Securing Bluetooth Communications

Tzu-Chang Yeh, Jian-Ren Peng, Sheng-Shih Wang, and Jun-Ping Hsu
(Corresponding author: Tzu-Chang Yeh)

Department of Information Management, Minghsin University of Science and Technology
No.1, Xinxing Rd., Xinfeng, Hsinchu 30401, Taiwan
(Email: cheer@must.edu.tw)
(Received Feb. 22, 2010; revised and accepted Mar. 30 & June 15, 2010)

Abstract

Following the increasing confidentiality of data being transferred, many concerns have been raised as to whether Bluetooth transmission is adequately secure. The Bluetooth 2.1 standard introduces a new security mechanism called Secure Simple Pairing (SSP). However, to avoid man-in-the-middle attacks, SSP uses a 6-digit number for authentication. If a human error occurs while conducting visual verification, then data security could be breached. This paper presents an improved protocol to address this problem. This protocol not only secures consumer privacy, but also increases operational efficiency.

Keywords: Authentication, bluetooth, privacy, pairing, security

1 Introduction

Bluetooth, a short range wireless communication standard that allows digital devices to be free from wires, has recently been applied to mobile payments [6, 8]. The Bluetooth SIG (Special Interest Group), founded in 1998, has developed Bluetooth standards to reduce the cost of implementation and speed up its adoption for various applications. IEEE has adopted Bluetooth as the IEEE 802.15 standard for Wireless Personal Area Networks [6].

According to an ABI Research forecast [1], Bluetooth-enabled devices are expected to number nearly 2.4 billion units in 2013. Bluetooth wireless technology has been applied in a wide range of market segments, including software developers, camera manufacturers, mobile PC manufacturers, handheld device developers, consumer electronics manufacturers, and car manufacturers.

As Bluetooth wireless technology is incorporated into more personal mobile devices, it enables new uses for those devices. Recently, it has been applied for use in mobile payments [2, 10, 13, 18]. However, making payments and transferring sensitive data in such a manner necessitates greater protection than existing basic security mechanisms afford [3]. In fact, experts have warned that unless higher security protection is delivered, all transmission of sensitive data over Bluetooth would be unwise [9, 16].

In the days to come, with its high compatibility, Bluetooth could incorporate UWB (Ultra Wide Band), a high bandwidth, short range, ultra-low-power wireless technology, to carry greater amounts of information across longer distances. Therefore, the security issues of Bluetooth transmission are in urgent need of being addressed. The security mechanisms for the resource-constrained devices should also be lightweight [11, 12].

During the authentication and key exchange process of legacy pairing (for Bluetooth 2.0 devices and earlier), much of the information is transferred in plaintext, providing opportunities for a malicious third party to spoof the legal Bluetooth device in order to pass the authentication, or to deduce the encryption key for the purpose of eavesdropping on the data being transferred [4, 15]. The Bluetooth SIG came up with a new standard, Bluetooth 2.1, in July 2007, to tackle the legacy pairing problems through the use of Secure Simple Pairing [4, 5, 14].

To meet the high security requirements for payment applications and to secure consumer privacy, this study thoroughly examines one of Secure Simple Pairing's three protocols, the Numeric Comparison Protocol. This protocol entails a higher degree of security, without demanding supporting communication technologies, and can be easily applied to payment devices, such as cell phones, PDAs, and POS terminals.

This study shows a security flaw, and, accordingly, proposes an easy and convenient improvement protocol by which users can achieve mutual authentication and confidentiality of data transmission using the familiar PIN (personal identification number) entry authentication method. This common authentication method has been widely applied in applications with high security demands, such as the withdrawal of money from ATMs or credit card payments. This improved protocol can ensure consumer privacy and also increase operational efficiency.

2 Secure Simple Pairing

For users, the major difference between Secure Simple Pairing and legacy pairing is that legacy pairing authenticates via PIN entry, while Secure Simple Pairing authenticates by visual number confirmation. The visual number confirmation is used by Secure Simple Pairing to prevent man-in-the-middle attacks caused by the Elliptic Curve Diffie-Hellman (ECDH) protocol.

As shown in Figure 1, the ECDH is a key exchange protocol used to establish a shared key between two connecting devices. Each connecting device starts generating its own random number (device A with SKa, device B with SKb) as its private key, computes the corresponding public key (device A with PKa, device B with PKb), and then send its public key to the other device. Now each connecting device can derive $DHKey$ with its secret key and the received public key. The shared key $DHKey$ can be used as a session key to encrypt all the data transferred between the two connecting devices.

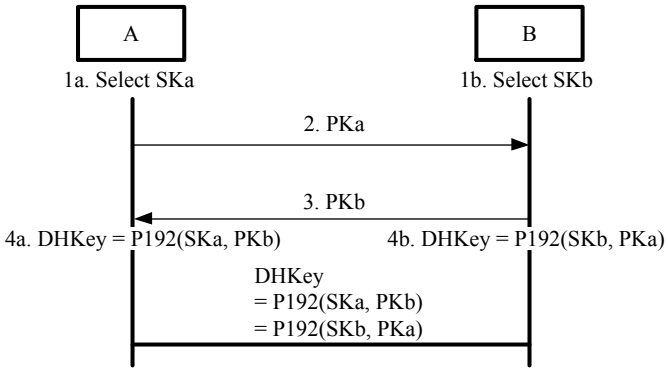

Figure 1: The ECDH key exchange protocol

The notations used in this paper are listed in Table 1.

Table 1: Notations used in this paper

Term	Definition
Cx	Commitment value from device X
$DHKey$	Diffie-Hellman key
Ex	Check value from device X
$P192()$	Used to compute Diffie-Hellman key
$f1()$	Used to compute commitment values
$f2()$	Used to compute link key
$f3()$	Used to compute check values
$g()$	Used to compute numeric check values
$E3()$	Used to compute encryption key
$IOcapX$	Input/Ouput capabilities of device X
LK	Link Key
Nx	Nonce (unique random value) from device X
PKx	Public Key of device X
SKx	Private Key of device X
rx	Random value generated by device X
Vx	Confirmation value on device X
X	Bluetooth device address of device X

A detailed illustration of this process is depicted in Figure 2 and is described below:

Phase 1: Public Key Exchange

A shared key $DHKey$ is established by means of the $ECDH$ key exchange protocol. At the same time, device addresses (A, B) and connecting capabilities $(IOcapA, IOcapB)$ are also exchanged to identify their counterpart and the counterpart's connecting capability.

Phase 2: Authentication Stage 1

The exchange of authentication parameters is confirmed here. This stage contains the following three protocols:

1) **Numeric Comparison Protocol**: Applicable when both devices, such as cell phones and POS terminals, are capable of displaying a 6-digit number and receiving input as "Yes" or "No." This is the protocol discussed in this paper. Each connecting device generates its own random number (device A with Na, device B with Nb); then device B produces commitment value Cb using commitment value function $f1$ and forwards it to device A. Both devices then exchange their random numbers (Na, Nb). With Nb, PKa and PKb, device A computes Cb accordingly and matches it with the Cb received from device B. If they are not identical, the communication is disconnected; otherwise, both devices use numeric verification function g to compute and display 6-digit numbers (Va, Vb), respectively, for the user's further confirmation. Authentication in this phase is completed as Va and Vb share the same value.
2) **Out of Band Protocol**: Applicable when both devices are capable of exchanging important authentication parameters over an out-of-band channel (e.g. Near Field Communication). The out-of-band channel should be able to mitigate both eavesdropping and man-in-the-middle attacks to keep the pairing process as secure as possible.
3) **Passkey Entry Protocol**: Applicable when one of the devices is capable of receiving input but incapable of displaying a 6-digit number, while the other is capable of displaying a 6-digit number, such as Bluetooth keyboards and PCs.

Phase 3: Authentication Stage 2

With the values produced and exchanged, both devices first produce, then exchange check values (Ea, Eb) computed by check function $f3$ to verify

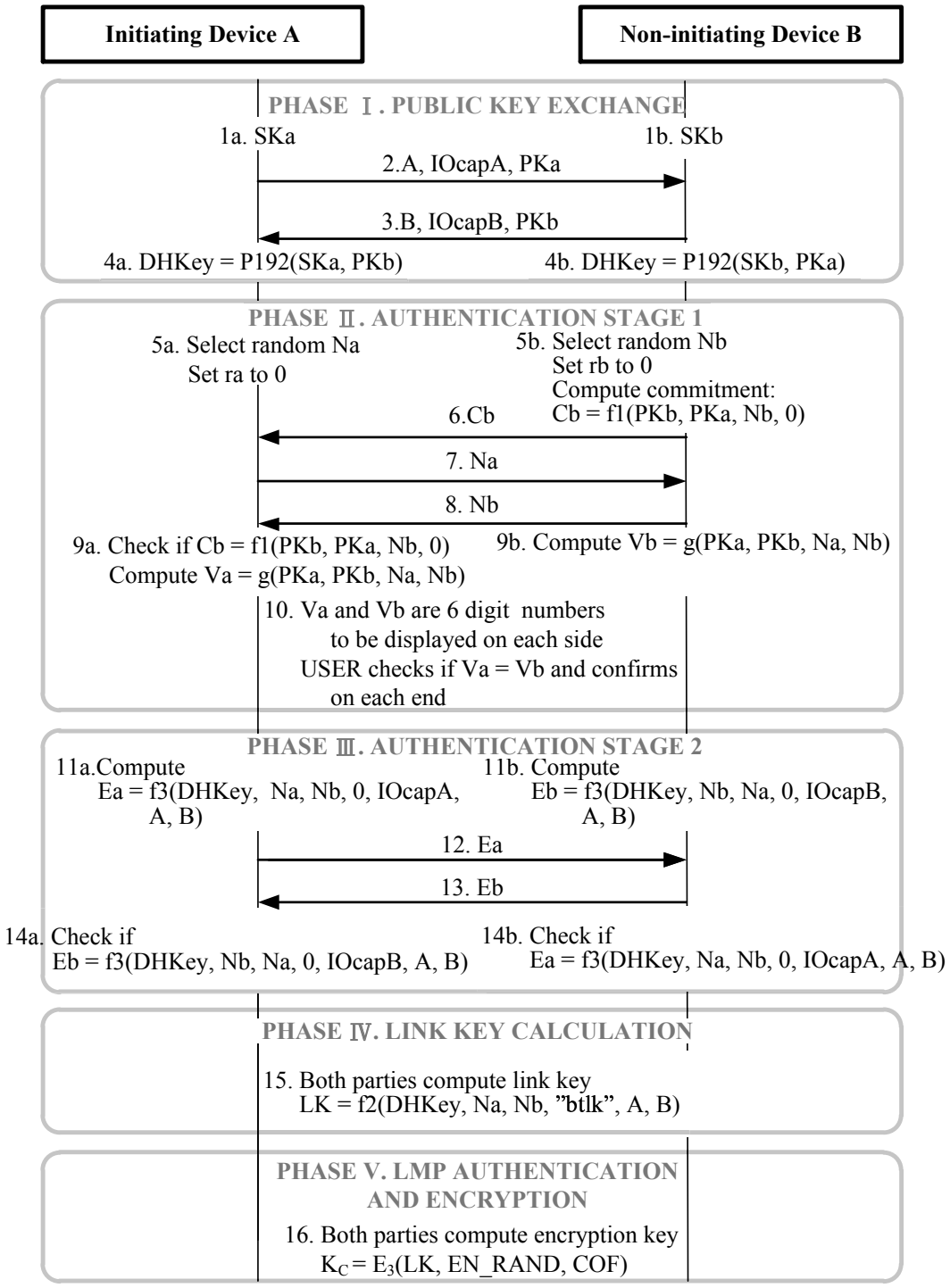

Figure 2: Numeric comparison protocol for secure simple pairing

the complete exchange of all parameters ($DHKey$, Na, Nb, $IOcapA$, $IOcapB$, A, B).

Phase 4: Link Key Calculation

With the input of all the parameters ($DHKey$, Na, Nb, A, B, and string "btlk") gathered from the three previous phases, both devices compute link key (LK) using key derivation function $f2$.

Phase 5: LMP Authentication and Encryption

With the input of COF (Ciphering Offset) having been produced through prior pairing or linkage of both device addresses, random number EN_RAND having been produced in device A and been passed to device B, and link key LK having been generated in Phase 4; both devices compute encryption key K_C using encryption key generation function $E3$.

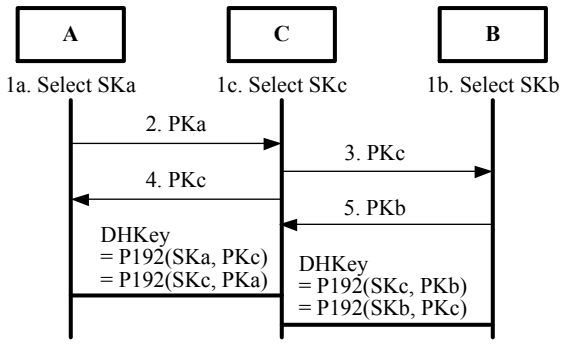

Figure 3: The man-in-the-middle attack against the ECDH key exchange protocol

3 Weaknesses of Secure Simple Pairing

The ECDH key exchange protocol is used by Secure Simple Pairing to provide confidentiality for the data being transferred. However, because the senders of the public keys (PKa, PKb) are not authenticated, the protocol is subject to the man-in-the-middle attack [7]. As shown in Figure 3, the attack works as follows. When A sends PKa to B, an attacker C intercepts this value and impersonates B by replying PKc to A. At the same time, C pretends to be A and sends B the value PKc, and then intercepts the respondence value PKb from B. The result is that C and A share $P192(SKa, PKc)$= $P192(SKc, PKa)$, C and B share $P192(SKc, PKb) = P192(SKb, PKc)$, but A and B mistakenly think they have successfully agreed on a shared key $P192(SKa, PKb) = P192(SKb, PKa)$. Then the attacker C can relay messages between A and B, making them believe that they are talking directly to each other over a private connection where in fact the entire conversation is controlled by the attacker.

To prevent the man-in-the-middle attacks caused by the ECDH key exchange protocol, the visual number confirmation is designed. However, there are still a number of circumstances in which user error can result in security breaches. In a usability experiment conducted by Nokia Research Laboratory [17], each of the two devices computed a 6-digit number which was then displayed on its screen for "Yes" or "No" confirmation by the test users. Despite the assumed ease of operation, the experiment revealed that one in five of the test users pressed "Yes" to indicate that the two displayed numbers matched when, in fact, they did not match. The same result occurred for other applications which relies on the user's visual confirmation, like phishing (similar website), bogus winning bid notice (similar user account), and software installation (to proceed to the next step without going through the terms and conditions).

Because of user error, Secure Simple Pairing has remained vulnerable to man-in-the-middle attacks and, thus, has been unsuitable for applications requiring a high level of security.

4 Our Proposed Protocol

Using the method with which users are most familiar, i.e., entering PIN numbers instead of confirming the displayed numbers, this study proposes an improved protocol. A detailed illustration of this protocol is depicted in Figure 4 and is described below:

Phase 1: Public Key Exchange & Authentication

1) The user inputs PIN on each of the two devices.
2) Each connecting device starts generating its own random number (device A with SKa, device B with SKb) as its private key, and then computes the corresponding public key (device A with PKa, device B with PKb).
3) Device A XORs PKa with PIN and sent the result with A and $IOcapA$ to device B.
4) Device B XORs the received $(PKa \oplus PIN)$ with the PIN entered by the user to obtain PKa, which is computed with SKb to get $DHKey$. In the end, $DHKey$, $IOcapA$, $IOcapB$, B, and A are computed via commitment value function $f1$ to obtain Cb.
5) Device B XORs public key PKb with PIN, and then sends it together with B, $IOcapB$ and Cb to device A.
6) Device A XORs the received $(PKb \oplus PIN)$ with PIN entered by the user to obtain PKb, which is computed with SKa to get $DHKey$, and computes Cb, which is compared with the received Cb. If the result shows any inconsistency, the connection is terminated. Furthermore, Ca = $f1(DHKey, IOcapA, IOcapB\ A, B)$ is computed.
7) Device A sends Ca to device B.
8) Device B computes Ca, which is compared with the received Ca. If the result shows any inconsistency, the connection is terminated.

Phase 2: Link Key Calculation

With the input of the all parameters ($DHKey$, A, B, and string "btlk") received from the previous phase, both devices compute link key LK using key derivation function $f2$.

Phase 3: LMP Authentication and Encryption

With the input of COF (Ciphering Offset) having been produced through prior pairing or linkage of both device addresses, the random number EN_RAND having been produced in device A and passed to device B, and link key LK having been

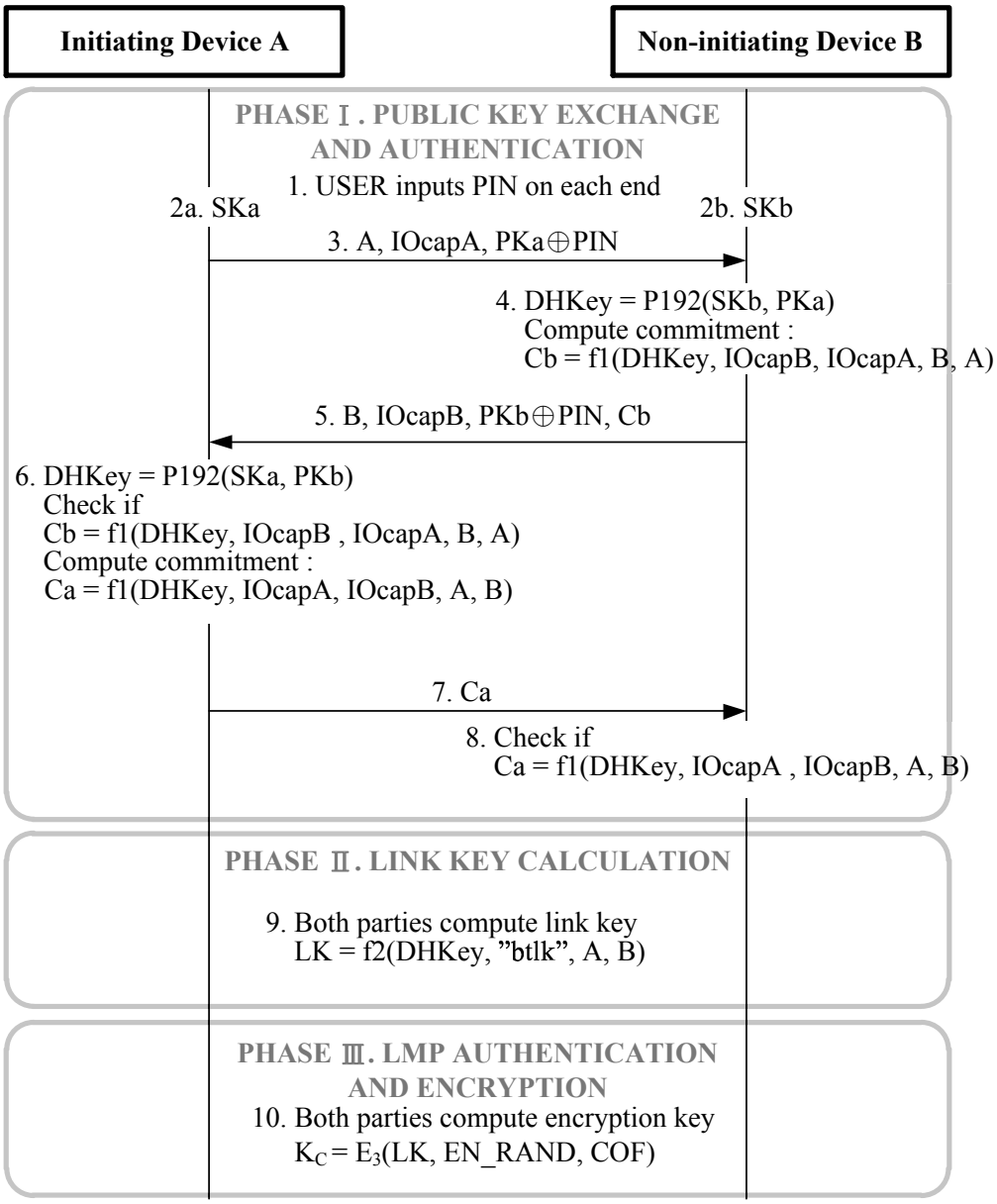

Figure 4: Our proposed protocol

generated in Phase 2; both devices compute encryption key K_C using encryption key generation function $E3$.

5 Analysis

Man-in-the-middle attack: Because the senders of the public keys (PKa, PKb) are not authenticated, the ECDH key exchange protocol used by Secure Simple Pairing is subject to the man-in-the-middle attack. This study follows a common method, which is entering PIN number. Payment devices such as cell phones, PDAs and POS terminals are typically able to receive input from a keyboard or keypad. This method remains in wide use among applications requiring a high level of security, such as the withdrawal of money from ATMs and credit card payments. Only the legal device with the correct PIN can retrieve the correct public key and then derive the correct $DHKey$. If both devices fail to get the same $DHKey$, then the verification of commitment values Ca and Cb fails and the connection is shut down in short order. Moreover, a different PIN can be used for each payment to protect the transmission between the two connecting devices. The Man-in-the-middle attacks can thus be avoided.

Efficiency: The process of Secure Simple Pairing is simplified. The proposed protocol saves computing time for Va, Vb and avoids the need for producing, transmitting,

and comparing Na, Nb, Ea and Eb. The delivery of parameters is verified by Ca and Cb. As a result, the iterative verification process in Phases 2 and 3 of Secure Simple Pairing is simplified so that operational efficiency is increased.

6 Conclusions

As Bluetooth technology dominates data transmission for various kinds of digital devices, major security concerns have been raised. To avoid man-in-the-middle attacks, the Numeric Comparison Protocol for Secure Simple Pairing of new standard Bluetooth 2.1 achieves authentication by conducting visual number confirmation. Given the security problems caused by user error, this study proposes an easy, convenient and improved protocol which applies the familiar authentication method of entering the same PIN number on both connecting devices, as an alternative to confirming displayed numbers. This protocol not only secures consumer privacy, but also increases the efficiency of the operation. The diffusion of Bluetooth technology can therefore be advanced, especially among applications requiring a high level of security.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant NSC 95-2416-H-159-003. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] ABI Research, *Nearly 2.4 Billion Units of Bluetooth-enabled Equipment to Ship in 2013*, 2008. (http://www.abiresearch.com/abiprdisplay.jsp?pressid=1146)

[2] C. Adams and J. J. Chen, "Short-range wireless technologies with mobile payments systems," *The 6th International Conference on Electronic Commerce*, ACM Press, pp. 649-656, 2004.

[3] Bluetooth SIG Security Expert Group, *Bluetooth Security White Paper*, 2002.

[4] Bluetooth SIG, *Bluetooth Specification Version 2.1 + EDR*, 2007. (http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications/Default.htm)

[5] Bluetooth SIG, *Simple Pairing Whitepaper*, 2006. (http://bluetooth.com/nr/rdonlyres/0a0b3f36-d15f-4470-85a6-f2ccfa26f70f/0/simplepairing_wp_v10r00.pdf)

[6] C. Gehrmann, J. Persson, and B. Smeets, *Bluetooth Security*, United States of America : Artech House, 2004.

[7] K. Haataja and P. Toivanen, "Two practical man-in-the-middle attacks on Bluetooth secure simple pairing and countermeasures," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 384-392, Jan. 2010.

[8] D. Kammer, G. McNutt, B. Senese, and J. Bray, *Bluetooth Application Developer's Guide: The Short Range Interconnect Solution*, United States of America: Syngress, 2002.

[9] M. Kotadia, *Nokia Admits Multiple Bluetooth Security Holes*, ZDNet, 2004. (http://news.zdnet.co.uk/communications/0,1000000085,39145886,00.htm)

[10] M. Kwan, *Pay Toll Booths with Bluetooth Phones*, Mobile Magazine, 2007. (http://www.mobilemag.com/content/100/354/C13271/)

[11] Y. Lei, A. Quintero, and S. Pierre, "Mobile services access and payment through reusable tickets," *Computer Communications*, vol. 32, no. 4, pp. 602-610, 2009.

[12] C. T. Li, M. S. Hwang, and Y. P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks," *Computer Communications*, vol. 31, no. 12, pp. 2803-2814, 2008.

[13] S. Pradhan, E. Lawrence, and A. Zmijewska, "Bluetooth as an enabling technology in mobile transactions," *The International Conference on Information Technology: Coding and Computing (ITCC 2005)*, pp. 53-58, 2005.

[14] K. Scarfone and J. Padgette, *Guide to Bluetooth Security*, National Institute of Standards and Technology, 2008. (http://csrc.nist.gov/publications/nistpubs/800-121/SP800-121.pdf)

[15] Y. Shaked and A. Wool, "Cracking the Bluetooth PIN," *The 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 39-50, ACM, Seattle, 2005.

[16] L. Tan, *Symantec Warns Users over Bluetooth Security*, CNET News, 2007. (http://www.news.com/Symantec-warns-users-over-Bluetooth-security/2100-1029_3-6209361.html)

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

[18] A. Zmijewska, "Evaluating wireless technologies in mobile payments - a customer centric approach," *The International Conference on Mobile Business (ICMB)*, pp. 354-362, Sydney, Australia, 2005.

Tzu-Chang Yeh received the B.E. degree in Computer Science and Information Engineering, the M.S. degree in Management Science, and the Ph.D. degree in Information Management from National Chiao Tung University, Taiwan, in 1988, 1990 and 2003, respectively. Now, she is currently an associate professor of the Department of Information Management, Mingshin University of Science and Technology. Her research interests include information security and electronic payment systems.

Jian-Ren Peng received the B.S. degree in Information management from Yuanpei University of Science and

Technology, Taiwan, in 2005. Now, he is currently a graduate student of the department Information Management, Mingshin University of Science and Technology. His major current research interests include information security and wireless communications.

Sheng-Shih Wang received the B.S. degree in the Department of Computer Science and Information Engineering from Tunghai University, Taiwan, in 1993 and the M.S. degree in the Department of Transportation and Communication Management Science from National Cheng Kung University, Taiwan, in 1995, respectively. He received the Ph.D. degree in the Department of Computer Science and Information Engineering from Tamkang University, Taiwan, in 2006. Since 2006, he has been an assistant professor with the Department of Information Network Technology, Chihlee Institute of Technology, Taiwan, and the Department of Information Management, Minghsin University of Science and Technology, Taiwan, respectively. His current research interests include protocol design in wireless sensor networks, Bluetooth networks, and wireless MANs.

Jun-Ping Hsu received the M.S. and Ph.D degrees in Computer Science and Information Engineering in 1992 and 1999, respectively, from the National Chiao Tung University, Hsinchu,Taiwan. From August 1999 to July 2005, he was an assistant professor of the De-partment of Information Engineering at I-Shou University, Kaohsiung County, Taiwan. Since August 2005, he has been an assistant professor of the Department of Information Management at Minghsin University of Science and Technology, Hsinchu County, Taiwan. His research interests include information security, multimedia communication and peer-to-peer networks. Dr. Hsu is a member of the IEEE Communication Society and the Phi Tau Phi honor society of the Republic of China.