Man-In-The-Middle Attacks on Bluetooth: a Comparative Analysis, a Novel Attack, and Countermeasures

Keijo M.J. Haataja, Konstantin Hyppönen
Department of Computer Science
University of Kuopio
P.O.Box 1627, FIN-70211 Kuopio, Finland
E-mail: {keijo.haataja, konstantin.hypponen}@cs.uku.fi

***Abstract*—We provide a comparative analysis of the existing MITM (Man-In-The-Middle) attacks on Bluetooth. In addition, we propose a novel Bluetooth MITM attack against Bluetooth-enabled printers that support SSP (Secure Simple Pairing). Our attack is based on 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. Moreover, we propose improvements to the existing Bluetooth SSP in order to make it more secure.**

***Index Terms*—Authentication, Bluetooth, Man-In-The-Middle attack, 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, disrupting and jamming of wireless RF communication is easier than that of wired communication. Because wireless RF communication can suffer from these new threats, additional countermeasures are needed to protect against them.

Bluetooth [1] is a technology for short range wireless data and realtime two-way voice transfer. It operates at 2.4 GHz frequency in the free ISM-band (Industrial, Scientific, and Medical) by using frequency hopping. 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.

Many kinds of Bluetooth devices, such as mobile phones, laptops, PCs, headsets, mice, keyboards and printers, are widely used all over the world. On November 14th 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 SIG (Special Interest Group), the target volume for 2010 is as high as two billions Bluetooth devices. Therefore, it is very important to keep Bluetooth security issues up-to-date.

Our results: In this paper, we provide a comparative analysis of the existing MITM attacks on Bluetooth. In addition, we propose a novel Bluetooth MITM attack against Bluetooth-enabled printers that support SSP. Moreover, we propose improvements to the existing Bluetooth SSP in order to make it more secure.

The rest of the paper is organized as follows. Section II provides an overview of Bluetooth security. A comparative analysis of the existing Bluetooth MITM attacks is provided in Sect. III. Section IV proposes a novel Bluetooth MITM attack against Bluetooth-enabled printers that support SSP. Improvements to the existing SSP are proposed in Sect. V. Finally, Sect. VI concludes the paper.

II. An 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 *BD_ADDR (Bluetooth Device Address)* of the device 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 [3], [4]. It is a rifle stock with a powerful directional antenna 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 his own BlueSniper

978-1-4244-1688-2/08/$25.00

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 also possible to transform a standard $30 Bluetooth dongle into a full-blown Bluetooth sniffer [5], [6]. We have also verified this fact in our research laboratory with many different CSR-based (Cambridge Silicon Radio) Bluetooth 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 [7]. 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 within a few minutes will be ported onto a standard CSR dongle via a custom firmware [8], [9]. 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 *PIN (Personal Identification Number)* code 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 [10], [11], [12].

Bluetooth version 2.1+EDR [1] adds a new specification for the pairing procedure, namely SSP. 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*), models named *Passkey Entry* and *Just Works* are defined. The Passkey Entry 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 Passkey Entry 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 Just Works 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.

The choice of the association model depending on the device capabilities is shown in Table I. DisplayYesNo indicates that the device has a display and at least two buttons that are mapped to "yes" and "no": using the buttons the user can either accept the connection or decline it. Other notation in the table is self-explanatory.

TABLE I
DEVICE CAPABILITIES AND SSP ASSOCIATION MODELS

Device 1	Device 2	Association Model
DisplayYesNo	DisplayYesNo	Numeric comparison *
	DisplayOnly	Numeric comparison
	KeyboardOnly	Passkey Entry *
	NoInputNoOutput	Just Works
DisplayOnly	DisplayOnly	Numeric comparison
	KeyboardOnly	Passkey Entry *
	NoInputNoOutput	Just Works
KeyboardOnly	KeyboardOnly	Passkey Entry *
	NoInputNoOutput	Just Works
NoInputNoOutput	NoInputNoOutput	Just Works

* The resulting link key is considered *authenticated*.

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 IO (Input/Output) capabilities (see Table I) 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, and used notations are explained in Table II.

TABLE II
PROTOCOL NOTATION

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

Even though SSP improves the security of Bluetooth pairing, it has been shown that MITM attacks against Bluetooth 2.1+EDR devices are also possible [13], [14].

III. A COMPARATIVE ANALYSIS OF MITM ATTACKS

The first MITM attack on Bluetooth was devised by Jakobsson and Wetzel [10] 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.

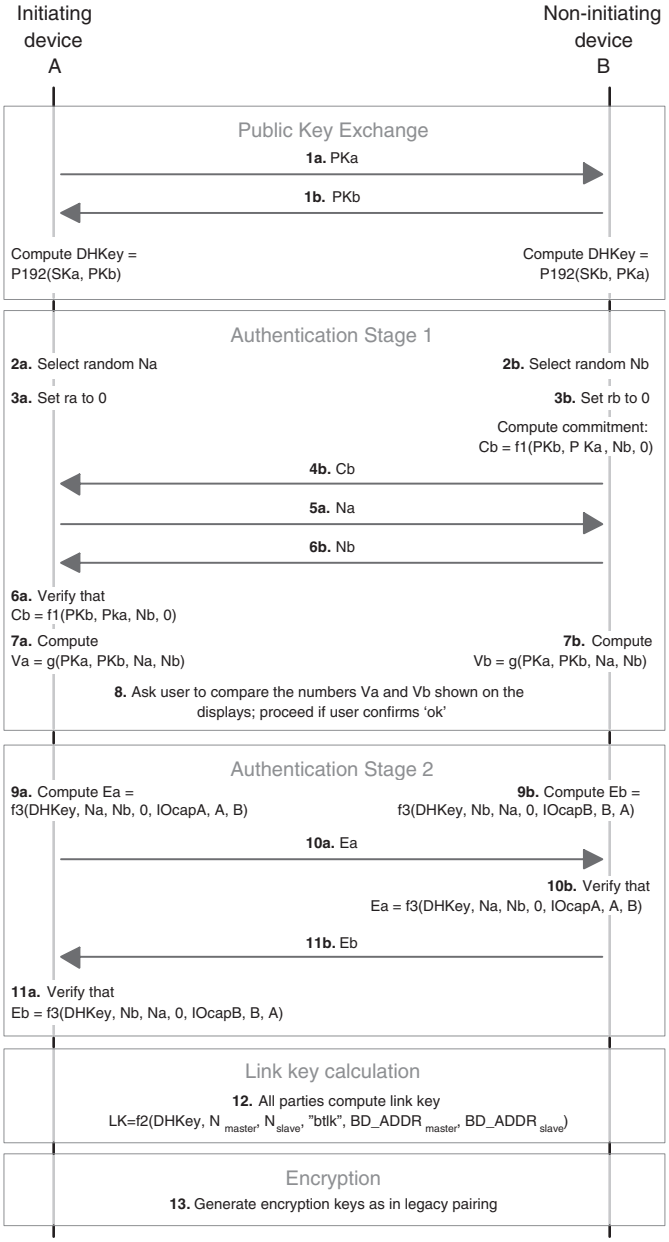

Fig. 1. SSP with Numeric Comparison association model

Kügler [11] 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 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 CRCs (Cyclic Redundancy Checks) 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 [15] 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 [12] (also referred to as *relay attacks*) aim at impersonating the victim devices. The attacker does not need to know any secret information, because he 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 his 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 [11] may be used to achieve this. After this, the attacks work on the LMP (Link Manager Protocol) 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 he 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 version 2.1+EDR of Bluetooth provides protection against the MITM attacks described above, by the means of SSP described in section II. However, it has been shown that MITM attacks against Bluetooth 2.1+EDR devices are also possible [13], [14]. 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 Just Works 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 III 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.

IV. A Novel Bluetooth MITM Attack

We call our new attack as *BT-SSP-Printer-MITM attack* (Bluetooth - Secure Simple Pairing - Printer - Man-In-The-Middle attack). In the attack we exploit the fact that almost all Bluetooth-enabled printers that support SSP (especially those connected using Bluetooth USB printer adapters) will use the Just Works association model in order to make printing user friendly. It is not likely that users will be required to press any printer buttons just to accept the connection establishment in the initial pairing process of SSP. Therefore, the Just Works association model seems to be the only logical choice for SSP-enabled printers. Our attack is possible because the Just Works association model does not provide any protection against MITM attacks.

It is also possible that some SSP-enabled printers that have displays and buttons could use the Numeric Comparison or the Passkey Entry association model, which provide protection against MITM attacks. However, victim devices can be forced to use any association model that the attacker chooses [13], [14]. Therefore, our attack works even against such SSP-enabled printers that should provide MITM protection.

The main idea of the attack is depicted in Fig. 2. In what follows, we describe two scenarios for the attack.

We assume in the first scenario that the victim devices are using the Just Works association model. Our attack works in the following way:

1) *The MITM disrupts the PHY*: The MITM disrupts (jams) the PHY (physical layer) 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 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 his Bluetooth devices and deletes previously stored link keys.
2) *The MITM impersonates the legitimate printer*: Since the user has deleted the previously stored link keys,

TABLE III
MITM ATTACKS ON BLUETOOTH: SUMMARY AND COMPARISON

Attack properties	[10]	[11]	[12]	[13]	[14]
Bluetooth versions	1.0 – 2.0+EDR	1.0 – 2.0+EDR	1.0 – 2.0+EDR	2.1+EDR	2.1+EDR
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 countermeasures	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.

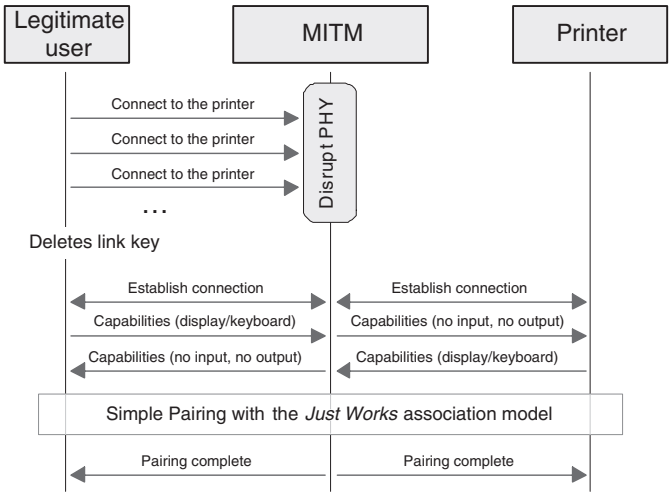

Fig. 2. Main idea of the attack

he will initiate a new pairing process through SSP. The SSP pairing details are illustrated in Fig. 3. It is worth noting that the user has deleted all information about the legitimate printer including its BD_ADDR, and therefore the MITM is not even required to clone the BD_ADDR of the legitimate printer in order to impersonate it. Now, the MITM only clones the user-friendly name (1–248 bytes long user-defined string describing the Bluetooth device) of the legitimate SSP-enabled printer to impersonate it. Moreover, the MITM must be able to disrupt the legitimate printer in such a way that it cannot communicate with other legitimate Bluetooth devices. Therefore, when the user searches available Bluetooth printers in the range, the only printer that is found will be the MITM with a different BD_ADDR but the same familiar user-friendly name. It is very likely that the user will not notice anything strange, because BD_ADDRs are much harder to remember than the user-friendly names. Therefore, the user will most likely choose the "MITM printer" that looks familiar to him. In this way, the MITM has replaced the legitimate printer in Bluetooth network by the "MITM printer" with different BD_ADDR. It is worth noting that by using a BD_ADDR different from that of the legitimate printer, the MITM can also eliminate possible BD_ADDR collisions that may occur, i.e. the attack works more reliably and plausibly.

3) *The MITM intercepts all data*: When the legitimate Bluetooth devices are printing via Bluetooth connection, the MITM captures (receives) all data and is also capable of decrypting it if encryption is used. Moreover, the MITM may even be capable of using certain services that these victim devices offer.
4) *The MITM relays the data to the legitimate printer*: Finally, the MITM relays the captured data to the legitimate printer. In this way, everything seems to work normally in the user's point of view: all documents are printed without any problems.

We assume in the second scenario that the victim devices are using the Numeric Comparison or the Passkey Entry association model. This attack works exactly the same way as our first attack scenario except that one additional phase is required: *the legitimate devices must be forced to use the Just Works association model* by using the attack scenarios described in [14].

Note that since our two attack scenarios are designed against Bluetooth 2.1+EDR (SSP-enabled) printers, a MITM device is required between the victim devices for the attacks to work. Attacks against Bluetooth 2.0+EDR and earlier printers are easier in practice, because the MITM device is not required. Such attack scenarios and their practical implementations are described in [16].

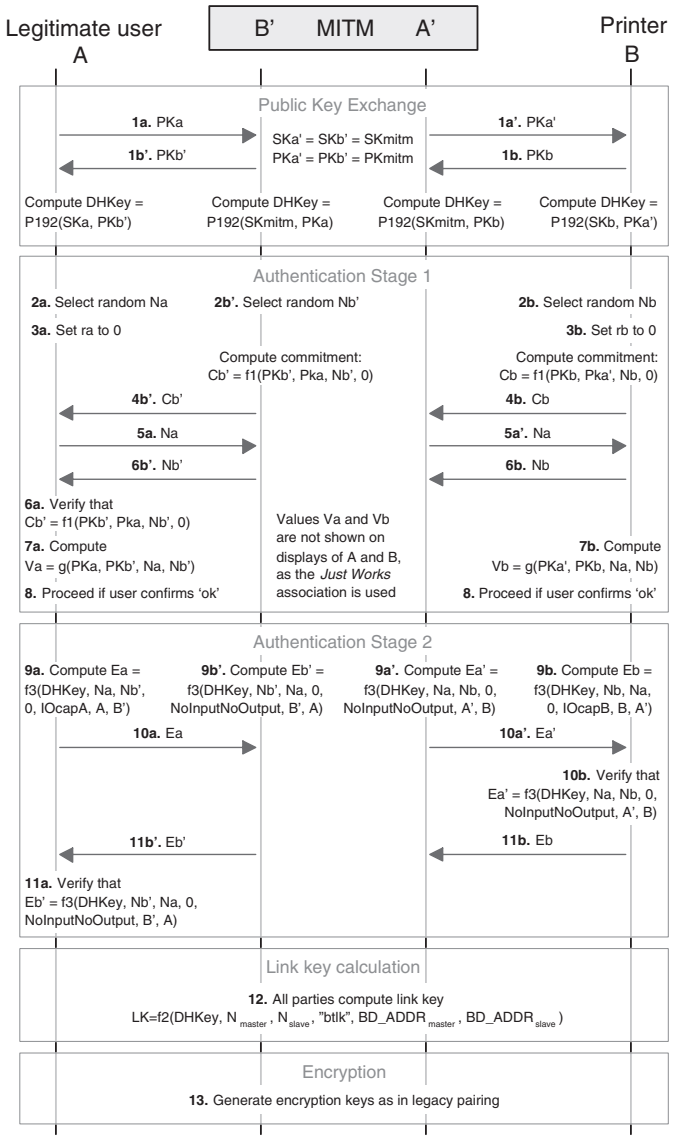

Fig. 3. SSP Pairing Details

V. Improvements to the existing SSP

As a result of the work of Bluetooth SIG, SSP has gone through a series of reviews by experts; the released version does altogether a good work in improving the security of Bluetooth pairing.

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 [17]. They have also provided guidelines for designing the user interface, to decrease the number of fatal errors and thus to improve both usability and security of SSP.

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 Just Works 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 Just Works association model can still be a part of the future Bluetooth SSP specifications without any changes.
2) *Just Works 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 Just Works 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 Just Works association model. Moreover, if the Just Works 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.

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, careful selection of place where sensitive information is exchanged, 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.

VI. Conclusions

A comparative analysis of the existing MITM attacks on Bluetooth was provided in the paper. In addition, a novel Bluetooth MITM attack against Bluetooth-enabled printers that support SSP was proposed. Moreover, improvements to the existing Bluetooth SSP was proposed.

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. Our Bluetooth MITM attack presented in this paper is based exactly on this problem.

In general, MITM attacks are hard to prevent in wireless networks. By far the best way to stop the attacks is to use an OOB channel, and SSP supports this option. However, 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.

REFERENCES

[1] Bluetooth SIG, *Bluetooth specifications 1.0 – 2.1+EDR*. Technical specifications, 1999–2007. http://www.bluetooth.com

[2] Bluetooth SIG, *Bluetooth Technology in Hands of One Billion*. Press release, Nov. 14, 2006. http://www.bluetooth.com/Bluetooth/SIG/Billion.htm

[3] H. Cheung, *How To: Building a BlueSniper Rifle – Part 1*. SmallNetBuilder, Pudai LLC, homepage, 2005. http://www.smallnetbuilder.com/content/view/24256/98

[4] H. Cheung, *How To: Building a BlueSniper Rifle – Part 2*. SmallNetBuilder, Pudai LLC, homepage, 2005. http://www.smallnetbuilder.com/content/view/24228/98

[5] Bluetooth security & Bluetooth hackers community blog, *Bluetooth Sniffing For Less*. Homepage, 2007. http://bluetoothsecurity.wordpress.com/2007/05/12/bluetooth-sniffing-for-less

[6] M. Moser, *Busting The Bluetooth Myth – Getting RAW Access*. Remote-exploit.org, Research Report, 2007. http://www.remote-exploit.org/research/busting_bluetooth_myth.pdf

[7] Darkircop, *CSR Sniffer – Firmware assembler and disassembler*. Homepage, 2007. http://darkircop.org/bt/bt.tgz

[8] D. Spill, A. Bittau, *BlueSniff – Eve meets Alice and Bluetooth*. Proceedings of the First USENIX Workshop on Offensive Technologies (WOOT'2007), Boston, MA, August 6, 2007.

[9] D. Spill, A. Bittau, *BlueSniff*. University College London, 2007. http://www.cs.ucl.ac.uk/staff/a.bittau/gr-bluetooth.tar.gz

[10] M. Jakobsson, S. Wetzel, *Security weaknesses in Bluetooth*. LNCS, Vol. 2020, pp. 176-191, Springer-Verlag, 2001.

[11] D. Kügler, *Man in the middle attacks on Bluetooth*. Financial Cryptography, LNCS, vol. 2742, pp. 149–161, Springer-Verlag, 2003.

[12] A. Levi, E. Cetintas, M. Aydos, C. Koc, M. Caglayan, *Relay Attacks on Bluetooth Authentication and Solutions*. Computer and Information Sciences (ISCIS'2004), 19th International Symposium, Kemer-Antalya, Turkey, 2004.

[13] J. Suomalainen, J. Valkonen, N. Asokan, *Security Associations in Personal Networks – A Comparative Analysis*. Nokia Research Center Technical Report NRC-TR-2007-004, 2007. http://www.springerlink.com/content/dk04356586jg4g00/fulltext.pdf

[14] K. Hyppönen, K. Haataja, *"Niño" Man-In-The-Middle Attack on Bluetooth Secure Simple Pairing*. Proceedings of the IEEE Third International Conference in Central Asia on Internet, The Next Generation of Mobile, Wireless and Optical Communications Networks (ICI'2007), Tashkent, Uzbekistan, September 26-28, 2007.

[15] N. Borisov, I. Goldberg, D. Wagner, *Intercepting Mobile Communications – The Insecurity on 802.11*. Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, ACM Press, 2001.

[16] K. Haataja, *New Practical Attack Against Bluetooth Security Using Efficient Implementations of Security Analysis Tools*. Proceedings of the IASTED International Conference on Communication, Network and Information Security (CNIS'2007), Berkeley, California, USA, September 24-26, 2007.

[17] E. Uzun, K. Karvonen, N. Asokan, *Usability Analysis of Secure Pairing Methods*. Nokia Research Center Technical Report NRC-TR-2007-002, 2007. http://research.nokia.com/tr/NRC-TR-2007-002.pdf