“Niño” Man-In-The-Middle Attack on Bluetooth Secure Simple Pairing

Konstantin Hyppönen, Keijo M.J. Haataja
Department of Computer Science, University of Kuopio
P.O.Box 1627, FIN-70211 Kuopio, Finland
Tel. +358 17 16 2587, +358 17 16 2594; Fax +358 17 16 2595
E-mail: {konstantin.hypponen, keijo.haataja}@cs.uku.fi

***Abstract*—As an interconnection technology, Bluetooth has to address all traditional security problems, well known from the distributed networks. Moreover, as Bluetooth networks are formed by the radio links, there are also additional security aspects whose impact is yet not well understood. In this paper, we describe a new Man-In-The-Middle (MITM) attack on Bluetooth Secure Simple Pairing. The attack is based on the falsification of information sent during the input/output capabilities exchange. In addition, we propose countermeasures that render the attack impractical, although not totally eliminating its potential danger.**

***Index Terms*—Authentication, Authorization, 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 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 describe a new practical MITM attack on Bluetooth Secure Simple Pairing. Moreover, we propose countermeasures against the attack.

The rest of the paper is organized as follows. Section II provides an overview of Bluetooth security. Secure Simple Pairing is described in Section III. Our new practical Bluetooth security attack is described in Section IV. Section V discusses related work. Finally, Section VI proposes possible countermeasures and concludes the paper.

II. An Overview of Bluetooth Security

The Bluetooth security begins, when a user 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 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, so there is no need for carrying any heavy laptops in a backpack just to gather data. 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 BlueSniper Rifle. Therefore, the possibility that an attacker is using range enhancement for improving the performance of the attacks should be taken seriously.

1-4244-1007-X/07/$25.00 ©2007 IEEE.

Bluetooth security is based on building a chain of events, none of which provides 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 the 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 (Enhanced Data Rate), 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.

PIN code selection can be done by assigning the same PIN to all Bluetooth devices which belong to a personal Bluetooth network environment. For example, a user can select the same PIN code for a mobile phone, a printer, a DVD player, a mouse and a keyboard, because the user owns and therefore also trusts all Bluetooth devices that are used in the personal Bluetooth network.

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 can also be performed [5].

III. Secure Simple Pairing

Bluetooth version 2.1+EDR [1] adds a new specification for the pairing procedure, namely, Secure Simple Pairing [6]. 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, Secure Simple Pairing 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 the Secure Simple Pairing, 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, Secure Simple Pairing either uses an Out-Of-Band (OOB) channel (e.g., Near Field Communication), 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.

Secure Simple Pairing 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.

TABLE I
Device capabilities and simple pairing 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*.

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.

Secure Simple Pairing 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

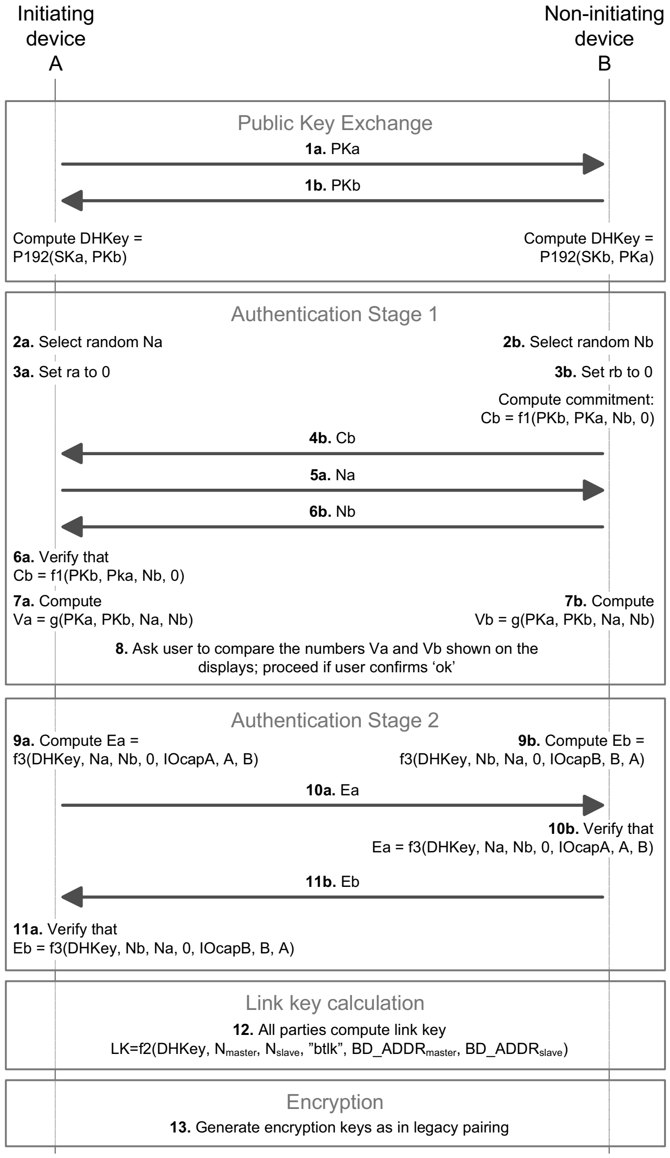

Fig. 1. Secure Simple Pairing with Numeric Comparison association model

generated in this phase, which is the same as the final steps of pairing in Bluetooth 2.0+EDR and earlier.

The contents of messages sent during the Secure Simple Pairing are outlined in Fig. 1, and used notations are explained in Table II. In all figures, and all following text, we concentrate on the Numeric Comparison and Just Works association models only, as these are the models used in our attack.

IV. BT-Niño-MITM Attack

We call our new attack as *BT-Niño-MITM attack* (also referred to as *Bluetooth - No Input, No Output - Man-In-The-Middle attack*). In the attack we exploit the fact that the devices must exchange the information about their IO capabilities during the first phase of the Secure Simple Pairing. 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 his choice. In our attack, the devices are forced to use the Just Works association model, which does not provide protection against the MITM attack.

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

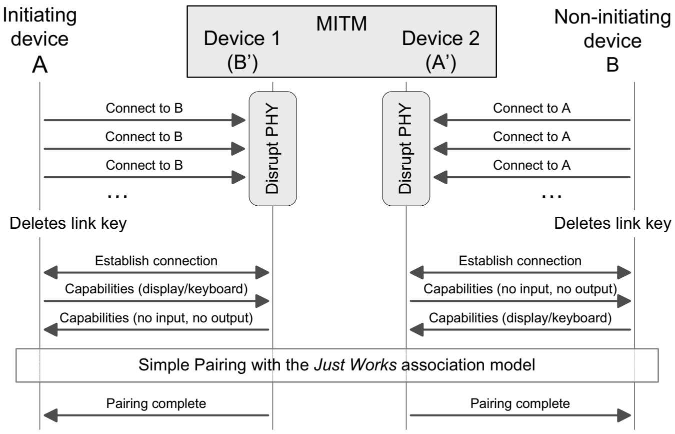

Fig. 2. Main idea of the attack

The MITM uses two separate Blutooth 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. In what follows, we describe three scenarios for the attack.

In the first scenario, the MITM first 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 altogether by using a wideband signal. 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. After that the user initiates a new pairing process by using Secure Simple Pairing, and the MITM can forge messages exchanged during the IO capabilities exchange phase. When the Just Works 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

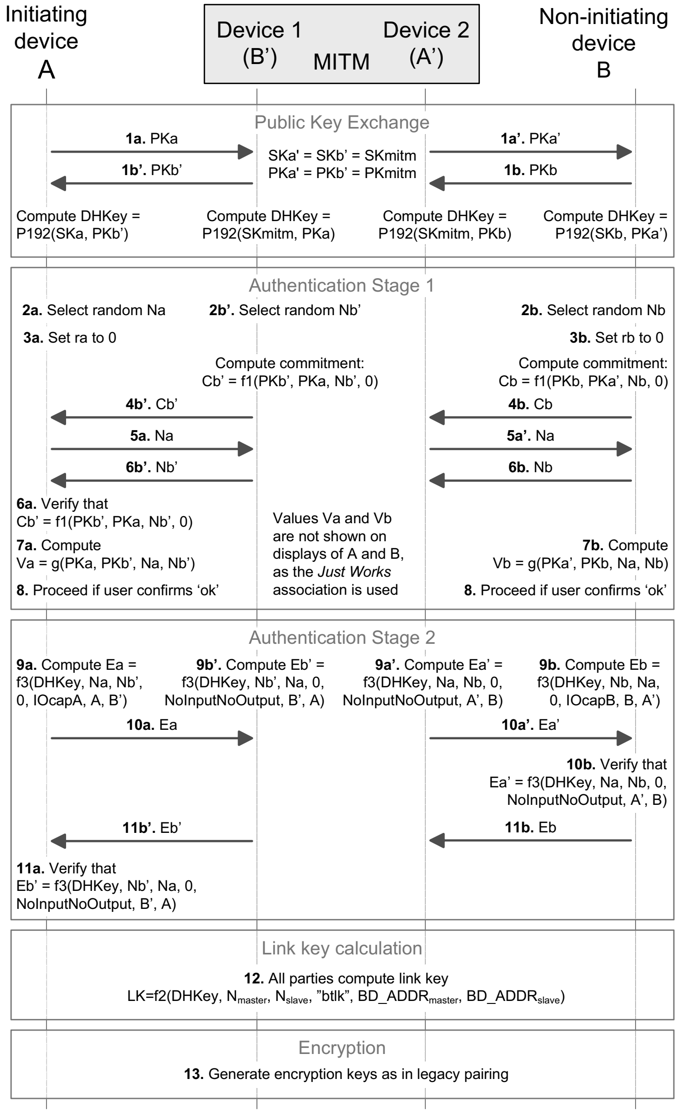
Initiating device A
Device 1 (B')
MITM
Device 2 (A')
Non-initiating device B
Public Key Exchange
1a. PKa
1b'. PKb'
SKa' = SKb' = SKmitm
PKa' = PKb' = PKmitm
1a'. PKa'
1b. PKb
Compute DHKey = P192(SKa, PKb')
Compute DHKey = P192(SKmitm, PKa)
Compute DHKey = P192(SKmitm, PKb)
Compute DHKey = P192(SKb, 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')
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'
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, N_master, N_slave, "btlk", BD_ADDR_master, BD_ADDR_slave)
Encryption
13. Generate encryption keys as in legacy pairing

Fig. 3. Pairing details

on the devices for use in subsequent connections, i.e. the victim devices normally use Secure Simple Pairing without capabilities exchange (see Sect. III).

Other 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. There can be two different scenarios for this kind of devices:

1) *The victim device (A or B) initiates Secure Simple Pairing*: In this scenario, the MITM waits until A or B initiates Secure Simple Pairing. After that, the attack proceeds as illustrated in Fig. 2 and 3.
2) *The MITM (A' and B') initiates Secure Simple Pairing*: In this scenario, the MITM first initiates Secure Simple Pairing 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 Secure Simple Pairing 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 attack, the MITM can intercept and modify all data exchanged between the victim devices, and even use certain services that victim devices offer.

V. Related work

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

Suomalainen *et al.* have performed a comparative analysis of Bluetooth Simple Pairing, Wi-Fi Protected Setup, Wireles USB Association Models, and HomePlugAV security modes [7]. They present an attack against Secure Simple Pairing similar to the one described in this paper. In their attack the MITM prompts one device to use the normal Numeric Comparison association model, while forcing the other device to use the insecure Just Works association model. This leads to the fact that one of the devices (the one which uses the Numeric Comparison 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 the Suomalainen *et al.*, only 6 users out of 40 accepted the pairing on both devices. Compared with this attack, our 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, Secure Simple Pairing 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 Secure Simple Pairing 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 [8]. 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 Secure Simple Pairing.

VI. Conclusions

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. The attack presented in this paper is based exactly on this problem.

The attack can be prevented on the user interface level. If the device that is being paired in the Just Works association model has a display, it must show the integrity checksum and advise the user to compare it with the number shown on the second device, if it has a display. This is a clear trade-off between security and usability.

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. In addition, prior to an access to a sensitive information or services, a Bluetooth-independent re-authentication should be required.

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 Secure Simple Pairing 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 Secure Simple Pairing on their suggestion to use Near Field Communication as the OOB channel.

References

[1] Bluetooth SIG, *Bluetooth specifications 1.0, 1.1, 1.2, 2.0+EDR and 2.1+EDR*. Technical specifications, 1999–2007 https://www.bluetooth.org

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

[6] Bluetooth SIG, *Simple Pairing Whitepaper*, Revision V10r00, August 2006 http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf

[7] 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://research.nokia.com/tr/NRC-TR-2007-004.pdf

[8] 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