses Diffie-Hellman public-key cryptography in its communication. Despite the high security mechanism provided by this method, it is still vulnerable to attacks, especially Man-In-The-Middle (MITM). Several solutions have been proposed to tackle this vulnerability but the process of securing the exchange of public keys in SSP was not taken into consideration, consequently this threatens the complete pairing process. In this paper a new method for securing the exchange of public keys between the communicating Bluetooth devices that uses SSP method is introduced. The details of the proposed method which is an Enhancement to the SSP (ESSP) and how security is assured are illustrated. Moreover, a case study is presented to demonstrate the effectiveness of the proposed ESSP. The paper finally discusses the security strength of the proposed ESSP against different types of MITM attacks as compared to other related work.

Key words: Bluetooth · Security · Secure Simple Pairing (SSP) · Man-In-The-Middle Attack · Public Key Authentication

INTRODUCTION

Bluetooth is an open wireless technology for short range wireless data and real-time voice exchange that is used in billions of devices and consumer electronics [1]. It operates in the license-free 2.4-2.4835 GHz frequency band. This band corresponds to the Industrial, Scientific and Medical (ISM) band using Frequency-Hopping Spread Spectrum (FHSS).

Bluetooth devices that communicate with each other form a piconet. The device that initiates a connection is the piconet master. One piconet can have a maximum of seven active slave devices and one master device. Piconet master and piconet slave(s) form a Bluetooth Personal Area Network (PAN). Like other technologies, Bluetooth technology faces some problems from attackers who are attempting to intrude data transmission through different methods.

In wired networks a Certificate Authority (CA) can be used for the purpose of securing the exchange of public keys. CAs are servers that can be used for verification; i.e. to certify the issued certificates which include the public keys. However, CAs cannot be used reliably in wireless networks. For example in Bluetooth technology it was proposed to nominate one of the piconet devices in the PAN to act as a CA. It was proposed that this device will generate the keys for all other devices. This proposal is not practical as the nominated device may go down or it may go out of range, consequently other devices relying on it will not be able to communicate unless the device acting as a CA certify their public keys.

Instead of using a CA, Bluetooth standard "Bluetooth 2.1+EDR" (Enhanced Data Rate) adopted a Secure Simple Pairing (SSP) procedure which is a secure method for establishing the Bluetooth connection that uses Diffie-Hellman public-key cryptography in its communication. However, the process of securing the exchange of public keys in the communicating Bluetooth devices that uses SSP method was not fully considered.

Accordingly, a new method is proposed in this paper to secure the exchange of public keys in SSP. The proposal is to add two phases. In the first phase that occurs before the SSP process even starts, the public keys and the Bluetooth device addresses are exchanged

Corresponding Author: Iman ALMomani, King Abdullah II School for Information Technology,
The University of Jordan. The University of Jordan, P.O. Box 13835, Amman 11942, Jordan.
Tel: + 962 (0) 6 53 55 000 Ext. 22602, Fax: + 962 (0) 6 53 00 233. E-mail: i.momani@ju.edu.jo.

and a unique password for each device's public key is assigned. The second phase occurs during the exchange of public keys in the SSP. In the second phase each device must encrypt its unique password with the other device's public key which was exchanged previously. Next the destination device decrypts the password and compares it with its own as will be shown in the following sections.

By adopting the proposed technique, the exchange of public key becomes more secure and consequently, the process of SSP will be secure, reliable and provide protection against Man-In-The-Middle (MITM) attacks.

The rest of this paper is organised as follows. Section 2 gives an overview about Bluetooth security. Section 3 introduces the Bluetooth SSP. Section 4 reviews some possible attacks on SSP and some proposed solutions from the literature. The proposed approach is introduced and fully analysed in section 5. Section 6 presents a case study to demonstrate the proposed ESSP; the section also illustrates the security strength of the proposed ESSP against different types of MITM attacks. Finally conclusions are drawn in section 7.

Bluetooth Security: Due to the open nature of the wireless media, the transmissions can be easily intercepted or jammed causing fake or modified information to be injected and delivered to the piconet devices. An attacker can use a "powerful directional antennas" in order to increase the scanning, eavesdropping and attacking range to be able to attack any kind of Bluetooth device. BlueSniper Rifle [1, 2] is a good example of a device with powerful antennas that can scan over a mile away from the target devices. In addition, anyone can transform a standard 30 USD Bluetooth dongle into a full-blown Bluetooth sniffer [3].

Because of the security threats and problems imposed by the use of Bluetooth technology [4], several Bluetooth security techniques have been developed. The basic configuration of Bluetooth security is done by the user who chooses the device's discoverability and connectivity options. The discoverability and connectivity capabilities and their combinations can be divided into three security levels [5]:

- Silent: The device does not accept any connections. It only observes Bluetooth traffic.
- Private: The device is not discoverable and can accept connections only if its unique Bluetooth Address (BD_ADDR) is known by the master device.
- Public: The device is discoverable and can be connected to.

According to Scarfone and Padgette [6], each Bluetooth device must operate in one of four different security modes. These modes are:

- Non-secure: Bluetooth devices do not initiate any security measure.
- Service-level Enforced Security Mode: Bluetooth devices initiate security procedures after Link Manager Protocol (LMP) link establishment and before Logical Link Control and Adaptation Protocol (L2CAP) channel establishment. This mode supports different security measures such as Authentication, Authorisation and Confidentiality [7]. Authentication is used to proof the identity of the communicating devices. Authorisation controls the access of the devices to the provided services. Confidentiality of data that is exchanged between Bluetooth devices is achieved using different encryption mechanisms.
- Link-Level Enforced Security Mode: Security procedures are initiated before the physical link is fully established.
- Service-level Enforced Security Mode for SSP Devices: This mode is similar to mode 2, except that only Bluetooth devices that adopt SSP can use it. Consequently, it is a mandatory mode to be available for Bluetooth devices "Bluetooth 2.1+EDR" or later.

Bluetooth devices usually use Secure And Fast Encryption Routine + (SAFER+) block-based cipher that applies 128-block size [8]. It is used in Bluetooth as an algorithm for key derivation and authentication. Although of the existence of some optimisations for faster breaking of SAFER+, it is still considered secure [9, 10].

Before Bluetooth standard "Bluetooth 2.1+EDR" was released, any two devices that would like to communicate need to start a pairing process. During the paring process, the two involved devices generate the same shared secret. This generation depends on one source of entropy that is called Personal Identification Number (PIN) code. If the users enter the same PIN code or passkey in both devices, the same shared secret will be created. The security problem is resulted from the short length of the PIN code (often four decimal digits) which can be easily cracked. Even with longer 16-character alphanumeric PINs, full protection against different eavesdroppers cannot be achieved. This is especially true with the existence of an On-Line PIN cracking attacks that are able to discover the secret PIN code of the victim device [11-13].

Bluetooth standard “Bluetooth 2.1+EDR” added a new specification for the pairing procedure, namely SSP [9]. SSP aims to protect the pairing process from different types of eavesdropping and MITM attacks. In order to achieve this, SSP employs Elliptic Curve Diffie-Hellman public-key cryptography [14] instead of using the often short passkeys as the only source of entropy to build the link keys. The link key is constructed using: devices' public-private keys, collection of nonces and Bluetooth addresses of the devices. SSP effectively thwarts passive eavesdropping as it is infeasible to run an exhaustive search to crack a private key with approximately 95 bits in a short period of time [5].

Bluetooth Secure Simple Pairing (SSP): Before any two Bluetooth devices start communicating, pairing must be performed. As a result of the pairing, two devices form a trusted pair and a link key is established which is used later for creating a data encryption key for each session.

There are four association models that are used in SSP. Choosing an association model depends on the device capabilities. The first one is *Numeric Comparison* where both devices have the ability to display six digits and enter “Yes” or “No”. The second one is *Just Works* which is used when at least one device has displaying capabilities but no keyboard for entering six digits. *Out of Bound (OOB)* is the third association model (e.g. Near Field Communication, NFC) that is used for scenarios using OOB mechanism for both discovering the devices and exchanging the cryptographic numbers used in the pairing process. Finally, the *Passkey Entry* association model is designed for scenarios where one device has input capabilities and no ability to display six digits and the other device has output capabilities.

SSP comprises of six phases, these phases are [9, 15]:

- Capabilities exchange: The devices in this phase exchange their Input/Output (IO) capabilities to determine the best association model to be used. This phase takes place when the devices had never met before or when they want to re-perform the pairing process for some reason.
- Public key exchange: To initiate pairing, devices exchange their public keys and then Diffie-Hellman key is computed.
- Authentication stage 1: This phase depends on IO capabilities of the two devices and consequently on the used association model. It aims to provide protection against MITM attacks. It is achieved by exchanging set of nonces, commitments to the nonces and the exchanged public keys to check their integrity.
- Authentication stage 2: This phase is the same in all association models. It confirms that the public keys exchange ended successfully.
- Link key calculation: Once pairing is confirmed by both devices, the link key is computed using their Bluetooth addresses, nonce values and the Diffie-Hellman key generated in phase 2.
- Link Manager Protocol (LMP) authentication and encryption: This is the final phase in the SSP where the encryption keys are generated. It is similar to the one used in the legacy pairing.

BT-NIÑO-MITM Attack: A SSP-specific attack called Bluetooth -No Input, No Output - Man-In-The-Middle attack (BT-NIÑO-MITM) has been identified [5, 16].

This attacker benefits from the first phase of the SSP when the Information related to the IO capabilities is exchanged over an unauthenticated channel. This allows the attacker to modify such information and enforce the devices to use a less secure association model such as *Just Works* which does not provide protection against MITM attacks. BT-NIÑO-MITM attack uses two Bluetooth devices with modifiable BD_ADDRs for attacking purposes. Such devices are available in the market. The attacker copies the BD_ADDRs of the victim devices and their names to impersonate them.

The main idea of the attack is shown in Figure 1. The attacker disturbs (jams) the physical layer (PHY) through hopping along with the victim devices and send fake data at every timeslot. Another possibility is to jam the whole 2.4 GHz band by using a wideband signal in order to shut down all piconets within the range to frustrate the user and let her/him thinks that there is something wrong with the Bluetooth devices and delete previously stored link keys to enforce her/him initiating a new pairing process using SSP. In this case, the BT-NIÑO-MITM attack will interfere the exchanging of messages related to IO capabilities in order to force the victim devices to use the *Just Work* association model. Then the attacker continues as illustrated in the Figure 2. The notation used in SSP is listed in Table 1.

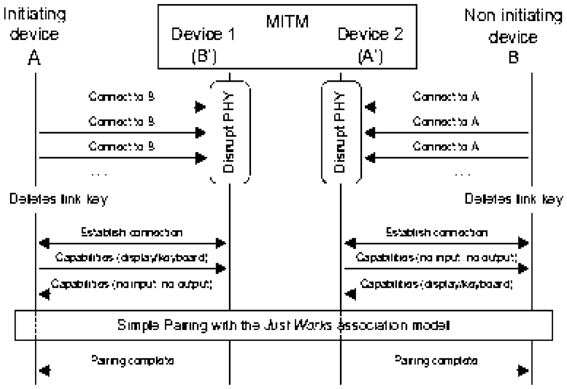


Fig. 1: The Main Idea of the Attack [16]

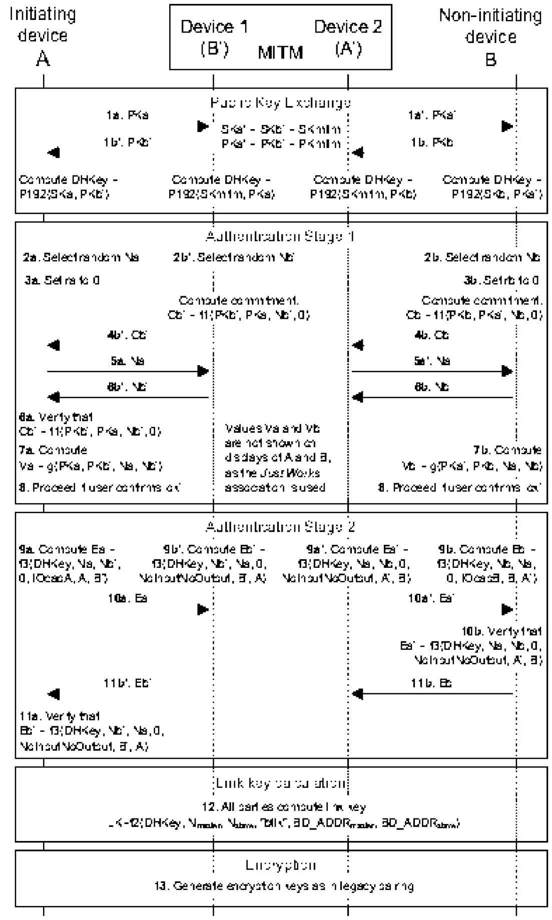

Initiating device A
Device 1 (B')
MITM
Device 2 (A')
Non-initiating device B
Public Key Exchange
1a. PKa
1b'. PKb'
SKa' = SKb' = SKmitm
PKa' = PKb' = PKmitm
1a'. PKa'
1b. PKb
Compute DHKey = P192(SKa, PKb')
Compute DHKey = P192(SKmitm, PKa)
Compute DHKey = P192(SKmitm, PKb)
Compute DHKey = P192(SKb, PKa')
Authentication Stage 1
2a. Select random Na
2b'. Select random Nb'
2b. Select random Nb
3a. Set ra to 0
3b. Set rb to 0
Compute commitment, Cb' = f1(PKb', PKa, Nb', 0)
Compute commitment, Cb = f1(PKb, PKa', Nb, 0)
4b'. Cb'
4b. Cb
5a. Na
5a'. Na
6b'. Nb'
6b. Nb
6a. Verify that Cb' = f1(PKb', PKa, Nb', 0)
7a. Compute Va = g(PKa, PKb', Na, Nb')
8. Proceed if user confirms ok
Values Va and Vb are not shown on displays of A and B, as the Just Works association is used
7b. Compute Vb = g(PKa', PKb, Na, Nb)
8. Proceed if user confirms ok
Authentication Stage 2
9a. Compute Ea = f3(DHKey, Na, Nb', 0, IOcapA, A, B')
9b'. Compute Eb' = f3(DHKey, Nb', Na, 0, NoInputNoOutput, B', A)
9a'. Compute Ea' = f3(DHKey, Na, Nb, 0, NoInputNoOutput, A', B)
9b. Compute Eb = f3(DHKey, Nb, Na, 0, IOcapB, B, A')
10a. Ea
10a'. Ea'
10b. Verify that Ea' = f3(DHKey, Na, Nb, 0, NoInputNoOutput, A', B)
11b'. Eb'
11b. Eb
11a. Verify that Eb' = f3(DHKey, Nb', Na, 0, NoInputNoOutput, B', A)
Link key calculation
12. All parties compute link key
LK=f2(DHKey, Nmaster, Nslave, "btlk", BD_ADDRmaster, BD_ADDRslave)
Encryption
13. Generate encryption keys as in legacy pairing

Fig. 2: Pairing Details [16]

It is worth noting that in this first scenario two victim devices that have already performed initial pairing (including the capabilities exchange). Therefore, the link keys are saved on the devices for use in subsequent connections; i.e. the victim devices normally use SSP without capabilities exchange.

Table 1: Protocol Notation (extracted from 15)

Term	*Definition*
PKx	Public key of device X
SKx	Private key of device X
DHKey	Diffie-Hellman key generated after key exchange
Nx	Nonce generated by device X
Rx	Random number generated by device X; equals 0 in the Numeric Comparison association model
Cx	Commitment value from device X
f1	One-way function used to compute commitment values
f2	One-way function used to compute the link key
f3	One-way function used to compute check values
G	One-way function used to compute numeric check values
IOcapX	Input/Output capabilities of device X
BD_ADDR	48-bit Bluetooth device address

Another two scenarios for the BT-NIÑO-MITM occur when the victim devices have never met before. In such scenarios the devices are easier to be attacked. This is because in such scenarios there is no need to disrupt the physical layer. The two scenarios are [17]:

- The victim device (A or B) initiates SSP: The attacker waits until one of the devices attempts to initiate SSP and then proceeds as demonstrated in Figure 1 and Figure 2.
- The BT-NIÑO-MITM (A' or B') initiates SSP: The attacker initiates SSP with the victim devices and then proceeds as demonstrated in Figure 1 and Figure 2. It may be possible to perform SSP even without involving the user to accept the connection. That depends on the implementation of the victim devices.

After a successful attack, the BT-NIÑO-MITM attacker will be able to intercept and modify all the messages exchanged between the victim devices.

Some solutions have been proposed to tackle the above attacking scenarios on SSP, but none of these solutions was effective. Haataja and Hyppönen [5] proposed adding an extra message to the SSP to be used when *Just works* association model is used. This message says "The second device has no display and keyboard! Is this true?", then the user may choose either to "Proceed" or to "Stop". The problem of such proposal is when a hacker attempts to mislead the user that the device she/he is communicating with has not any IO capabilities although it may have, if that device is far away may be the user will accept the connection.

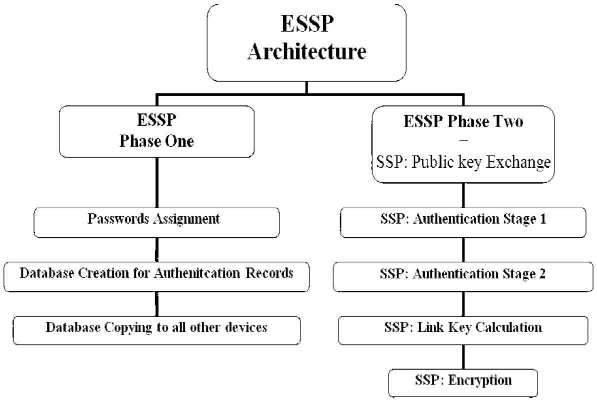

ESSP Architecture
ESSP Phase One
ESSP Phase Two – SSP: Public key Exchange
Passwords Assignment
Database Creation for Authenitcation Records
Database Copying to all other devices
SSP: Authentication Stage 1
SSP: Authentication Stage 2
SSP: Link Key Calculation
SSP: Encryption

Fig. 3: Enhanced SSP (ESSP) Architecture

It is also proposed to use OOB as a mandatory association model [16, 18]. The problem of such proposal is that in OOB the devices need to be near each other every time they want to communicate and start the SSP six phases. Moreover, the devices should have special capabilities to support OOB connections which make it limited in its use. Finally, OOB does not support a user that activated a connection using Bluetooth technology and would like to apply OOB for authentication during a connection. Another work is presented in [19] which suggests adding more values to secure Authentication Stage 1 of the SSP, but this suggestion will not solve the interference of the MITM attacks before this stage while exchanging the public keys.

The proposed method in this paper provides a secure exchange of the public keys. This will prevent any interception from MITM attacks and consequently provide a successful SSP. It needs to be performed only once, when a new device is bought to be used in the Bluetooth network. The devices are not necessarily need to be near each other to communicate securely; they can be within the maximum range the Bluetooth technology can support. Finally, it works for new and old devices.

The Proposed Enhanced SSP (ESSP) Method: The current solutions for the MITM attacks on SSP did not solve the main problem which is how to exchange the public keys securely so that no attacker can intercept the exchanged messages.

The proposed method provides an enchantment to the SSP (ESSP) to authenticate the public keys of the communicating parties. ESSP adds two phases to secure the exchange of public keys as depicted in Figure 3. The first phase of the ESSP occurs before starting the SSP process. This phase is executed once, when new devices are added to the Personal Area Network (PAN) of the user. The second phase of the ESSP occurs during the exchange of the public keys in SSP so that no attacker can operate in the first phase of the SSP six phases. In other words, if an attacker attempts to get into the Bluetooth network it will be detected as will be illustrated shortly.

Table 2: ESSP Notation

Term	*Definition*
Pub.X	Public key of device X
Pri.X	Private key of device X
Pass.X	Password assigned for Public key of X (Pub.X)
DB.X	Database at device X
BD_ADDR.X	Bluetooth device address of X
Epub.X (Pass.Y)	Encrypts the password of Y's device with the public key of X's device.
Dpri.X (Pass.Y)	Decrypts the password of Y's device with the private key of X's device.

In the first phase of the proposed method, the public keys and Bluetooth device addresses are exchanged and a unique password is assigned for each device's public key in all devices. This phase occurs before the SSP process starts and it is done just once when a new device is added. The abbreviations used in the proposed system diagrams are shown in Table 2.

Phase one of the proposed method is shown in Figure 4 and is illustrated in the following steps:

1. Choose one of the devices to act as a main device for distribution purposes. This main device is used to store the whole records of all devices. The process starts by asking the device that is chosen to be main (A's device in Figure 4) if it has a password for its public key. If not, the user will enter a password for A's public key and a record is created for A's public key, Bluetooth device address (BD_ADDR.A) and the newly created password (Pass.A) and proceed to the next step. On the other hand, if there is a password, A proceeds to the next step without recreating a new password.

Repeat steps 2 – 10 for all devices

2. The same question is repeated to the other device(s) (e.g. B's device) if it has a password for its public key. If not, the user will enter a password for B's public key and a record is created for B's public key, Bluetooth device address (BD_ADDR.B) and the newly created password (Pass.B) and proceed to the next step.

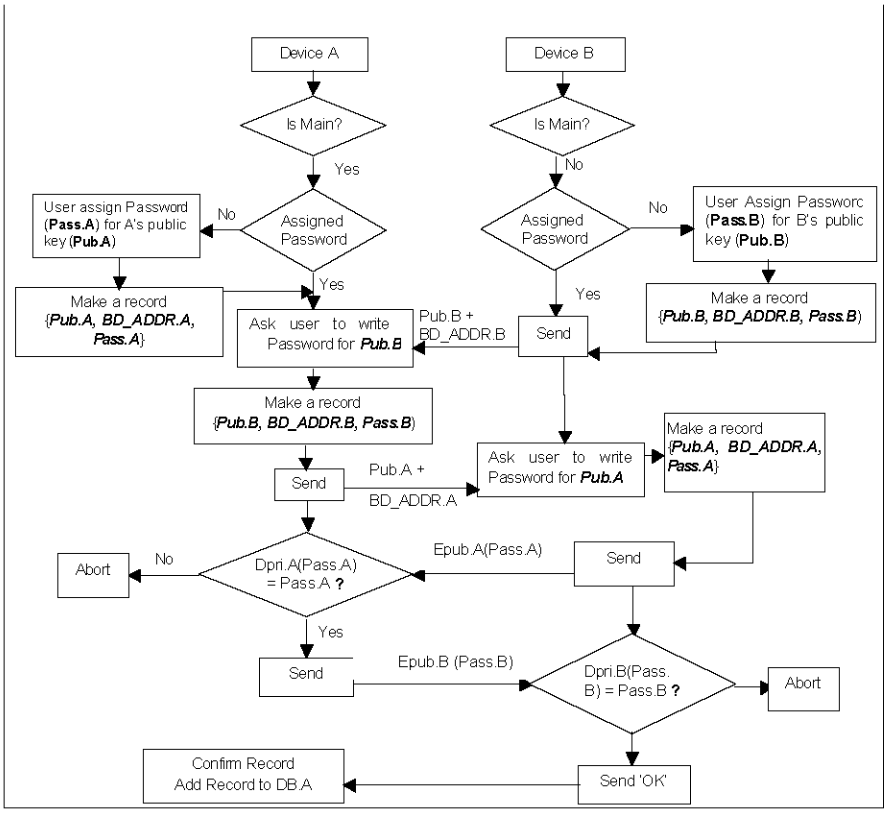

Device A
Device B
Is Main?
Is Main?
Yes
No
User assign Password (Pass.A) for A's public key (Pub.A)
No
Assigned Password
Assigned Password
No
User Assign Passworc (Pass.B) for B's public key (Pub.B)
Yes
Yes
Make a record {Pub.A, BD_ADDR.A, Pass.A}
Ask user to write Password for Pub.B
Pub.B + BD_ADDR.B
Send
Make a record {Pub.B, BD_ADDR.B, Pass.B)
Make a record {Pub.B, BD_ADDR.B, Pass.B)
Ask user to write Password for Pub.A
Make a record {Pub.A, BD_ADDR.A, Pass.A}
Send
Pub.A + BD_ADDR.A
Abort
No
Dpri.A(Pass.A) = Pass.A ?
Epub.A(Pass.A)
Send
Yes
Send
Epub.B (Pass.B)
Dpri.B(Pass. B) = Pass.B ?
Abort
Confirm Record Add Record to DB.A
Send 'OK'

Fig. 4: Phase One of the Proposed Method (part 1)

3. The public key is sent from the B's device (Pub.B) to the A's device accompanied with the B's BD_ADDR (BD_ADDR.B).
4. In A's device, the same password associated with the received B's public key will be entered and a record will be created for it. This record contains the received B's public key and BD_ADDR.B and B's password (Pass.B).
5. In B's device, the same password associated with the received A's public key will be entered and a record will be created for it. This record contains the received A's public key and BD_ADDR.A and A's password (Pass.A).
6. Reply to the A's device by sending to it the password (created in 3) encrypted with the A's public key (Epub.A (Pass.A)) that was sent previously.
7. In A's device the received encrypted password will be decrypted (Dpri.A (Pass.A)) and compared with A's password stored in the A's record. If they match, the next step will be performed.
8. A's device will send the password of 'B' encrypted with the public key of 'B' (Epub.B (Pass.B)) to B's device.
9. In B's device the received encrypted password will be decrypted (Dpri.B (Pass.B)) and compared with B's password stored in the B's record. If they match, B's device will send 'OK' message to A's device.
10. A's device confirms the process and save the B's record at A's DataBase (DB.A).

The flow chart illustrating the creation of the new database is depicted in Figure 4.

11. Once A's device has finished with all devices, the complete database will be ready for copying to all other devices.
12. The whole database is copied from the main device (A's device DB.A) to all other devices that have record in DB.A such as B's device. The A's device uses B's public key to encrypt the Database (Epub.B(DB.A)) and then it sends it with the BD_ADDR.A to B.
13. In B's device, check if BD_ADDR.A is stored from the previous steps of database creation then decrypt the encrypted database (Dpri.B(DB.A)) and save it.

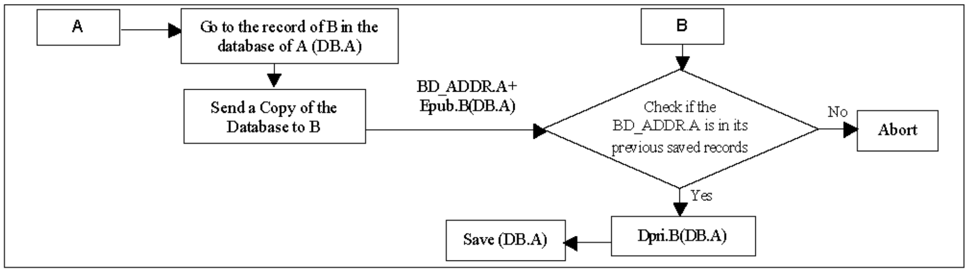

A
Go to the record of B in the database of A (DB.A)
Send a Copy of the Database to B
BD_ADDR.A+ Epub.B(DB.A)
B
Check if the BD_ADDR.A is in its previous saved records
No
Abort
Yes
Dpri.B(DB.A)
Save (DB.A)

Fig. 5: Phase One of the Proposed Method (Part 2, Copying the database)

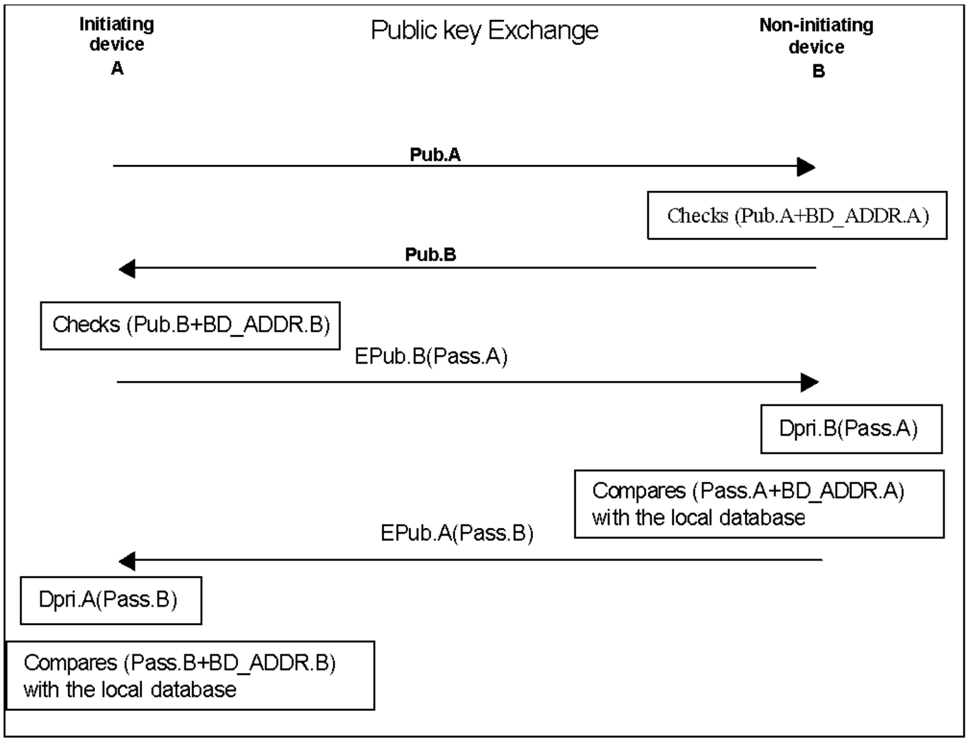


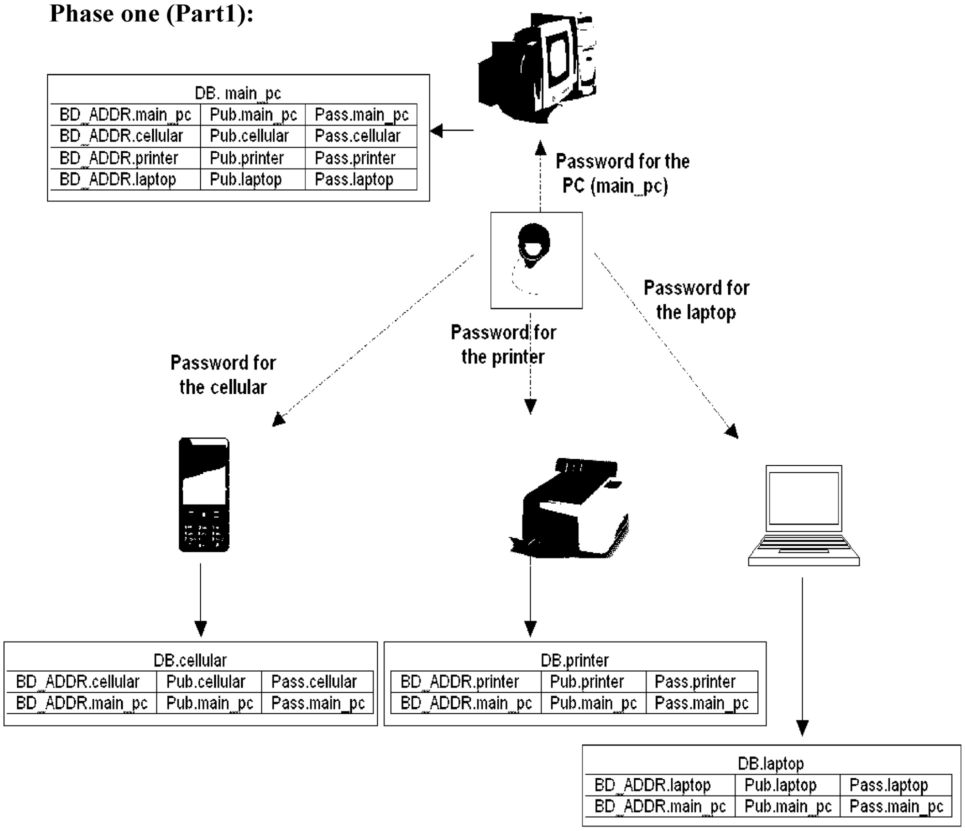
Fig. 6: Phase Two of the Proposed Method

The flow chart for copying the new database is depicted in Figure 5.

The second phase begins while exchanging public keys in the SSP process. This phase is done by the following steps as illustrated in Figure 6:

1. The verifier device checks in its database the public key received from the initiating device and the associated BD_ADDR, given in the inquiry step of scanning devices in the range.
2. The verifier device then sends its public key to the initiating device.
3. The initiating device checks in its database the public key received from the verifier device and the associated BD_ADDR.
4. The initiating device sends its own password encrypted with the public key of the verifier.
5. The verifier device decrypts the encrypted password and compares this information sent and the BD_ADDR of the initiating device with its own recorded information in the database.
6. The verifier device then sends its own password encrypted with the initiating device public key.
7. The initiating device decrypts the encrypted password and compares it and the sent BD_ADDR of the verifier with its corresponding recorded information in the database.

Phase one (Part1):



DB. main_pc

BD_ADDR.main_pc	Pub.main_pc	Pass.main_pc
BD_ADDR.cellular	Pub.cellular	Pass.cellular
BD_ADDR.printer	Pub.printer	Pass.printer
BD_ADDR.laptop	Pub.laptop	Pass.laptop

DB.cellular

BD_ADDR.cellular	Pub.cellular	Pass.cellular
BD_ADDR.main_pc	Pub.main_pc	Pass.main_pc

DB.printer

BD_ADDR.printer	Pub.printer	Pass.printer
BD_ADDR.main_pc	Pub.main_pc	Pass.main_pc

DB.laptop

BD_ADDR.laptop	Pub.laptop	Pass.laptop
BD_ADDR.main_pc	Pub.main_pc	Pass.main_pc

Fig. 7: Case Study: Phase One (Part 1) of the Proposed Method

DISCUSSIONS AND ANALYSIS

In this section a case study will be presented in order to illustrate the proposed security method (ESSP). Moreover, different security attackers that could exist before or during the SSP process will be addressed. The effectiveness of the proposed method in defending the SSP process from such attacks will be explained.

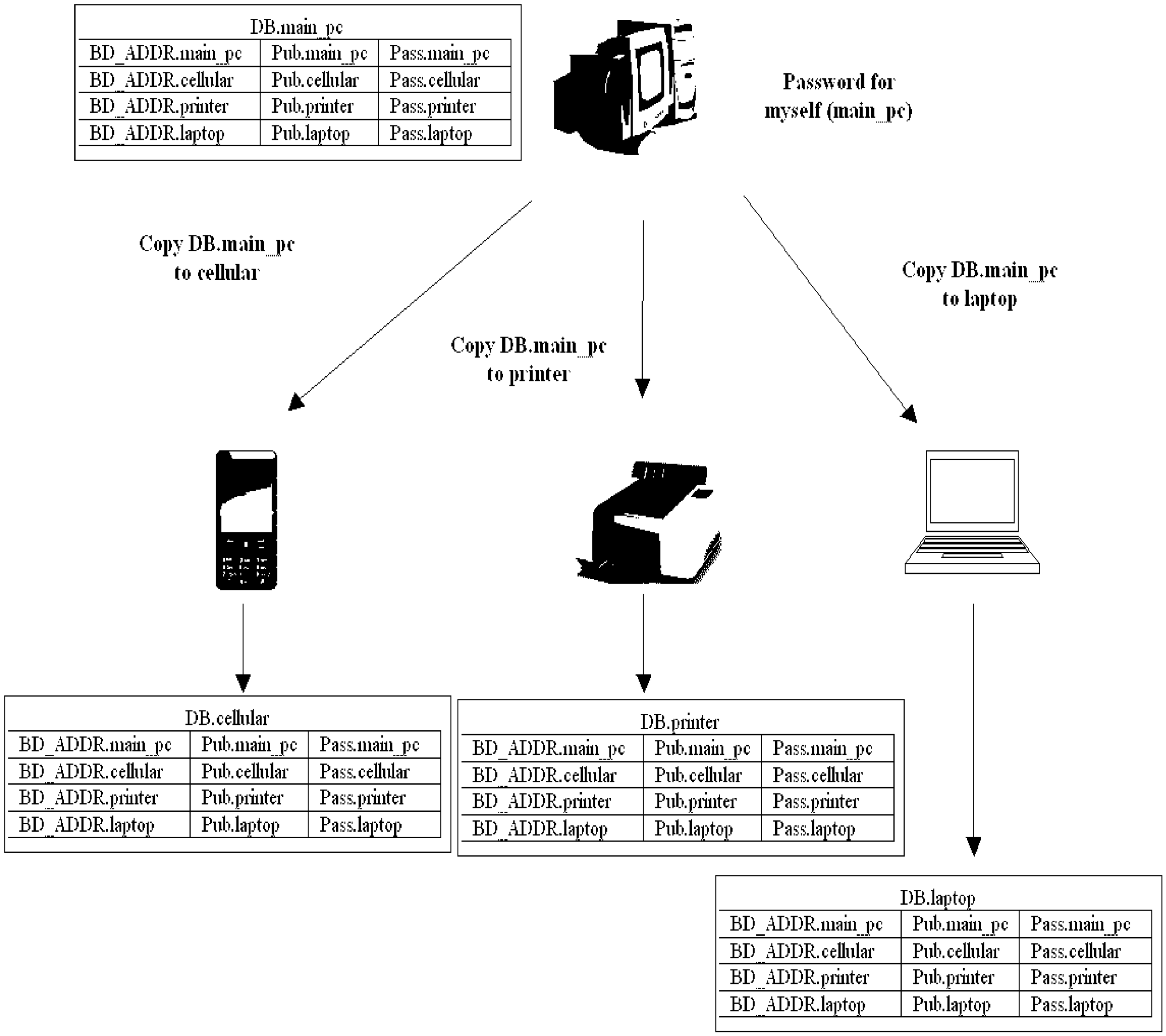
Case Study Illustrating the Proposed Technique: This section presents a case study for a Personal Area Network (PAN) consisting of personal computer (PC), Laptop, cellular phone and printer. This PAN is using Bluetooth connections enhanced by the proposed security approach presented in this paper (ESSP). Figure 7 shows part one of phase one of the ESSP. The user of the PAN starts by choosing the PC to act as the main device. Then the user assigns passwords to the PC and to all other devices. After that, the public keys and the Bluetooth addresses will be sent from all devices to the PC (main device) and confirmed by asking the PC to write the corresponding passwords for all devices. The user will enter passwords that are already assigned to these Bluetooth devices. In contrast, the PC will send its public key and Bluetooth address to the three other devices and this information is also confirmed by asking the three devices to write the password of the PC as illustrated before in Figure 4.

At the end of the this phase, the Main PC will have records for all the Bluetooth devices in the PAN where each other device (laptop, cell phone and printer) will have records about itself and the PC only as shown in Figure 7.

Part two of phase one aims to distribute the complete database from the main device (pc-main) to the other three devices. This database will be sent encrypted using the intended device's public key. At the end of this part, all devices will have the complete database that contains records about all devices in the PAN as shown in Figure 8.

Phase one of the ESSP is executed once when the PAN network is firstly configured and in case of adding a new device(s) to the PAN. Phase two is added at the top of the normal SSP stages which will ensure the authenticity of the exchanged public keys between the communicating parties that will prevent any interference from MITM attacks that could lead to failed SSP process.

Phase one (part2), copying database:


Password for myself (main_pc)
Copy DB.main_pc to cellular
Copy DB.main_pc to printer
Copy DB.main_pc to laptop

DB.main_pc

BD_ADDR.main_pc	Pub.main_pc	Pass.main_pc
BD_ADDR.cellular	Pub.cellular	Pass.cellular
BD_ADDR.printer	Pub.printer	Pass.printer
BD_ADDR.laptop	Pub.laptop	Pass.laptop

DB.cellular

BD_ADDR.main_pc	Pub.main_pc	Pass.main_pc
BD_ADDR.cellular	Pub.cellular	Pass.cellular
BD_ADDR.printer	Pub.printer	Pass.printer
BD_ADDR.laptop	Pub.laptop	Pass.laptop

DB.printer

BD_ADDR.main_pc	Pub.main_pc	Pass.main_pc
BD_ADDR.cellular	Pub.cellular	Pass.cellular
BD_ADDR.printer	Pub.printer	Pass.printer
BD_ADDR.laptop	Pub.laptop	Pass.laptop

DB.laptop

BD_ADDR.main_pc	Pub.main_pc	Pass.main_pc
BD_ADDR.cellular	Pub.cellular	Pass.cellular
BD_ADDR.printer	Pub.printer	Pass.printer
BD_ADDR.laptop	Pub.laptop	Pass.laptop

Fig. 8: Case Study: Phase One (part 2) of the Proposed Method

The pairing process is performed successfully by authenticating the public keys of the communicating devices as illustrated in the former SSP stages shown in Figure 6.

ESSP Attackers: This section discusses the security strength of the proposed ESSP against different types of MITM attacker.

Scenario 1: the Attacker Starts at the Beginning of Phase One in ESSP When Exchanging Public Keys: The attacker in this scenario attempts to impersonate both the printer and the PC of the user such that any request sent from the user's PC will go to the attacker printer, the user printer will be connected to the attacker laptop instead of the user's PC. This scenario if applied on the original SSP it will not be discovered by the user while in the ESSP it will be discovered and prevented as can be seen in Figure 9.

Scenario 2: the Attacker Attempts to Start by Jamming the Physical Layer and Enters at the Beginning of Phase Two of ESSP: The MITM attacker in this scenario attempts to intercept the public key exchange between the pairing entities. If this occurs in the original SSP, the attacker will succeed as illustrated in the literature. But because of applying the ESSP in this scenario, the attacker will fail because the pubic keys are authenticated and stored through phase one of the ESSP, as shown in Figure 10.

		MITM (Attacker)		
Steps	main_pc	attacker_printer	attacker_laptop	main_pc_printer
1	User assigns password for main_pc (Pass.main_pc) (e.g. "123")	Attacker assigns password for attacker_printer (Pass.attacker_printer) as if it is main_pc_printer (e.g. "666")	Attacker assigns password for attacker_laptop (Pass.attacker_laptop) as if it is main_pc (e.g. "777")	User assigns password for main_pc_printer (Pass main_pc_printer) (e.g. "345")
2	{Pub.attacker_printer, BD_ADDR.attacker_printer } ←		{Pub.main_pc_printer, BD_ADDR.main_pc_printer} ←	
3	A message asks user to enter the password for attacker_printe. User enters the same assigned password for main_pc_printer which is '345'		A message asks the attacker to enter the password for main_pc_printer. Attacker is not aware of the correct password of main_pc_printer, So attacker enters any password (e.g. "999")	
4	Pub.main_pc+ BD_ADDR.main_pc →		Pub.attacker_laptop + BD_ADDR.attacker_laptop →	
5		Attacker could not guess the password of main_pc let say '888' in order to save a record		Main will write the same password of main_pc which is '123'
6	Epub.main_pc('888') ←		Epub.attacker_laptop('123') ←	
7	No match between main_pc password which is '123' and the received one '888'			
8	Abort			

Fig. 9: MITM Attack at the Beginning of Phase One of the Proposed Method

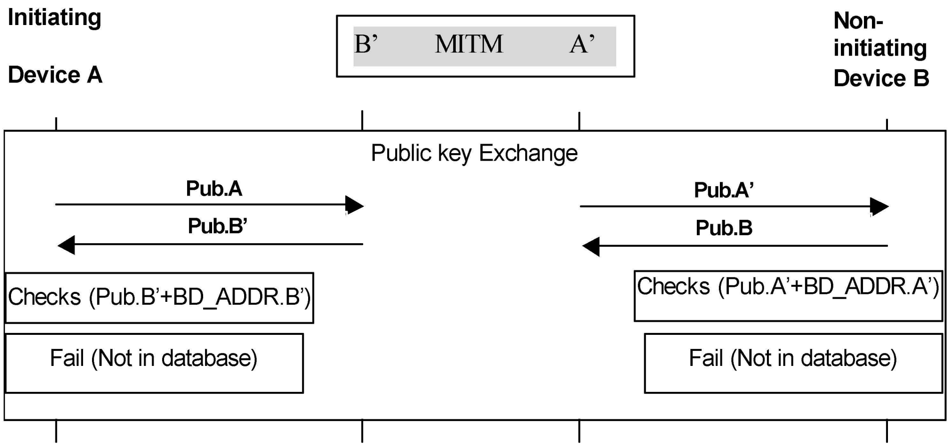


Fig. 10: MITM Failed Attacking in ESSP

Discussions: There are still some points that need to be addressed regarding the proposed method:

- ESSP is performed similar to SSP except for the checking of public keys which is added at the beginning. If any hacker attempted to jam the whole band and the user has deleted the final link key because of the noise, the database in the proposed ESSP method will not be deleted and simply the ESSP will be re-executed and it will automatically check the public keys again in order to stop hackers.

- In the previous related papers the researchers attempted to add a new message in order to inform the user if the device she/he is communicating with is using *Just Works* association model. The problem is when a hacker attempts to mislead the user that the device she/he is communicating with has not any IO capabilities although it may have. So if that device is far away may be the user will accept the device but in our method checking public keys in the background will solve the problem automatically in the first stage of ESSP.
- Passwords used in the proposed method are just used once when exchanging public keys. So it is impossible to attempt to guess the right password in one hit since the passwords are also encrypted with public keys.
- The proposed ESSP in comparison with the OOB. In OOB, the devices are necessarily near each other every time they communicate. In ESSP, only during the password assignments they need to be near each other to easily write the password on them. The devices are not necessarily need to be very close in order to communicate securely; they can be within the maximum range the Bluetooth technology can support. Additionally, the Bluetooth devices do not need any special capabilities in order to execute the proposed ESSP.

CONCLUSIONS

In this paper, some Bluetooth security issues were introduced and then the weaknesses of the old security methods are highlighted. The main steps involved in Secure Simple Pairing (SSP) adopted in "Bluetooth 2.1+EDR" are illustrated, the problem in SSP is focused when the attacker forces the devices to use the *Just Works* association model by being present right from the beginning of the SSP process or by jamming the whole Bluetooth Band.

Some researchers proposed the use of OOB and an extra message to confirm the capabilities of the communicating devices. Because of the limitations of these solutions a method is proposed in this paper for securing the exchange of public keys which consists of two main phases. The first one occurs before the SSP process and the second one occurs during the first step of SSP process. The proposed approach is an Enhanced SSP (ESSP) that ensures the authentication of the exchanged public keys against impersonating initiated by MITM attacks and consequently ensures a successful SSP. A case study has been introduced to illustrate the proposed ESSP. Moreover, the strength of the ESSP is explained by introducing the defence against different types of MITM attackers that could take place before and during the pairing process.

REFERENCES

1. Cheung, H., 2005. How To: Building a BlueSniper Rifle - Part 1. SmallNetBuilder, Pudai LLC, [Online]. Available: http:// www.smallnetbuilder.com/ content/ view/ 24256/98, Published March 2005, last access October 2010.
2. Cheung, H., 2005. How To: Building a BlueSniper Rifle-Part 2. SmallNetBuilder, Pudai LLC, [Online]. Available: http:// www.smallnetbuilder.com/ content/ view/ 24228/98, Published August 2005, last access November 2010.
3. Moser, M., 2007. Busting The Bluetooth Myth-Getting RAW Access. Remote-exploit.org, Research Report, [Online]. Available: http://www.remote-exploit.org/wp-content/ uploads/ 2010/ 01/ busting_bluetooth_myth.pdf, Published 2007, last access November 2010.
4. Suri, P.R. and S. Rani, 2008. Bluetooth security - Need to increase the efficiency in pairing, *IEEE Southeastcon 2008*, pp: 607-609, 3-6 April 2008.
5. Haataja, K.M.J. and K. Hyppönen, 2008. Man-In-The-Middle attacks on bluetooth: a comparative analysis, a novel attack and countermeasures. In: Proceedings of *3rd International Symposium on Communications, Control and Signal Processing, 2008 (ISCCSP 2008)*, pp: 1096-1102, 12-14 March 2008.
6. Scarfone, K. and J. Padgette, 2008. Guide to Bluetooth Security, Technical Report Special Publication SP 800-121, National Institute of Standards and Technology (NIST).
7. Ramli, D.A., S. Abdul Samad and A. Hussain, 2010. A Correlation Filter Based Biometric Speaker Authentication Systems, World Appl. Sci. J., 9(3): 259-267.
8. Massey, J. Khachatrian G. and M. Kuregian, 1998. SAFER+. In: Proceedings of the *1st Advanced Encryption Standard Candidate Conference*, 20-22 August 1998. National Institute of Standards and Technology (NIST), Ventura, CA.
9. Bluetooth, S.I.G., 1999-2007. Bluetooth specifications 1.0, 1.1, 1.2, 2.0+EDR and 2.1+EDR. Technical specifications, https://www.bluetooth.org.
10. Shaked, Y. and A. Wool, 2005. Cracking the Bluetooth PIN. In: Proceedings of the 3rd *International Conference on Mobile Systems, Applications and Services* (MobiSys'2005), pp: 39-50, 6-8 June 2005, Seattle, Washington.

11. Whitehouse, O., 2004. @Stake - Where Security & Business Intersect. Research report, CanSecWest/core04, Vancouver, [Online]. Available: http://cansecwest.com/csw04archive.html, last access October 2010.
12. Haataja, K.M.J., 2005. Two Practical Attacks Against Bluetooth Security Using New Enhanced Implementations of Security Analysis Tools. In: Proceedings of the *2nd IASTED International Conference on Communication, Network and Information Security* (CNIS'2005), pp: 13-18, 14-16 November 2005, Phoenix, Arizona, USA.
13. Haataja, K.M.J., 2008. New efficient intrusion detection and prevention system for Bluetooth networks. In: Proceedings of the *1st international conference on MOBILe Wireless MiddleWARE, Operating Systems and Applications*, 13-15 February 2008, Innsbruck, Austria.
14. Barker, E. D. Johnson and M. Smid, 2007. Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, Technical Report Special Publication SP 800-56A, National Institute of Standards and Technology (NIST).
15. Core Specification Working Group, 2006. Simple Pairing Whitepaper, [Online]. Available: http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/ 0/ SimplePairing_WP_V10r00.pdf, last access October 2010.
16. Hyppönen, K. and K.M.J. Haataja, 2007. Niño man-in-the-middle attack on bluetooth secure simple pairing. In: *3rd IEEE/IFIP International Conference in Central Asia on Internet, The Next Generation of Mobile, Wireless and Optical Communications Networks (ICI), ICI 2007*, pp: 1-5, 26-28. September 2007, Tashkent, Uzbekistan.
17. Haataja, K.M.J. and P. Toivanen, 2010. Two Practical Man-In-The-Middle Attacks on Bluetooth Secure Simple Pairing and Countermeasures. IEEE Transactions on Wireless Communications, 9(1): 384-392.
18. Sharmila, D. R. Neelaveni and K. Kiruba, 2009. Bluetooth Man-In-The-Middle attack based on Secure Simple Pairing using Out Of Band association model. In: Proceedings of *International Conference on* Control, Automation, Communication and Energy Conservation (INCACEC 2009), pp: 1-6, 4-6 June 2009.
19. Alam, Md.A. and M.I. Khan, 2010. Security Enhancement of Pairing and Authentication Process of Bluetooth, IJCSNS International J. Computer Science and Network Security, 10(6).