Two Practical Man-In-The-Middle Attacks on Bluetooth Secure Simple Pairing and Countermeasures

Keijo Haataja and Pekka Toivanen

***Abstract*—We propose two new Man-In-The-Middle (MITM) attacks on Bluetooth Secure Simple Pairing (SSP). The attacks are based on the falsification of information sent during the input/output capabilities exchange and also the fact that the security of the protocol is likely to be limited by the capabilities of the least powerful or the least secure device type. In addition, we devise countermeasures that render the attacks impractical, as well as improvements to the existing Bluetooth SSP in order to make it more secure. Moreover, we provide a comparative analysis of the existing MITM attacks on Bluetooth.**

***Index Terms*—Bluetooth, man-in-the-middle attack, out-of-band channel, secure simple pairing, wireless security.**

I. Introduction

THE use of wireless communication systems and their interconnections via networks have grown rapidly in recent years. Because radio frequency (RF) waves can penetrate obstacles, wireless devices can communicate with no direct line-of-sight between them. This makes RF communication easier to use than wired or infrared communication, but it also makes eavesdropping easier. Moreover, it is easier to disrupt and jam wireless RF communication than wired communication. Because wireless RF communication can suffer from these threats, additional countermeasures are needed to protect against them.

Bluetooth [1] is a technology for short range wireless data and realtime two-way audio/video transfer providing data rates up to 24 Mb/s. It operates at 2.4 GHz frequency in the free Industrial, Scientific, and Medical (ISM) band. Bluetooth devices that communicate with each other form a *piconet*. The device that initiates a connection is the piconet *master* and all other devices within that piconet are *slaves*.

Many kinds of Bluetooth devices, such as laptops, PCs, mice, keyboards, printers, mobile phones, headsets and hands-free devices, are widely used all over the world. In many countries, a hands-free device or headset connected to a mobile phone is mandatory in moving vechiles for safety reasons. Therefore, the markets for easy-to-use wireless Bluetooth headsets and hands-free devices are huge!

Already in 2006, the one billionth Bluetooth device was shipped [2], and the volume is expected to increase rapidly in the near future. According to the Bluetooth Special Interest Group (SIG), the target volume for 2010 is as high as two billion Bluetooth devices [3]. Therefore, it is very important to keep Bluetooth security issues up-to-date.

The results of this paper: In this paper, we propose two new MITM attacks on Bluetooth SSP. In addition, we devise countermeasures that render the attacks impractical as well as improvements to the existing Bluetooth SSP in order to make it more secure. Moreover, we provide a comparative analysis of the existing MITM attacks on Bluetooth including our attacks described in this paper.

The rest of the paper is organized as follows. Section II provides an overview of Bluetooth security. Our practical MITM attacks against Bluetooth SSP are proposed in Sect. III. Countermeasures for these attacks and improvements to the existing Bluetooth SSP are devised in Sect. IV. Section V provides a comparative analysis of the existing MITM attacks on Bluetooth. Finally, Sect. VI concludes the paper and sketches future work.

II. Overview of Bluetooth Security

The basic Bluetooth security configuration is done by the user who decides how a Bluetooth device will implement its connectability and discoverability options. The different combinations of connectability and discoverability capabilities can be divided into three categories, or *security levels*:

1) *Silent*: The device will never accept any connections. It simply monitors Bluetooth traffic.
2) *Private*: The device cannot be discovered, i.e. it is a so-called *non-discoverable device*. Connections will be accepted only if the *Bluetooth Device Address* (BD_ADDR) is known to the prospective master. A 48-bit BD_ADDR is normally unique and refers globally to only one individual Bluetooth device.
3) *Public*: The device can be both discovered and connected to. It is therefore called a *discoverable device*.

Because Bluetooth is a wireless communication system, there is always a possibility that the transmission could be deliberately jammed or intercepted, or that false or modified information could be passed to the piconet devices.

Powerful directional antennas can be used to increase the scanning, eavesdropping and attacking range of almost any kind of Bluetooth attack considerably. One very good example of a long-distance attacking tool is the BlueSniper Rifle [4], [5]. It is a rifle stock with a powerful directional antenna

Manuscript received July 2, 2009; revised September 23, 2009; accepted November 4, 2009. The associate editor coordinating the review of this paper and approving it for publication was W. Liao.

The authors are with the Department of Computer Science, University of Kuopio, P.O. Box 1627, FI-70211 Kuopio, Finland (e-mail: {keijo.haataja, pekka.toivanen}@uku.fi).

Digital Object Identifier 10.1109/TWC.2010.01.090935

attached to a small Bluetooth compatible computer. Scanning, eavesdropping and attacking can be done over a mile away from the target devices. Moreover, anyone with some basic skills and a few hundred dollars can build her own BlueSniper Rifle. Therefore, the possibility that an attacker is using range enhancement for improving the performance of the attacks should be taken seriously.

Nowadays it is possible to transform a standard $30 Bluetooth dongle into a full-blown Bluetooth sniffer [6]. We have also verified this fact in our research laboratory [7] with many different Cambridge Silicon Radio (CSR) based Bluetooth Universal Serial Bus (USB) dongles supporting Bluetooth versions up to 2.0+EDR (Enhanced Data Rate). In addition, tools for reverse engineering the firmware of CSR-based Bluetooth dongles are available [8]. The tools include a disassembler for the official firmware, and an assembler that can be used for writing custom firmware. With these tools anyone can now write custom firmware for CSR-based Bluetooth dongles to include raw access for Bluetooth sniffing. The tools also include the source code for sniffing Bluetooth under Linux. Moreover, it is expected that in the near future techniques for finding hidden (non-discoverable) Bluetooth devices in an average of one minute will be ported onto a standard CSR dongle via a custom firmware [7], [9], [10]. This will open new doors for practical Bluetooth security research and it will also provide a cheap basic weapon to all attackers for Bluetooth sniffing. It is expected that Bluetooth sniffing will soon become a very popular sport among attackers and hackers, thus making Bluetooth security concerns even more alarming.

Bluetooth security is based on building a chain of events, none of which should provide meaningful information to an eavesdropper. All events must occur in a specific sequence for security to be set up successfully.

In order for two Bluetooth devices to start communicating, procedure called *pairing* must be performed. As a result of pairing, two devices form a trusted pair and establish a link key which is used later on for creating a data encryption key for each session. In Bluetooth versions up to 2.0+EDR, pairing is based exclusively on the fact that both devices share the same *Personal Identification Number* (PIN) or passkey. When the user enters the same passkey in both devices, the devices generate the same shared secret which is used for authentication and encryption of traffic exchanged by them.

The PIN is the only source of entropy for the shared secret. As the PINs often contain only four decimal digits, the strength of the resulting keys is not enough for protection against passive eavesdropping on communication. Even with longer 16-character alphanumeric PINs, full protection against active eavesdropping cannot be achieved: it has been shown that MITM attacks on Bluetooth communications (versions up to 2.0+EDR) can be performed [7], [11], [12], [13].

Bluetooth versions 2.1+EDR and 3.0+HS (High Speed) add a new specification for the pairing procedure, namely SSP [1]. Its main goal is to improve the security of pairing by providing protection against passive eavesdropping and MITM attacks.

Instead of using (often short) passkeys as the only source of entropy for building the link keys, SSP employs Elliptic Curve Diffie-Hellman public-key cryptography. To construct the link key, devices use public-private key pairs, a number of nonces, and Bluetooth addresses of the devices. Passive eavesdropping is effectively thwarted by SSP, as running an exhaustive search on a private key with approximately 95 bits of entropy is currently considered to be infeasible in short time.

In order to provide protection against MITM attacks, SSP either uses an Out-Of-Band (OOB) channel (e.g., Near Field Communication, NFC), or asks for the user's help: for example, when both devices have displays and keyboards, the user is asked to compare two six-digit numbers. Such a comparison can be also thought as an OOB channel which is not controlled by the MITM. If the values used in the pairing process have been tampered with by the MITM, the six-digit integrity checksums will differ with the probability of 0.999999.

SSP uses four *association models*. In addition to the two association models mentioned previously, *OOB* and *Numeric Comparison* (NC), models named *Passkey Entry* (PE) and *Just Works* (JW) are defined. The PE association model is used in the cases when one device has input capability, but no screen that can display six digits. A six-digit checksum is shown to the user on the device that has output capability, and the user is asked to enter it on the device with input capability. The PE association model is also used if both devices have input, but no output capabilities. In this case the user chooses a 6-digit checksum and enters it in both devices. Finally, if at least one of the devices has neither input nor output capability, and an OOB cannot be used, the JW association model is used. In this model the user is not asked to perform any operations on numbers; instead, the device may simply ask the user to accept the connection.

SSP is comprised of six phases:

1) *Capabilities exchange*: The devices that have never met before or want to perform re-pairing for some reason, first exchange their Input/Output (IO) capabilities to determine the proper association model to be used.
2) *Public key exchange*: The devices generate their public-private key pairs and send the public keys to each other. They also compute the Diffie-Hellman key.
3) *Authentication stage 1*: The protocol that is run at this stage depends on the association model. One of the goals of this stage is to ensure that there is no MITM in the communication between the devices. This is achieved by using a series of nonces, commitments to the nonces, and a final check of integrity checksums performed either through the OOB channel or with the help of user.
4) *Authentication stage 2*: The devices complete the exchange of values (public keys and nonces) and verify the integrity of them.
5) *Link key calculation*: The parties compute the link key using their Bluetooth addresses, the previously exchanged values and the Diffie-Hellman key constructed in phase 2.
6) *LMP authentication and encryption*: Encryption keys are generated in this phase, which is the same as the final steps of pairing in Bluetooth versions up to 2.0+EDR.

The contents of messages sent during the SSP phase are outlined in Fig. 1. The used notations are also explained in

the figure.

Even though SSP improves the security of Bluetooth pairing, it has been shown that MITM attacks against Bluetooth 2.1+EDR and 3.0+HS devices are also possible by forcing victim devices to use the JW association model [7], [14], [15], [16], [17].

III. Novel MITM Attacks

In this section, we propose two new MITM attacks on Bluetooth SSP. The first attack is based on the falsification of information sent during the IO capabilities exchange (see Sect. III-A). The second attack requires some kind of visual contact to the victim devices in order to mislead the user to select a less secure option instead of using a more secure OOB channel (see Sect. III-B).

A. BT-Niño-MITM attack

We call our first new attack as *BT-Niño-MITM attack*, or *Bluetooth - No Input, No Output - Man-In-The-Middle attack* [7], [15]. In the attack we exploit the fact that the devices must exchange the information about their IO capabilities during the first phase of the SSP. The exchange is done over an unauthenticated channel, and an attacker that controls this channel can therefore modify the information about capabilities and force the devices to use the association model of her choice. In our attack, the devices are forced to use the JW association model, which does not provide protection against the MITM attack.

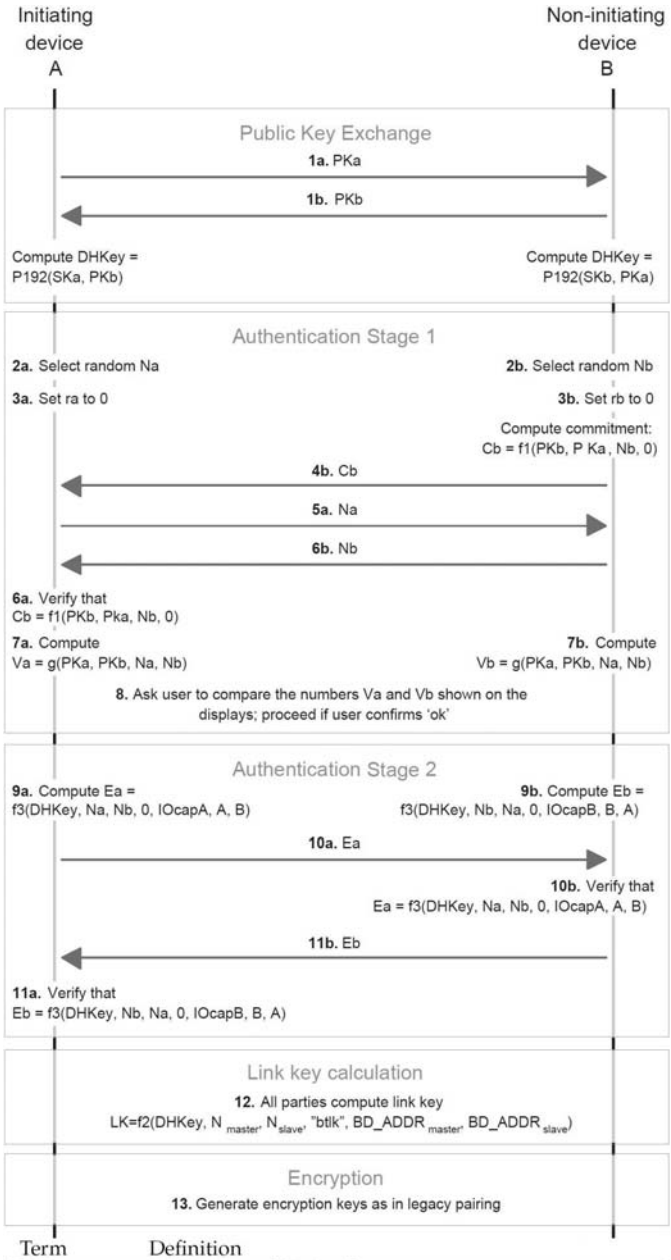
The MITM uses two separate Bluetooth devices with adjustable BD_ADDRs for the attack. Such devices are readily available on the market. The MITM clones the BD_ADDRs and user-friendly names (1–248 bytes long user-defined strings describing the Bluetooth devices) of the victim devices, in order to impersonate them more plausibly.

The main idea of the attack is depicted in Fig. 2. We next describe three scenarios for the attack.

In the first scenario, the MITM first disrupts (jams) the physical layer (PHY) by hopping along with the victim devices and sending random data in every timeslot. Another possibility is to jam the entire 2.4 GHz band altogether by using a wideband signal. In this way, the MITM shuts down all piconets within the range of susceptibility and there is no need to use a Bluetooth chipset to generate hopping patterns. Finally, a frustrated user thinks that something is wrong with her Bluetooth devices and deletes previously stored link keys. After that the user initiates a new pairing process by using SSP, and the MITM can forge messages exchanged during the IO capabilities exchange phase. When the JW association model has been forced into use, the attack continues as illustrated in Fig. 3.

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

Other two scenarios, where victim devices have never met before, are easier for the MITM, because in those cases the first phase of the attack (disrupting the PHY) can be skipped:



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 NC 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

Fig. 1. SSP with the NC association model.

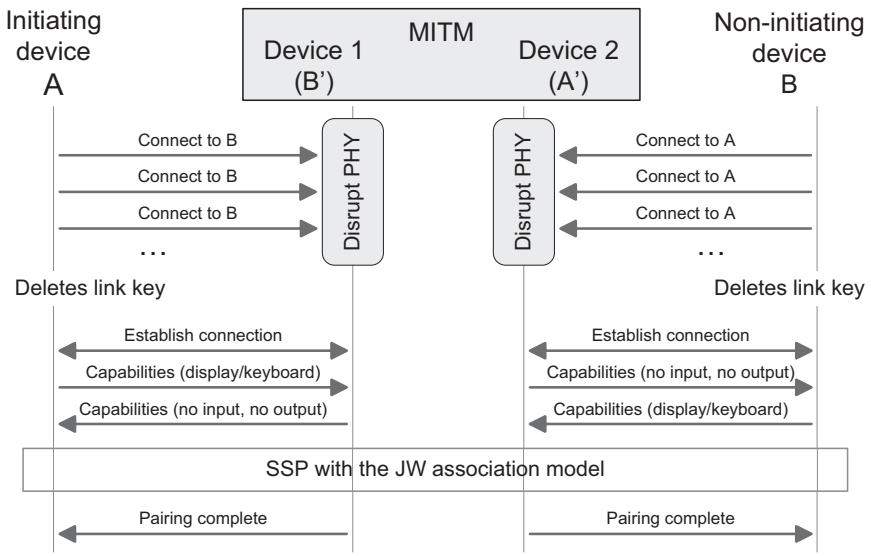


Fig. 2. Main idea of BT-Niño-MITM attack.

1) *The victim device (A or B) initiates SSP*: In this scenario, the MITM waits until A or B initiates SSP. After that, the attack proceeds as illustrated in Fig. 2 and 3.
2) *The MITM (A' and B') initiates SSP*: In this scenario, the MITM first initiates SSP with the victim devices. After that, the attack proceeds as illustrated in Fig. 2 and 3. Depending on the implementation of the victim devices, it may be possible to perform SSP without asking the user to accept the connection.

Depending on the situation, the MITM can use any of our three described attack scenarios. The applicability of a certain attack scenario obviously depends on the implementation of victim devices.

After a successful BT-Niño-MITM attack, the MITM can intercept and modify all data exchanged between the victim devices, and even use certain services that victim devices offer.

Suomalainen *et al.* have performed a comparative analysis of Bluetooth SSP, Wi-Fi Protected Setup, Wireles USB Association Models, and HomePlugAV security modes [14]. They present an attack against SSP similar to our BT-Niño-MITM attack. In their attack, the MITM prompts one device to use the normal NC association model, while forcing the other device to use the insecure JW association model. This leads to the fact that one of the devices (the one which uses the NC association model) treats the resulting link key as authenticated, and might choose to trust it even for an application which requires a high level of security. However, this attack looks somewhat suspicious from the point of view of the user: One of the devices asks the user to compare the integrity checksums, while the other device does not display any numbers. In the tests performed by Suomalainen *et al.*, only 6 users out of 40 accepted the pairing on both devices. Compared with this attack, our BT-Niño-MITM attack looks less dubious: Indeed, the user is only asked to confirm the pairing on both devices by pressing a button. In addition, according to the specification, even this confirmation request is optional, meaning that some of the manufacturers might choose to skip it, to improve usability. Moreover, as the MITM in our attack uses two Bluetooth devices with BD_ADDRs and Bluetooth names equal to those of the victim devices, the user gets even more confident that the pairing is proceeding correctly and securely. It is also worth noting that by using two MITM devices, SSP can be performed at the same time with both victim devices and it also ends at the same time with both victim devices, thus making the user even more confident.

Because it is difficult to combine high levels of security with good usability, other research of SSP has mostly concentrated on analyzing and improving its usability. Uzun *et al.* have analyzed different ways of prompting the user to perform the comparison of integrity checksums or to enter passkeys [18]. They have also provided guidelines for designing the user interface, to decrease the number of fatal errors, and thus improve both usability and security of SSP.

B. BT-SSP-OOB-MITM attack

We call our second attack as *BT-SSP-OOB-MITM attack*, or *Bluetooth - Secure Simple Pairing - Out-Of-Band - Man-In-The-Middle attack* [7], [17]. The attack requires that an attacker can somehow see the victim devices, i.e. there must be some kind of visual contact (for example, a hidden video camera or direct line-of-sight) to the victim devices. In the attack, legitimate users are misled to select a less secure option instead of using a more secure OOB channel (e.g., USB cable, infrared or NFC). The attack works against any two OOB-capable Bluetooth devices that support SSP: it will also work in the cases when devices use a cable for the OOB connection. The main idea of the attack is depicted in Fig. 4. We next describe a general scenario for the attack.

Our attack works in the following way:

1) *The MITM impersonates the legitimate devices*: The MITM uses two Bluetooth devices with BD_ADDRs and Bluetooth names equal to those of the victim devices, i.e. the user gets confident that the pairing is proceeding correctly and securely.

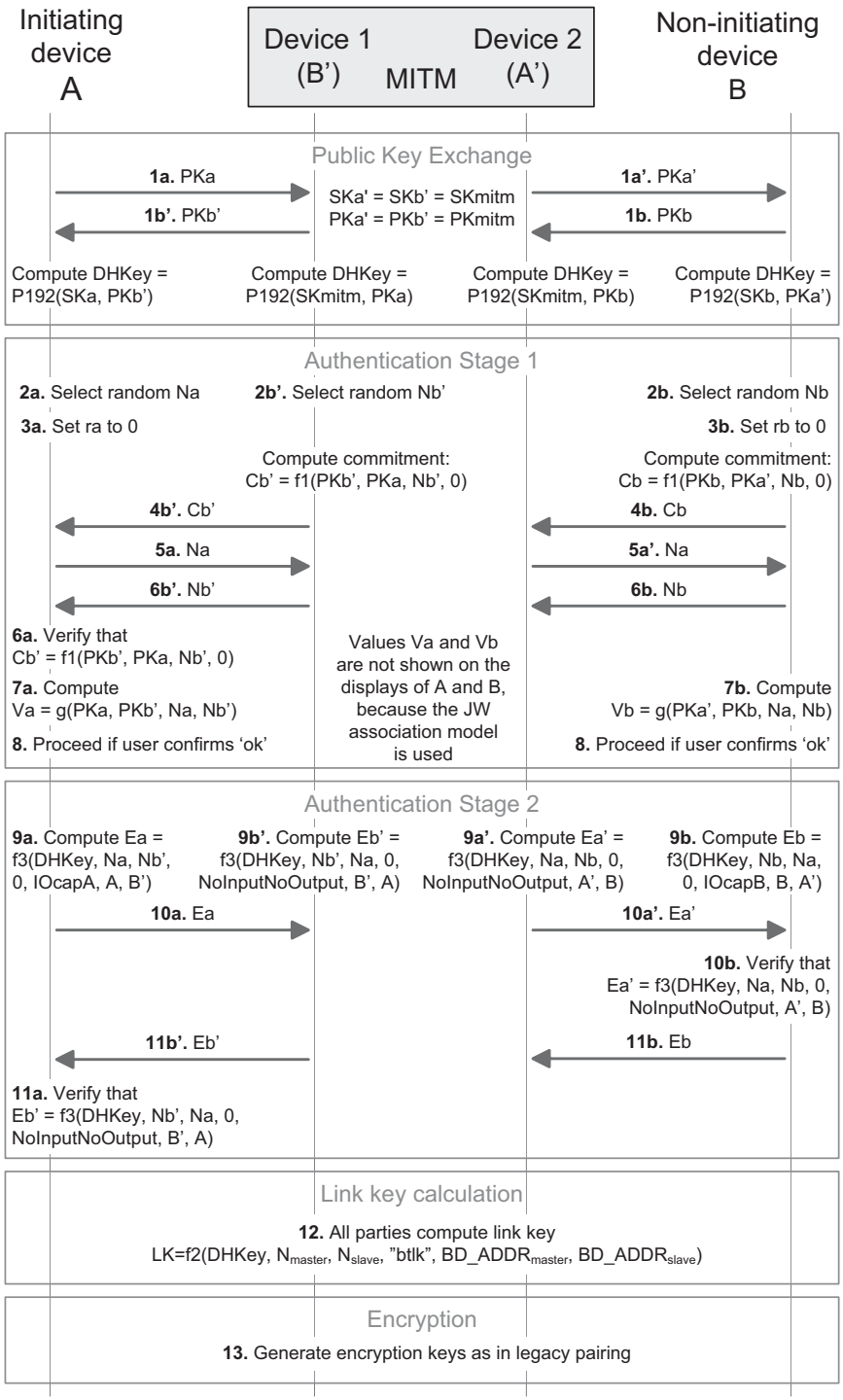


Fig. 3. Pairing details of our attacks.

2) *The MITM acts just before the legitimate user*: The attacker has visual contact to the victim devices and she notices that the user is about to start the SSP with the OOB association model. The MITM acts just before the legitimate user and establishes connections to both victim devices in order to start the capabilities exchange.
3) *The MITM forces the JW association model into use*: The MITM forges messages exchanged during the IO capabilities exchange phase and therefore forces the victim devices to use the JW association model. When the JW association model has been forced into use, the attack continues as illustrated in Fig. 3.

After a successful BT-SSP-OOB-MITM attack, the MITM can intercept and modify all data exchanged between the victim devices, and even use certain services that victim devices offer.

IV. Countermeasures and SSP Improvements

As a result of the work of Bluetooth SIG, SSP has gone through a series of reviews by experts; the released version

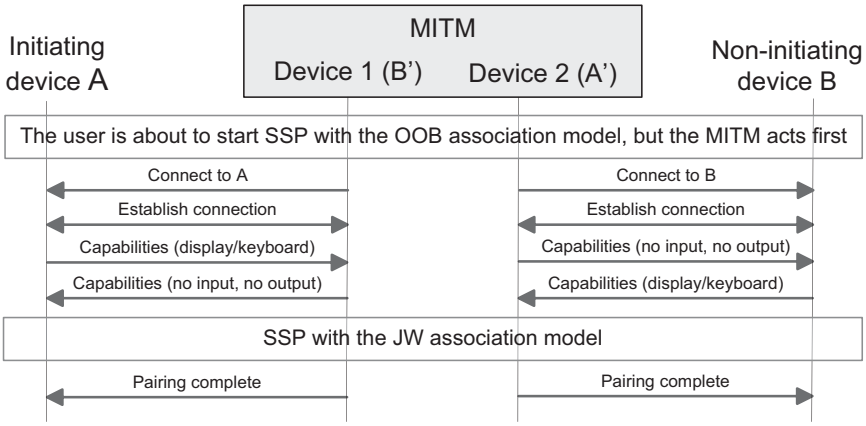


Fig. 4. Main idea of BT-SSP-OOB-MITM attack.

does altogether a good work in improving the security of Bluetooth pairing.

Our recommendations to improve the security of SSP are the following:

1) *An additional window at the user interface level*: We recommend that an additional window, "*The second device has no display and keyboard! Is this true?*", should be displayed at the user interface level of SSP when the JW association model is to be used. The user is asked to choose either "*Proceed*" or "*STOP*". In practice, future Bluetooth specifications should strongly recommend Bluetooth device/software manufacturers to implement this new window as a security improvement of SSP. The advantage of this approach is that the JW association model can still be a part of the future Bluetooth SSP specifications without any changes.
2) *The JW as an optional (not mandatory) association model*: Such devices that cannot use the new window at the user interface level or alternatively NFC as an OOB channel (better way), should implement their security either in the same way as old Bluetooth devices (versions up to 2.0+EDR) do or not to use Bluetooth security at all (if no sensitive data is exchanged). In this way, the implementation of the JW association model can be made optional and perhaps even removed altogether from the Bluetooth SSP specification. The one advantage of this approach is to eliminate all MITM attacks against the JW association model. Moreover, if the JW association model is not supported anymore in the future Bluetooth devices, it is not possible to force victim devices to use it.
3) *OOB as a mandatory association model*: Future Bluetooth specifications should make OOB a mandatory association model in order to radically improve the security and usability of SSP. However, it is likely that such a radical change in the specification will not be possible at once. Therefore, future Bluetooth specifications should at least strongly recommend the use of an OOB channel (e.g., NFC) to all Bluetooth device manufacturers.
4) *Using SSP's OOB channel only in a secure environment*: SSP's OOB channel should only be used in such a location where the attacker cannot have any kind of visual contact to the victim devices.
5) *Using RF fingerprints*: Even if devices are produced by the same manufacturer using the same components, there are differences between signals sent by these RF-devices. These small differences are the result of variations in the electronical components of a device. Thus, the RF-devices can be identified and differentiated from each other. Variances are most evident when the device is being activated or when it tries to access to the network, because then there exists a short transient phase in the signal. This transient phase lasts only $2-10$ ms. In the transient phase, there often occurs significant changes in frequency, amplitude and phase: the RF fingerprint (also referred to as RF signature) [7], [19], [20], [21], [22], [23], [24] can be formed from this particular part of the signal. Since every transmitter has a unique RF fingerprint, it can be used to differentiate the legitimate devices from devices that have alien RF fingerprints. For this purpose, a sample RF fingerprint is needed from each legitimate device in order to detect alien RF fingerprints. Wireless devices, such as Bluetooth devices, can be equipped with signal processing capabilities to check every RF fingerprint before accepting any connections. We feel that RF fingerprints could be in the major role for improving the security of Bluetooth, because RF fingerprints are extremely hard to duplicate (clone). As far as we know, nobody has ever performed a successful RF fingerprint duplication.

Also other countermeasures may be used, and most of them can be efficient against other Bluetooth security attacks as well. Such countermeasures include the use of the private or silent security level, increasing user awareness of security issues, minimization of transmit powers, and using additional security at the application level. Moreover, prior to an access to a sensitive information or services, a Bluetooth-independent re-authentication should be required.

V. Comparative Analysis of MITM Attacks

The first MITM attack on Bluetooth was devised by Jakobsson and Wetzel [11] for the version 1.0B of the standard. However, it works with all Bluetooth versions up to 2.0+EDR,

because no major security improvements were implemented in those Bluetooth specifications. The attack assumes that the link key used by two victim devices is known to the attacker. The authors also showed how to obtain the link key using offline PIN crunching, by passive eavesdropping on the initialization key establishment protocol. The MITM attack requires that both devices are in public or private security level (see Sect. II), i.e. both victim devices must be connectable. In the attack, the BD_ADDRs of the attacker's devices must be cloned to equal the addresses of the victim devices. Moreover, to prevent the jamming of the communication channel, the victim devices must be both masters or both slaves (in two different piconets). In this case they transmit in unsynchronized manner and cannot see the messages of each other, while communicating with the attacker. After establishing connection to both victims, the attacker sets up two new link keys.

Kügler [12] further improves the attack of Jakobsson and Wetzel. By manipulating with the clock settings, the attacker forces both victim devices to use the same channel hopping sequence but different clocks. In this way, the victim devices are unsynchronized, and can see only the messages the attacker sends them.

In addition, Kügler shows how a MITM attack can be performed during the paging procedure. The attacker responds to the page request of the master victim faster than the slave victim, and restarts the paging procedure with the slave using a different clock. The master and slave use the same channel hopping sequence, but a different offset in this sequence. The attack works also in case when both victim devices send and receive data packets over an encrypted link. Even though the Initialization Vector (IV) used for encryption depends on the clock, the last bit of the clock is unused. Therefore, the attacker can flip this last bit, forcing the victims to use clocks which have the difference of approximately 11.65 hours. Although the integrity of data is protected with Cyclic Redundancy Checks (CRCs) which are appended to the plaintext prior to encryption, the attacker can manipulate intercepted ciphertext. After modifying the ciphertext in a certain way, the attacker updates the CRC bits (see [25] for details); the integrity checks performed by the victims do not detect the modification. It must be noted, however, that the attacker does not have much time for manipulating the transmitted data.

Reflection attacks [13] (also referred to as *relay attacks*) aim at impersonating the victim devices. The attacker does not need to know any secret information, because she only relays (reflects) the received information from one victim device to another during the authentication. The reflection attack can be seen as a type of a MITM attack against authentication, but not encryption. The only information needed is the BD_ADDRs of the victim devices.

The reflection attack can be *one-sided*, in which only one victim device is impersonated, and *two-sided*, whereby both victim devices are impersonated. The attacker must use two Bluetooth devices with adjustable BD_ADDRs (for example, protocol analyzers). In addition, the attacker must be capable of relaying the received information between her devices, because victim devices can be far away from each other. During the paging procedure, the attacker responds to the request of the first victim device (A), and initiates a connection to the second victim device (B), posing as A. If the victim devices can hear each other, the mechanisms described in [12] can be used to achieve this. After this, the attacks work on the Link Manager Protocol (LMP) layer of Bluetooth. The messages of the protocol are simply relayed by the attacker's devices. In case of the one-sided attack, only a part of messages must be relayed, and connection to A is dropped when the attacker has impersonated it to B. The attacker can successfully perform authentication by using reflection attacks, but she cannot continue the attack if the target devices encrypt their communication. By combining reflection attacks with a known secret PIN code, link key or encryption key, the attacker can both impersonate the victim devices and decrypt the information transferred between them.

Victim devices can detect the attack by noticing a considerable increase in latency of getting the response to authentication challenge, caused by relaying. This countermeasure is not described in the standards, and it is up to the discretion of manufacturers to provide it.

The versions 2.1+EDR and 3.0+HS of Bluetooth provide protection against the MITM attacks described above, by the means of SSP described in Sect. II. However, it has been shown that MITM attacks against Bluetooth 2.1+EDR and 3.0+HS devices are also possible [7], [14], [15], [16], [17]. Because SSP supports several association models, selection of which depends on the capabilities of the target devices, the attacker can force the devices into the use of a less secure mode by changing the capabilities information. For example, by forcing the JW association model into use, the attacker can bypass all security checks which would normally be in place. The association is then unauthenticated; the devices are aware of this fact, but it depends on the manufacturer how they react to this. If the victim devices have already established a link key, the attacker can use jamming to disrupt the communication, and then initiate the connection under a chosen association model with both devices. As a result, the attacker learns the link key used by the devices and can intercept all data transmitted between the devices.

In Table I we summarize the properties of the MITM attacks overviewed in this section. It is interesting to note the connection of MITM attacks to other developments in the Bluetooth security analysis. For instance, at the time when most of the MITM attacks were introduced, implementing them was not an easy task, as there were no devices with adjustable BD_ADDRs, except sophisticated and expensive protocol analyzers. Now the situation has changed: Bluetooth devices with an adjustable BD_ADDR are readily available and techniques for finding hidden (non-discoverable) Bluetooth devices have been invented (see Sect. II). Therefore, the danger of MITM attacks has recently increased.

VI. Conclusion and Future Work

Two new MITM attacks on Bluetooth SSP were proposed. In addition, countermeasures that render the attacks impractical as well as improvements to the existing Bluetooth SSP in order to make it more secure were devised. Moreover, a comparative analysis of the existing MITM attacks on Bluetooth was provided.

TABLE I
MITM ATTACKS ON BLUETOOTH: SUMMARY AND COMPARISON

Attack properties	[11]	[12]	[13]	[14]	[15], [16], [17]
Bluetooth versions	1.0 – 2.0+EDR	1.0 – 2.0+EDR	1.0 – 2.0+EDR	2.1+EDR – 3.0+HS	2.1+EDR – 3.0+HS
Attack goals	Impersonation, modification	Impersonation, modification	Impersonation	Impersonation, modification	Impersonation, modification
Attacking devices	2	2	2	2+1; note that a jamming device is also required	2+1; note that a jamming device is also required
Devices attacked	Connectable	Connectable or non-connectable	Connectable or non-connectable	Connectable or non-connectable	Connectable or non-connectable
Distances	Any[1]	Any[1]	Any[1]; note that the victim devices must be out of each other's range	Any[1]	Any[1]
Detection	By user: entering PIN to renegotiate	The attack remains undetected	By devices: delays in getting the LMP authentication response	By user: one of the devices asks to compare numbers, the other one does not	By user: no Numeric Comparison is used although both devices have displays and keyboards
Main countermeasure	Policies protecting against MITM attacks	Cryptographic integrity checks of packets	Detecting the delays	At the user interface level	At the user interface level

[1] The attacker must use two Bluetooth adapters; actual distance is limited by speed of the link between the attacker's devices.

It is difficult to create a protocol which caters to all possible types of wireless devices, as the security of the protocol is likely to be limited by the capabilities of the least powerful or the least secure device type. Most Bluetooth MITM attacks are based exactly on this problem.

The MITM attacks against SSP can be prevented on the user interface level by using an additional window. However, this is a clear trade-off between security and usability.

In general, MITM attacks are hard to prevent in wireless networks. By far the best way to stop the attacks is to use SSP's OOB channel in such a way that an attacker cannot have visual contact to the victim devices. Moreover, the usability of the OOB channel is of great importance: If wires must be used for pairing wireless devices, one is likely to opt for less secure but more usable options. We concur with the designers of SSP on their suggestion to use NFC as the OOB channel.

The problems we want to investigate in our future research work are concerned with the following issues:

1) Bluetooth is a relatively new wireless technology and therefore new attacks against Bluetooth security are likely to be found. We want to further investigate Bluetooth security weaknesses and propose countermeasures against new attacks.
2) Issues related to Bluetooth user experience (ease of use) have become more and more important in recent years. Therefore, we want to investigate how enhanced user experience will affect Bluetooth security in various scenarios, including social aspects and user acceptance/habits in security management. Moreover, we want to devise best practices depending on the risk analysis within each scenario.
3) Since we have already proposed a new efficient Intrusion Detection and Prevention System for Bluetooth networks [26], we want to implement a working prototype of such a system and also analyse its efficiency.
4) We want to extend our Bluetooth intrusion detection/prevention related research to cover also RF-fingerprinting techiques [7], [19], [20], [21], [22], [23], [24], because we feel that the use of RF fingerprints could be the future of secure Bluetooth communications.
5) Because cheap Bluetooth devices with an adjustable BD_ADDR are readily available, tools for modifying official firmwares have been released, techniques for finding hidden Bluetooth devices in an average of one minute have been invented, and an open-source Bluetooth sniffer for Linux environments has been released (see Sect. II), we want to continue our practical Bluetooth security research under Linux using these new tools [7]. We also want to further develop the existing open-source Bluetooth sniffer to include the BD_ADDR duplication feature and the graphical user interface in order to make it user-friendly.
6) Since it is nowadays possible to acquire the hardware required for MITM attacks, we want to make practical implementations of all existing Bluetooth MITM attacks. Moreover, we want to analyse the results of the practical experiments, draw conclusions, and propose practical countermeasures based on our findings.
7) Since there are many new emerging wireless technologies, such as ZigBee [27] and Ultra-Wideband (UWB), which are quite similar to Bluetooth technology, it is expected that our Bluetooth security related research work can be quite easily extended to cover the security of these new technologies. Therefore, we want to investigate how various Bluetooth security attacks and their countermeasures can be ported to support ZigBee and UWB technologies.

Bluetooth security intimately depends on general problems of ad-hoc network security, on physical aspects of protecting wireless networks, on cryptographical solutions to key distribution without Trusted Third Party or Certification Authority (CA) infrastructure, and on application layers. The research in this area combines various skills and techniques, requires cooperation with other researchers, and also requires a certain infrastructure.

REFERENCES

[1] Bluetooth SIG, Bluetooth Specifications 1.0–3.0+HS. [Online]. Available: http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications. [Accessed Sep. 17, 2009].

[2] Bluetooth SIG, Bluetooth Wireless Technology Surpasses One Billion Devices. [Online]. Available: http://www.bluetooth.com/Bluetooth/Press/SIG/BLUETOOTH_WIRELESS_TECHNOLOGY_SURPASSES_ONE_BILLION_DEVICES.htm. [Accessed Sep. 17, 2009].

[3] Bluetooth SIG, 2008 Marks Ten Years of Bluetooth Wireless Technology. [Online]. Available: http://www.bluetooth.com/Bluetooth/Press/SIG/2008_MARKS_TEN_YEARS_OF_emBLUETOOTHem_WIRELESS_TECHNOLOGY.htm. [Accessed Sep. 17, 2009].

[4] H. Cheung, "How to: building a BlueSniper Rifle–part 1." [Online]. Available: http://www.smallnetbuilder.com/content/view/24256/98. [Accessed Sep. 17, 2009].

[5] H. Cheung, "How to: building a BlueSniper Rifle–part 2." [Online]. Available: http://www.smallnetbuilder.com/content/view/24228/98. [Accessed Sep. 17, 2009].

[6] M. Moser, "Busting the Bluetooth myth–getting RAW access." [Online]. Available: http://www.packetstormsecurity.org/papers/wireless/busting_bluetooth_myth.pdf. [Accessed Sep. 17, 2009].

[7] K. Haataja, "Security threats and countermeasures in Bluetooth-enabled systems," Ph.D. dissertation, University of Kuopio, Department of Computer Science, Feb. 6, 2009.

[8] Darkircop, "CSR Sniffer–firmware assembler and disassembler." [Online]. Available: http://bluetooth-pentest.narod.ru/software/xap2.zip. [Accessed Sep. 17, 2009].

[9] D. Spill and A. Bittau, "BlueSniff–Eve meets Alice and Bluetooth," in *Proc. First USENIX Workshop on Offensive Technologies (WOOT'2007)*, Boston, MA, Aug. 2007.

[10] D. Spill and A. Bittau, BlueSniff source codes. [Online]. Available: http://www.cs.ucl.ac.uk/staff/a.bittau/gr-bluetooth.tar.gz. [Accessed Sep. 17, 2009].

[11] M. Jakobsson and S. Wetzel, "Security weaknesses in Bluetooth," *Lecture Notes in Computer Science*, vol. 2020, pp. 176–191, Springer-Verlag, 2001.

[12] D. Kügler, "Man in the middle attacks on Bluetooth," *Lecture Notes in Computer Science*, vol. 2742, pp. 149–161, Springer-Verlag, 2003.

[13] A. Levi, E. Cetintas, M. Aydos, C. Koc, and M. Caglayan, "Relay attacks on Bluetooth authentication and solutions," *Lecture Notes in Computer Science*, vol. 3280, pp. 278–288, Springer-Verlag, 2004.

[14] J. Suomalainen, J. Valkonen, and N. Asokan, "Security associations in personal networks: a comparative analysis," *Lecture Notes in Computer Science*, vol. 4572, pp. 43–57, Springer-Verlag, 2007.

[15] K. Hyppönen and K. Haataja, " "Niño" man-in-the-middle attack on Bluetooth secure simple pairing," in *Proc. IEEE Third International Conference in Central Asia on Internet, The Next Generation of Mobile, Wireless and Optical Communications Networks (ICI'2007)*, Tashkent, Uzbekistan, Sep. 2007.

[16] K. Haataja and K. Hyppönen, "Man-in-the-middle attacks on Bluetooth: a comparative analysis, a novel attack, and countermeasures," in *Proc. IEEE Third International Symposium on Communications, Control and Signal Processing (ISCCSP'2008)*, St. Julians, Malta, Mar. 2008.

[17] K. Haataja and P. Toivanen, "Practical man-in-the-middle attacks against Bluetooth secure simple pairing," in *Proc. 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'2008)*, Dalian, China, Oct. 2008.

[18] E. Uzun, K. Karvonen, and N. Asokan, "Usability analysis of secure pairing methods," *Lecture Notes in Computer Science*, vol. 4886, pp. 307–324, Springer-Verlag, 2008.

[19] J. Shandle, "University research aims at more secure Wi-Fi." [Online]. Available: http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=192501255. [Accessed Sep. 17, 2009].

[20] M. Barbeau, J. Hall, and E. Kranakis, "Detecting impersonation attacks in future wireless and mobile networks," *Lecture Notes in Computer Science*, vol. 4074, pp. 80–95, Springer-Verlag, 2006.

[21] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," in *Proc. 15th USENIX Security Symposium*, Vancouver, B.C., Canada, July 2006.

[22] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," *Canadian J. Electrical and Computer Engineering*, vol. 32, no. 1, pp. 27–33, 2007.

[23] O. Ureten and N. Serinken, "Bayesian detection of radio transmitter turn-on transients," in *Proc. IEEE Non Linear Signal and Image Processing Conference*, Antalya, Turkey, June 1999, pp. 830–834.

[24] J. Hall, M. Barbeau, and E. Kranakis, "Detecting rogue devices in Bluetooth networks using radio frequency fingerprinting," in *Proc. IASTED International Conference on Communications and Computer Networks (CCN'2006)*, Lima, Peru, Oct. 2006.

[25] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity on 802.11," in *Proc. 7th Annual International Conference on Mobile Computing and Networking (MOBICOM'2001)*, ACM Press, 2001.

[26] K. Haataja, "New efficient intrusion detection and prevention system for Bluetooth networks," in *Proc. ACM International Conference on Mobile, Wireless MiddleWare, Operating Systems, and Applications (Mobilware'2008)*, Innsbruck, Austria, Feb. 2008.

[27] ZigBee Alliance, Technical specifications–ZigBee specifications. [Online]. Available: http://www.zigbee.org/Products/TechnicalDocumentsDownload/tabid/237/Default.aspx. [Accessed Sep. 17, 2009].

Keijo Haataja was born in Pielavesi, Finland, on April 25, 1978. He received the M.Sc. (Tech.) degree in data communications from the Lappeenranta University of Technology in 2001, Ph.Lic. degree in computer science from the University of Kuopio in 2007, and Ph.D. degree in computer science from the University of Kuopio in 2009. Since 2002, he has been a senior assistant of Wireless Communications and Data Security at the University of Kuopio, Finland. Since 2009, he has also been a project manager/coordinator of several Finnish and EU projects at the University of Kuopio, Finland. In addition, he is a member of Bluetooth Security Expert Group (SEG), which provides Bluetooth working groups and the Bluetooth SIG with expertise in a wide range of security issues. The goal of the Bluetooth SEG group is to identify threats to Bluetooth wireless security and to ensure that these threats are mitigated through Bluetooth specification enhancements, whitepapers, test cases, and test tools that identify security vulnerabilities in devices. Moreover, he has been a technical consultant of Unicta Oy since 2008. His main research interests include wireless communications, wireless security, mobile systems, sensor networks, data communications, and intelligent autonomous robots. His doctoral dissertation, "Security Threats and Countermeasures in Bluetooth-Enabled Systems," is the world's first practically oriented doctoral dissertation about Bluetooth security.

Pekka Toivanen graduated from Helsinki University of Technology in 1989. He worked at Nokia Electronics and Nokia Information Systems. From 1988 until 1991 he worked at the Technical Research Center of Finland as a researcher. From 1991 until 2005 he worked at the Lappeenranta University of Technology, in the Department of Information Technology. He received the Doctor of Technology degree in 1996. He worked as an acting associate professor from 1997 until 1999 and professor from 1999 until 2001 at the same university. From 2001 until 2005 he worked as a full professor. He was also the head of the Laboratory of Information Processing in 2001-2005. From 2007 onwards he has been working as a full professor at University of Kuopio. He is also the current Head of the Department of Computer Science at University of Kuopio. He is a member of SPIE and IEEE. He has published more than 200 reviewed research articles in international scientific conferences and journals. He has been a member of the organizing committees of conferences and is a member of several boards. His main research interests are computational intelligence, machine vision (especially multispectral image processing), and color and distance transforms.