Claudio Soriente · Gene Tsudik · Ersin Uzun*

BEDA:

Button-Enabled Device Association

Abstract Secure initial pairing of electronic gadgets is a challenging problem because of the usual lack of a common security infrastructure and the threat of so-called Man-in-the-Middle (MiTM) attacks, whereby an attacker inserts itself into the pairing protocol by impersonating one of the legitimate parties. A number of interesting techniques have been proposed, all of which involve the user in the pairing process. However, they are inapplicable to many common scenarios where devices to-be-paired do not possess required interfaces, such as displays, speakers, cameras or microphones.

In this paper, we introduce BEDA (Button-Enabled Device Association), a protocol suite for secure pairing devices with minimal user interfaces. The most common and minimal interface available on wide variety of devices is a single button. BEDA protocols can accommodate pairing scenarios where one (or even both) devices only have a single button as their "user interface". Our usability study demonstrates that BEDA protocols involve very little human burden and are quite suitable for ordinary users.

Keywords Secure pairing · Human assisted authentication · Man-in-the-middle attacks

1 Introduction

Proliferation of personal gadgets, such as PDAs, cellphones and media players, has brought new services and new possibilities to ordinary users. There are many common settings where two (or more) devices work together, e.g., a Bluetooth headset and a cellphone, a PDA and a wireless printer, or an MP3 player and a cellphone. Before the two devices can operate in tandem, the user must first securely pair them. As part of pairing, devices discover each other via a common – usually wireless – communication channel. Unfortunately, traditional cryptographic means (such as authenticated key exchange protocols) are unsuitable for securing this channel, since unfamiliar devices have no prior context and no common point of trust: no on-line Trusted Third Party (TTP), no off-line Certification Authority (CA), no Public Key Infrastructure (PKI) and no common secrets.

*Corresponding Author.

Computer Science Department, University of California, Irvine E-mail: {csorient,gts,euzun}@ics.uci.edu

The core problem is how to establish a secure communication channel between two previously unassociated devices. Since wireless communication is, by its very nature, human-imperceptible, there is a very real threat of Man-in-the-Middle (MitM) attacks. Such attacks can occur whenever unauthenticated communication is involved. A ready example is the textbook Diffie-Hellman Key Exchange protocol [1] wherein an attacker can easily impersonate either party, such that – at the end of the protocol – both parties think that they are talking to each other, whereas, in reality each is talking with (or through) the attacker.

Some initial pairing solutions require the user to put the two devices into scan/discover modes, respectively, and, once the channel is established, to secure it by entering a user defined password or a PIN into both devices. A number of security and usability issues arise with this general approach. (See [2] for an in-depth discussion.) To this end, a number of recent proposals [3–8] take advantage of certain out-of-band channels (e.g., audio, visual, etc.) to provide secure, yet usable, device pairing. Proposed techniques vary greatly in the assumptions about device capabilities, user competence and involvement, as well as environmental factors.

Several standardization bodies also recognized the importance of the problem and have begun working on specifying more usable and more secure procedures for device association. Wi-Fi Alliance is working on specifications for Wi-Fi Protected Setup [9]. Microsoft has released specifications for Windows Connect Now-NET [10], which is closely related to Wi-Fi Protected Setup. Bluetooth Special Interest Group has released specifications on Simple Pairing [11]. The Universal Serial Bus (USB) forum has recently released the specifications for Wireless USB Association Models [12] which specifies the procedures for pairing two Wireless USB devices. Unlike research proposals, standards specifications have to consider devices with a wide range of hardware capabilities. Consequently, specifications do not dictate a single pairing method. All of them support the use of at least one type of auxiliary channel. For example, Bluetooth Simple Pairing supports the use of Near Field Communication

(NFC) and Wireless USB Association Models support the use of USB cables.

Despite significant recent progress, the design space of the device pairing problem has not been extensively investigated. In particular, one main issue remains unresolved: exotic (or non-ubiquitous) device assumptions. All recent proposals and the standards specifications require certain hardware or interfaces that are not commonly available among across the wide variety of devices that would need secure pairing. Prior techniques envisage devices equipped with (at least one of): cameras, infrared or laser transceivers, accelerometers, speakers, microphones, NFC transceivers, USB ports, keypads and displays. Clearly such devices exist but they are not ubiquitous enough. Moreover, considering the extra cost as well as space and/or esthetic requirements, it seems unlikely that small personal devices will have such capabilities in the near future.

This paper attempts to fill the gap left by prior techniques. The proposed system – BEDA or Button-Enabled Device Association – obviates the need for special hardware in the association process. It aims to accommodate any pair of devices by using a very basic interface, a functional input button (i.e., a single key) that is almost universally available. Another notable feature of BEDA is the use of the human reflexes to transfer information between devices-to-be-paired, perceptibly lowering the threat of MiTM attacks. As described further in the paper and confirmed by our usability experiments, BEDA is both secure and very easy to use.

2 Related Work

There is a fairly large body of relevant prior work on secure device pairing.

The earliest work by Stajano, et al. [13] made a seminal contribution by bringing the problem into the spotlight. The proposed techniques, however, required the use of standardized physical interfaces and cables. The follow-on methods by Balfanz, et al. [14] and Feeney, et al. [15] made progress by using infrared communication as the human-verifiable side-channel. Though timely in its day, this approach is no longer viable since: (1) few modern devices are equipped with IrDA interfaces (they are too slow, short-distance, require line-of-sight and manual start-up) and (2) the infrared channel itself is not fully immune to MiTM attacks.

Another approach involves graphical visualization of the hash of the exchanged cryptographic material. The user then needs to compare the output on both devices. In order to make the comparison easier, researchers devised visual metaphors to represent the hash. Levien and Golberg proposed a "snowflake" mechanism [16,17], Perrig and Song [18] used "Random Art", while Dohrmann and Ellison devised a colorful "flag" representation [19]. Although these schemes avoid the cumbersome and error-prone process of comparing two hashes byte-by-byte, they require high-resolution displays, making the approaches suitable for only certain types of devices, such as laptops, PDAs and high-end phones.

The Seeing-is-Believing (SiB) technique by McCune, et al. [3] uses the visual channel to perform secure device pairing. The visual channel is established between the visual transmitter (bar-code displayed on a screen or a sticker) of one device and the visual receiver (camera) of the other device and devices take turn of taking pictures when mutual authentication is needed. The protocol does not rely on human visual ability (except that the human needs to take a picture) since the devices themselves compare the bar-codes. SiB is applicable to scenarios where at least one device has a camera. Saxena, et al. [5] developed an extension of SiB which achieves secure pairing if one device is equipped with a light detector or a camera, while the other has at least a single LED. The LED device uses its "blinking" capability to transmit authentication data and the other device records the blinking pattern, extracts the data and compares it with its own computed value. This protocol requires less in terms of device features, but not all devices have a light detector or a camera. Moreover, the comparative usability study in [2] indicates that users are generally no adept in following the prescribed order of interaction if it involves more than one device.

Another pairing approach uses a different human-perceptible channel – audio – in the Loud-and-Clear system [4]. As usual, the proposed protocols involve two devices exchanging their keys and computing the hash of the exchanged cryptographic material. The hash is later translated in a syntactically correct English-like "Madlib" (gibberish) sentence that can be either played or displayed depending on the available hardware and the user compares the sequences to verify the key exchange in a user friendly way. The authors consider many other scenarios and variations of the protocol, but each device is required to have a speaker or a display even at the simplest of them. Recently, Soriente et al. [8] took the approach of using audio one step further and realized the secure device pairing over the audio channel where no other common interface, such as Bluetooth or 802.11, is needed. Although using the audio channel for key transmission increases usability, by taking away the burden of establishing another channel, it is only applicable when both devices have a microphone and a speaker.

Other proposals suggested the use of technologies that more expensive and rather exotic. Kinberg, et al. suggested an approach requiring RF and ultrasound receiver/transmitters on both devices in [7] and laser technology (each device must be equipped with a laser transceiver) in a more recent proposal [6]. Holmquist, et al. [20], proposed the use of a common movement pattern as the security initiator when the two devices are shaken together. A similar approach was proposed by Mayrhofer et al. [21]. This requires both devices to be equipped with

two-axis accelerometers; it is also unsuitable for physically large/bulky devices.

Recently, some industrial research and standardization bodies have also published specifications for secure device pairing [9,11,12]. These emerging specifications take the typical approach of doing Diffie-Hellmann key agreement over the insecure channel and then authenticating it using an auxiliary channel. Although the implementation is not specified, each specification supports different hardware configurations at the first look. Bluetooth Simple Pairing [11] requires a display on one device and a display or a keypad on the other. Wi-Fi Protected setup [9] requires a display on one side and a keypad on the other and Wireless USB [12] supports devices with a display. Each of these specifications also support at least one Out-Of-Band channel which is usually even more demanding in terms of required hardware, e.g. USB ports, NFC transceivers or cables.

In summary, aforementioned techniques and recent specifications require specific hardware or interfaces that are simply not available on all devices. There are common pairing scenarios, such as a wireless printer and a laptop, an access point and a PDA, or a wireless headset and a desktop, which are not supported by any of the previously mentioned protocols. Even in some pairing scenarios where the previous schemes apply, one would still need a combination of several such schemes to accommodate a considerable fraction of possible pairing scenarios. Moreover, the usability of such a combination would be very questionable, especially, since no comprehensive usability study has been performed for many of those complex schemes. Moreover, even the very basic pairing methods have not fared well when used by ordinary users [2].

3 Protocol and General Operation

The main goal of BEDA is secure pairing of almost any pair of devices with the emphasis on usability and cost-effectiveness (i.e., minimal additional features to support pairing). To this end, BEDA uses the simplest user interface component, a single functional button, available on almost every device. The auxiliary channel enabled by a single button forms the basis for securing the main communication channel, such as Bluetooth or Wi-Fi. (Note that this main communication channel may need to be initially set up to use BEDA. We consider this to be a reasonable prerequisite, since BEDA aims to secure the already existing communication over it and it is independent of the specifics of the main channel.)

The general BEDA protocol consist of two phases. In the first phase, a short *21-bit* secret value is distributed to both devices over the auxiliary channel. In the second phase, devices authenticate their respective Diffie-Hellmann public keys by proving the knowledge of the secret value to each other in a 21-round protocol.

1. D_1
   - generate a large random R_{i1}
   - compute $h_{i1} = h(1, PK_1, PK_2, P_i, R_{i1})$
   - send h_{i1} to D_2
2. D_2
   - generate a large random R_{i2}
   - compute $h_{i2} = h(2, PK_2, PK_1, P_i, R_{i2})$
   - send h_{i2} to D_1
3. D_1
   - send R_{i1} to D_2
4. D_2
   - if $\hat{h}_{i1} = h(1, PK_1, PK_2, P_i, \hat{R}_{i1})$ then ACCEPT else ABORT
5. D_2
   - send R_{i2} to D_1
6. D_1
   - if $\hat{h}_{i2} = h(1, PK_2, PK_1, P_i, \hat{R}_{i2})$ then ACCEPT else ABORT

Fig. 1 Round i of authentication using the short secret P

The protocol is a variant of the MANA III protocol by Gehrmann, et al. [22]. In this variant of MANA III, the secret is split into 21 pieces and knowledge of one bit is proved in each round. The i^{th} round of the protocol between two devices (D1 and D2) is illustrated in Figure 1, where P_i represents the i^{th} bit of the short secret P and PK_1 and PK_2 are the respective public keys to be authenticated.

For the first phase of the protocol, we considered two main approaches to set and distribute the short secret.

1. Both devices acquire the short secret from the user.
2. One device chooses the short secret randomly and user transfers it to the second device.

In the first approach, both devices acquire the same secret through the use of a single functional button. This is achieved by measuring elapsed time between and during the button press and requiring the user to simultaneously press and release the buttons on both devices, until a long enough secret is acquired. Implementation details and usability analysis of this approach are discussed in the next two sections.

In the second approach, we assume that at least one device has an output interface which is not subject to observation by an attacker. Such output interfaces include: vibration, blinking LED, or a small display. The device with the output interface signals the user (at certain intervals) to press the button on the other device and idle times between button actions are used for transmitting the secret. In such schemes, press and release of a button may or may not be considered as two different actions. In other words, the user may be asked to change the button state from press to release or vice versa at every signal, or s/he may be asked to press and release the button at every signal. The former results in fewer button presses with longer pressing times, while the latter involves more button presses immediately followed by a release. We implemented several protocol variants (that use different output interfaces and button actions) and

performed comparative usability tests. The next section describes the implementation and our usability results are discussed in section 5.

4 Implementation

We implemented and tested four BEDA variants on cellphones. We used the comparative usability testing framework described in [23] for fast protocol development and testing. Pictures of the devices used can be found in appendix A.

In our first implementation, both devices acquire the secret directly from user. The user is required to press and release the buttons on the devices simultaneously and wait for a random (though short) time interval between key-presses. Each device is programmed to start a timer with the first button press and the elapsed time between each button event (either press or release) is then used in determining the short value to be used as the shared secret. Elapsed times between events are kept concatenated until seven events are observed. Each device takes this secret value to the second phase of the protocol. We used 300ms (0.3 seconds) as the smallest unit of measurable time. Exact times measured in milliseconds could not be used here due to the less-than-perfect synchrony between the two hands of an average human user. However, less sensitive (longer) time unit selection tolerates such imperfections and delays. Our choice of 300ms was determined empirically after conducting a small initial study.

We measured elapsed time between each event and reduced it modulo 8 (to obtain a 3-bit value). Over the total of 7 button actions, we thus collected 21 bits of random data. Our choice to construct the secret in 3-bit (i.e., 0 to 7) binary increments was determined after observing, in our pilot study, that users do not wait longer than 3-4 seconds (on average) after they get comfortable with the protocol. Acquiring the secret in 3-bit increments assures the randomness of the resulting secret, even the user is fast-paced and does not wait more than 2.1 seconds between successive events. Note that these values can be further adjusted in individual implementations. We use the term *B-To-B* (button to button) in the rest of the paper to refer to this protocol variation.

For the scenario of one device choosing the secret, we considered two modes of output: vibration and display. In the display implementation, one device shows a black square on its screen and the user is instructed to press a button on the other device whenever the square turns white. After the user starts the protocol, the display-equipped device generates a 21-bit random number and waits 3 seconds before giving the first signal (by coloring the square white for 0.5 seconds). It gives seven more such signals, where each signal is separated by the idle time determined by i^{th} 3-bit segment of the secret. The receiving device, on the other hand, measures intervals between button presses in milliseconds and rounds it to the closest full second. This is needed to tolerate up to 500ms of fluctuation caused by the user's reaction/reflex times. We use *D-To-B* (Display-to-Button) to refer to this version.

Table 1 Participant Profile

Gender	Male	75%
	Female	25%
Age	18-24	15%
	25-29	60%
	30-34	15%
	35+	10%
Education	Bachelor	50%
	Masters	25%
	PhD	25%
Any difficulty with visual abilities	YES (despite any aid)	10%
	NO	90%
Any difficultly with reflex abilities	YES	5%
	NO	95%

One vibration variant employs the same algorithm as D-To-B but gives its signal by vibrating for 500ms (instead of displaying a black square). We refer to it as *SV-To-B* (ShortVibrations-To-Button). The last variant takes a slightly different approach and requires fewer button presses. In it, the user is asked to press-and-hold the button on one device **while** the other one vibrates. This final variant is called *LV-To-B* (LongVibrations-To-Button). To transfer the i^{th} segment of the secret, the sending device either vibrates or remains idle (in alternating order) for t seconds, where t = integer value of i^{th} 3-bit segment of the secret and the sequence starts with vibration. The receiving device considers the pressing and releasing of the button as different events and computes each 3-bit segment of the secret by rounding the measured time between those events, as described earlier.

5 Usability Analysis

Having implemented all four protocol flavors (B-To-B, D-To-B, SV-To-B and LV-To-B), we investigated their usability factors by performing a number of usability experiments discussed in this section.

A total of 20 subjects were recruited. Subjects were chosen on a first-come first-serve basis from the respondents to recruiting posters. Subjects were mainly university students which resulted in a fairly young, well-educated and technology-savvy participant group. The demographics and related background information of the participants are summarized in Table 1.

Test Procedure: Our usability tests were conducted in a variety of campus venues (depending mainly on the subjects' preferences), including, but not limited to: cafés, student dorms/apartments, classrooms, office spaces and outdoor terraces. After giving a brief overview of our study goals, participants were asked to fill out the back-

Table 2 Summary of the related logged data

Method	Average completion time in seconds	Average number of retrials for success
B-To-B	53.2 (sd*=32.5)	2.45 (sd=1.43)
D-To-B	72.8 (sd=39.4)	1.45 (sd=0.89)
SV-To-B	60.1 (sd=18.3)	1.35 (sd=0.49)
LV-To-B	56.6 (sd=19.4)	1.20 (sd=1.41)

*sd= Estimated standard deviation

ground questionnaire to collect demographic information. In this questionnaire, we also asked users whether they were experiencing any visual impairments or have any condition that may interfere with their sensing of vibration or reflexes. Next, users were given a brief introduction to the cellphones used in the tests and the nature of BEDA protocols.

Each user was then given the two devices and asked to follow on-screen instructions shown before each task to complete it. Every user was asked to pair the devices four times in total, using each implementation described in the previous sections. To reduce the learning effect on test results, the four tasks were presented to the user in random order. User interactions throughout the tests were logged automatically by the testing framework. After completing the tasks, each user filled out a post-test questionnaire form and was given 5 minutes of free discussion time followed by a short interview.

Results: We collected data in two ways: (1) by timing and logging user interaction, and (2) via questionnaires and structured interviewing.

Completion time for each protocol was automatically logged by the software. According to this data, using a button on both devices was faster than all the other variants on average, although it needed more trial for success. Whereas, users successfully paired the devices with short signaling vibrations on least number of trials. Albeit, average completion time hovered around roughly a minute in all methods, as shown in table 2.

In B-To-B, the secret is derived from user actions and there is an obvious risk of it being insufficiently random. Our choice of 300ms as the measuring unit was made to achieve a balance between security (randomness) and usability (short completion time). Logs generated by our testing software clearly indicate that derived secrets were indeed random – each octal digit of every derived secret was uniformly distributed in the $[0, 7]$ interval (independently from its position and other numbers in the secret) over the 49 protocol runs, including re-trials. Although B-to-B was not the top choice of most users, it does not seem to suffer, in terms of security, from the fact that human actions serve as the source of randomness.

In the post-test questionnaire, we solicited user opinions about the tested methods. Participants rated each method for its ease-of-use and pointed whatever usability problems they experienced. They were also asked to compare each method to their previous experience (if any) with Bluetooth, Wi-Fi or infrared secure pairing. Users found BEDA variants using vibration to be easiest and commented that they need the least concentration from the user. On the other hand, they found B-To-B to be fairly hard. Such results were not surprising considering the relatively delay-intolerant implementation of B-To-B and the more attention-demanding nature of D-To-B, as compared to vibration variants. The ease-of-use ratings given by participants are summarized in Figure 2.

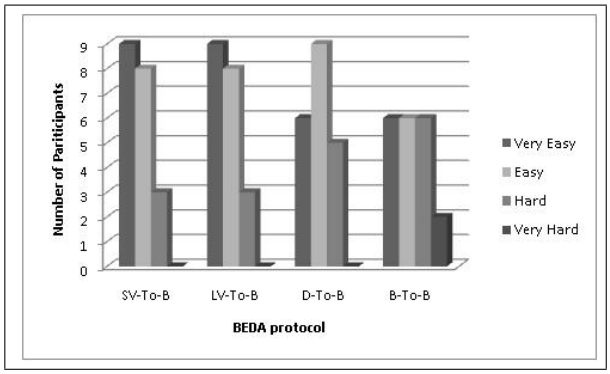

Fig. 2 Participant Opinion

In the post-test questionnaire, we asked users to order the methods they are already familiar with (such as: Wi-Fi, Bluetooth and Infrared secure pairing) and the BEDA suite from the easiest to the hardest. Among the 13 participants who were familiar with Wi-Fi pairing, 77% considered BEDA to be easier. Whereas, among 14 participants familiar with Bluetooth pairing, only 36% considered BEDA easier. During our short post-test interviews, users explained the reason for Bluetooth being easier than BEDA: the former involving just typing in a few (usually four) digits. However, when the number of required digits gets even a little higher (as in WEP or WPA keys in Wi-Fi secure pairing), they find BEDA easier. The interviews also demonstrated that almost all users liked BEDA protocols and enjoyed using them. More interestingly, majority of users (even the ones that rated some BEDA protocols as being hard and found current methods easier) told us they would like to use BEDA instead of current techniques because it is fun to use and simple. (In fact, they emphasized the difference between **simple** and **easy** and classified BEDA as simple).

6 Discussion and Limitation

All BEDA protocols require devices with minimal interface capabilities: a single button on one device and a button, vibration capability or an LED/display on the other. In its simplest flavor, BEDA requires both devices to have a single button. Note that, some forms of output might still be required for the user to acknowledge the outcome of the pairing process. An LED blinking

with a certain pattern, or a simple display might tell the user that the protocol execution was successful and that both devices share the same secret key. Implementation and user-friendliness might vary depending on the device user interface capabilities. However, the conclusion is that BEDA provides paring techniques for devices with the simplest form of user interface, i.e., a single button.

Among the 4 BEDA protocols we studies, there is a clear distinction between B-To-B and the other three variants. The latter use *"human response to a stimulus"* as a conduit for transferring a random secret value chosen by one device to the other device. Whereas, in B-To-B, devices derive the secret value from the user's actions themselves.

At first, B-To-B may not look different from widely adopted secure pairing techniques that require the user to choose a random key and enter it in both devices. However, results from [2] clearly show that the key obtained in such protocols is far from being random. B-To-B, on the other hand, uses the human actions and their timings as the source of randomness and, we believe that this data is more strongly random and would thus result in better overall security. (With 95% statistical confidence, we could not find any evidence to reject the randomness hypothesis over the 343 3-bit segments forming the 49 keys collected on each device). Moreover, existing protocols require a full keypad on both devices, where B-To-B only requires just one button.

During our user studies, the completion time for different BEDA protocols averaged between 53.2 and 72.8 seconds. Although the average completion times are expected to slightly improve as the users get experienced, BEDA protocols would still take longer than some other pairing techniques.

In all protocols, users' reflex time in reaction to different stimuli is very important. Our usability tests show that participants could easily accomplish the pairing using BEDA protocols. Although our participant group was fairly young and generalization to other age brackets is premature, our subjects included two who had experienced visual difficulties (one with cataracts and another – with 60% loss of vision on one eye) as well as one who was taking prescription medication (Xanax). Although either of these factors can influence reflexes and coordination all three of these subjects performed well.

Our D-To-B implementation uses a square turning from black to white. However, we believe that the protocol is equally applicable to simpler devices only equipped with an LED or a primitive one-line display. Turning on and off an LED or showing a one-line word "PRESS" and "RELEASE" would have a similar effect and offer similar usability features.

All BEDA protocols take advantage of the human user either as a conduit for transferring the secret or as a generator of the secret. This is resistant to MiTM attacks only if the transferred or generated secret cannot be observed. Assuming that participating devices are not compromised, the only way to mount a MiTM attack against BEDA is by being close enough to observe either user's or devices' actions. Since devices must be held in the user's hand and be physically close to the user, we claim that an MiTM attack can not remain unnoticed if the attacker gets close enough to the user. Of course, the attacker can always try to observe the user and devices though a hidden camera or binoculars. However, even in that case, the user can take some obvious steps to conceal her actions and/or device output.[1] Recall that the short initial secret is only used to authenticate the Diffie-Hellmann public keys (exchanged via a human-imperceptible medium). To be successful, the attacker must discover the short secret before the devices move into the second phase of the protocol, where they prove the knowledge of the secret. Also, the attacker has only one chance of guessing the secret, since failing to prove knowledge of any bit results in the devices aborting the protocol immediately. Finally, once a protocol terminates, obtaining the short secret key is useless since the security of the subsequent session is based on the Diffie-Hellman key which is of adequate size.

7 Future Work

Our usability tests showed that the relatively short 300ms interval used in the implementation of synchronous button press variant does not provide enough error tolerance for all users and sometimes requires several re-trials. We are in the process of performing further usability tests to enhance the usability of this variant. There is an obvious tradeoff between increased error tolerance and required number of button presses (which influences completion time) in acquiring 21 bits of random data; we hope that further testing and experimentation will aid in determining optimal parameters. We also plan to do conduct more usability experiments in participants' own environment with more comprehensive task scenarios, such as setting up a complete wireless home network with several devices, in the future. This will provide a better insight into the usability of BEDA and a more comprehensive comparison with the current techniques.

References

1. W. Diffie and M. E. Hellman. New directions in cryptography. In *IEEE Transactions on Information Theory*, pages IT–22(6):644–654, November 1976.
2. E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. Technical Report NRC-TR-2007-002, Nokia Research Center, 2007.
3. J.M. McCune, A. Perrig, M.K. Reiter. Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication. In *2005 IEEE Symposium on Security and Privacy*, pages pp. 110–124, 2005.

[1] For example, the user might press devices' buttons in his/her pockets.

4. M.T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human-verifiable authentication based on audio. In *ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, 2006.
5. N. Saxena, J.E. Ekberg, K. Kostiainen, and N. Asokan. Secure Device Pairing based on a Visual Channel. In *2006 IEEE Symposium on Security and Privacy*, 2006.
6. T. Kindberg and K. Zhang. Secure spontaneous device association. In A.K. Dey, A. Schmidt, and J.F. McCarthy, editors, *Ubicomp*, volume 2864 of *Lecture Notes in Computer Science*, pages 124–131. Springer, 2003.
7. T. Kindberg and K. Zhang. Validating and securing spontaneous associations between wireless devices. In C. Boyd and W. Mao, editors, *ISC*, volume 2851 of *Lecture Notes in Computer Science*, pages 44–53. Springer, 2003.
8. C. Soriente, G. Tsudik, and E. Uzun. Hapadep: Human asisted pure audio device pairing. Cryptology ePrint Archive, Report 2007/093, 2007.
9. WiFi Alliance. Wi-fi protected setup specification. WiFi Alliance Document, January 2007.
10. Microsoft. Windows connect now-ufd and windows vista specification. version 1.0. `http://www.microsoft.com/whdc/Rally/WCN-UFDVistaspec.mspx`, 2006.
11. Bluetooth Special Interest Group. Simple pairing whitepaper. `http://www.bluetooth.com/Bluetooth/Apply/Technology/Research/Simple\_Pairing.htm`, 2006.
12. Wireless USB Specification. Association models supplement. revision 1.0. USB Implementers Forum. `Http://www.usb.org/developers/wusb/`, 2006.
13. F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols, 7th International Workshop*, 1999.
14. D. Balfanz, D.K. Smetters, P. Stewart, and H.C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Symposium on Network and Distributed Systems Security (NDSS '02)*, February 2002.
15. L.M. Feeney, B. Ahlgren, and A. Westerlund. Demonstration abstract: Spontaneous networking for secure collaborative applications in an infrastructureless environment.: International conference on pervasive computing (pervasive 2002). 2002.
16. I. Goldberg. Visual key fingerprint code., 1996. Available at http://www.cs.berkeley.edu/iang/visprint.c.
17. R. Levien. PGP snowflake, 1996. Source code available at: `http://packages.debian.org/testing/graphics/snowflake.html`.
18. A. Perrig and D. Song. Hash visualization: A new technique to improve real-world security. In *Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, pages 131–138, July 1999.
19. C.E. and S. Dohrmann. Public-key support for group collaboration. 6(4):547–565, November 2003.
20. L.E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, pages 116–122, London, UK, 2001. Springer-Verlag.
21. R. Mayrhofer and H. Gellersen. Shake well before use: Authentication based on accelerometer data. *In Proc. Pervasive 2007: 5th International Conference on Pervasive Computing*, 2007.
22. C. Gehrmann, C.J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *RSA CryptoBytes*, 7(1):29–37, Spring 2004.
23. K. Kostiainen, E. Uzun, N. Asokan, and P. Ginzboorg. Framework for comparative usability of distributed applications. Technical Report NRC-TR-2007-005, Nokia Research Center, 2007.

A Pictures From Our Implementations

Fig. A1 From top to bottom: B-To-B, D-To-B and SV-To-B