Pairing Devices for Social Interactions: A Comparative Usability Evaluation

Ersin Uzun
Palo Alto Research Center
ersin.uzun@parc.com

Nitesh Saxena
Polytechnic Institute of
New York University
nsaxena@poly.edu

Arun Kumar
Polytechnic Institute of
New York University
aashok01@students.poly.edu

ABSTRACT

When users wish to establish wireless radio communication between/among their devices, the channel has to be bootstrapped first. The process of setting up a secure communication channel between two previously unassociated devices is referred to as "Secure Device Pairing". The focus of prior research on this topic has mostly been limited to "personal pairing" scenarios, whereby a single user controls both the devices. In this paper, we instead consider "social pairing" scenarios, whereby two different users establish pairing between their respective devices. We present a comprehensive study to identify methods suitable for social pairing, and comparatively evaluate the usability and security of these methods. Our results identify methods best-suited for users, in terms of efficiency, error-tolerance and of course, usability. Our work provides insights on the applicability and usability of methods for emerging social pairing scenarios, a topic largely ignored so far.

ACM Classification Keywords

H.5.m Information Interfaces and Presentation: miscellaneous

General Terms

Experimentation, Security, Human Factors

INTRODUCTION

Increasing proliferation of personal gadgets (including PDAs, cell-phones, headsets, cameras and media players) – equipped with wireless communication (e.g., Wi-Fi, Bluetooth) – continuously opens up new services and possibilities for ordinary users. There are many usage scenarios where two devices need to "work together." In commonly occurring, so called *personal* communication scenarios, both devices are controlled by a single user (Alice). Examples include communication between Alice's Bluetooth headset and her cellphone, her PDA and a wireless printer, or her laptop and a wireless access point. On the other hand, *social* communication scenarios, whereby two different users (Alice and Bob) control their respective devices, are also rapidly emerging. Examples include communication between Alice's and Bob's PDAs/laptops/cell phones for social or professional reasons, such as sharing files and music, exchanging digital business cards, multi-player games, messaging, chatting or collaborative applications.

The surge in popularity of wireless devices, however, brings about various security risks. The wireless radio communication channel is easy to eavesdrop upon and to manipulate, raising the very real threats, notably, of so-called *Man-in-the-Middle* (MitM) or *Evil Twin* attacks. To mitigate these attacks, secure communication must be first bootstrapped, i.e., devices must be securely "paired" or initialized.

One of the main challenges in secure device pairing is that, due to sheer diversity of devices and lack of standards, no global security infrastructure exists today and none is likely for the foreseeable future. Consequently, traditional cryptographic means (such as authenticated key exchange protocols) are unsuitable, since unfamiliar devices have no prior security context and no common point of trust.

One valuable and established research direction in secure device pairing is the use of auxiliary – also referred to as "out-of-band" (OOB) – channels, which are both perceivable and manageable by the user(s) of the devices. An OOB channel takes advantage of human sensory capabilities to authenticate human-imperceptible (and hence subject to MitM attacks) information exchanged over the "in-band" wireless channel. OOB channels can be realized using acoustic, visual and tactile senses. Unlike the in-band channel, the attacker can not remain undetected if it actively interferes with the OOB channel, although it can eavesdrop upon it.

For pairing methods based on OOB channels, some degree of human involvement is essential. Usability of the pairing process thus becomes extremely important. We observe that a large majority of existing device pairing methods are proposed by security professionals without giving much emphasis on their usability. Although a few methods have been tested for usability, the testing is done in isolation or with a limited focus on only "*personal pairing*" scenarios.

Motivation

Application domain for secure pairing methods is not limited to personal settings. Two users may wish to exchange files, digital business cards or play games. The main advantage of using Bluetooth or WiFi in such scenarios is that no

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

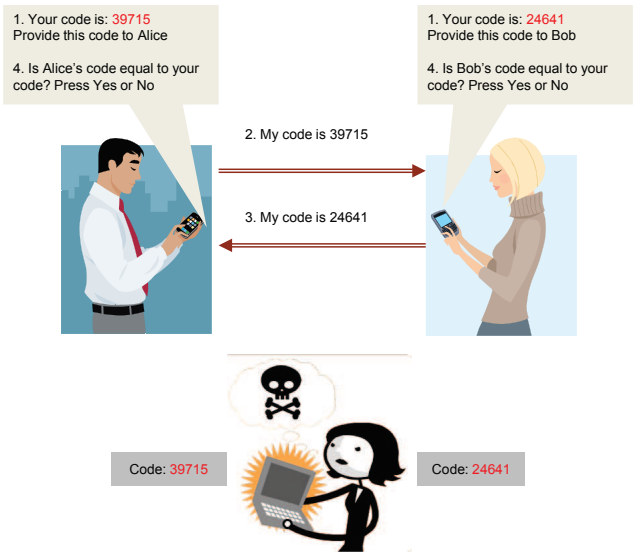

Figure 1. MitM attack scenario for social pairing based on numeric comparison: adversary is executing an instance of pairing with Bob's device and another instance with Alice's device [Step 1: Bob's and Alice's devices show their respective numeric codes as a result of pairing processes; Step 2-3: Bob and Alice exchange their respective codes via OOB communication; Step 4: Bob is asked to compare his code with the one provided by Alice, and Alice is asked to compare her code with the one provided by Bob, and accordingly accept or reject the pairing (in this case, both should be rejecting to prevent the attack)]

infrastructure is needed and *ad hoc* communication can take place without any extra cost to the users. For this reason, social scenarios have been emerging rapidly and are already quite popular, especially, in developing countries. Secure pairing is a natural way to prevent any eavesdropping and/or malicious intervention during intended communication.

Many personal pairing methods have been proposed. A personal pairing method could be directly used in a "*social pairing*" scenario, only if one of the users operates both devices. However, this might not always be desirable or feasible, as discussed below.

- PERSONAL DEVICES: Devices, such as mobile phones, are personal items regarded as extended self-identities of their bearers [9]. For example, users – especially in Asian countries – tend to personalize the exteriors of their phones to make them look unique and representative of their own selves [9]. Thus, users might become nervous or uneasy when asked to share their phones with others.

- SECURITY AND PRIVACY OF DEVICES: Sharing of phones, especially with strangers, might raise security and privacy concerns [14]. For example, Alice might become concerned that Bob would somehow read her emails or delete her folders when she hands over the phone to him for pairing. In fact, as the survey results of our study show (discussed later), majority of users understand such security and privacy implications and are not willing to share their devices with others, even temporarily, during the pairing process. To counter this, pairing application can be executed by Alice in a locked state prior to sharing the phone with Bob, as discussed in [21]. However, this requires Alice to unlock the phone after receiving it back from Bob, thus increasing overall user burden. Physical security might also deter users from sharing their phones. For example, Alice might be concerned that Bob would drop her new and expensive phone and damage it.[1]

- CONTEXT: Even when users are willing to share their phones, the underlying physical and social situation – where the pairing takes place – might not be conducive for phone sharing. For example, Alice and Bob may not be in very close proximity, e.g., sitting across a table in a conference room. Alternatively, users may not be comfortable with holding, using or even being seen with another person's device. An expensive and fragile-looking device (or a highly personalized one with flashy covers and stickers) may easily trigger discomfort. Moreover, taking responsibility for the pairing and interacting with an unfamiliar device in the process can be bothersome, as any failure may result in embarrassment.

For aforementioned reasons, the problem of social pairing can not be simply reduced to personal, one-user, pairing. In fact, social pairing presents unique challenges of its own. First, it is not clear whether personal pairing methods are suitable (and to what extent) when applied in a social pairing scenario. In fact, participation of two users makes the secure pairing process more complicated and potentially error-prone.[2] As an example, to achieve social pairing based on numeric comparison [34] (see Figure 1), an additional layer of interaction between Alice and Bob is needed to compare numbers displayed on their respective devices. Furthermore, not all personal pairing methods are applicable if each device is controlled by a different user. For example, a pairing method that requires both devices to be shaken simultaneously [22] is not suitable for social pairing.

To summarize, there is a pressing need to evaluate applicability, performance and usability of pairing methods suitable for social setting, which is the focus of our paper. Such a study is essential to identify most suitable pairing method(s) for everyday users.

BACKGROUND

Over the recent years, a number of pairing methods have been proposed. They operate over different OOB channels, use different cryptographic protocols and offer varying degrees of usability. All these methods have been proposed in the context of personal pairing and some parts of the following discussion about these methods are adopted from Kobsa et al. [16], a paper that focuses on usability in personal pairing. However, the focus of this paper is social pairing, not personal pairing, and we discuss the applicability of these methods to the social pairing scenarios in Section "Study Preliminaries".

[1]Another extreme possibility is theft of devices. However, it is quite unlikely that a user will indulge in social pairing with someone she does not trust.

[2]On the other hand, unlike personal pairing, devices taking part in social pairing are not usually constrained in terms of input/output interfaces. This simplifies the process to a certain extent.

The initial attempt to address the device pairing problem in the presence of MiTM attacks was "Resurrecting Duckling" [33]. It requires standardized physical interfaces and cables. Although it was appropriate in the 1990-s, this is clearly obsolete today, due to the greatly increased diversity (and decreased size) of devices and the requirement of a physical equipment (i.e., a cable) which defeats the purpose and convenience of wireless connections.

"Talking to Strangers" [1] was another early method, which relies on infrared (IR) communication as the OOB channel and requires almost no user involvement, except for initial setup. Moreover, it has been experimented with user (unlike many other methods), as reported in [2]. However, this method is deceptively simple since IR is line-of-sight and, setting it up requires the user to find IR ports on both devices – not a trivial task for many users – and align them. Also, despite its line-of-sight property, IR is not completely immune to MiTM attacks. Another drawback is that IR has been largely displaced by other wireless technologies (e.g., Bluetooth) and is available on few modern devices.

Another approach involves image comparison. It encodes the OOB data into images and asks the user to compare them on two devices. Prominent examples include: "Snowflake" [6], "Random Arts Visual Hash" [27] and "Colorful Flag" [5]. Such methods, however, require both devices to have displays with sufficiently high resolution and applicability is limited to high-end devices, such as: laptops, PDAs and cell phones. These methods are based on the protocol proposed in [1] which was reviewed earlier. A more practical approach, based on Short Authenticated Strings (SAS) protocols [26, 19], suitable for simpler displays and LEDs has been investigated in [29].

More recent work [24] proposed the "Seeing-is-Believing" (SiB) pairing method. In its simplest instantiation, SiB requires a uni-directional visual OOB channel for one-way authentication: one device encodes OOB data into a two-dimensional barcode which it displays on its screen and the other device "reads it" using a photo camera, operated by the user. At a minimum, SiB requires one device to have a camera and the other – a display for uni-directional authentication and both devices to have a camera and display for bi-directional authentication. Thus, it is not suitable for small or low-end devices. From the user's perspective, SiB is a relatively undemanding pairing method as user actions amount to taking a photo of a barcode.

A related approach, called "Blinking Lights" has been explored in [30]. Like SiB, it uses the visual OOB channel and requires one device to have a visual receiver, e.g., a light detector or a video camera. The other device must have at least one LED. The LED-equipped device transmits OOB data via blinking while the other receives it by recording the transmission and extracting information based on inter-blink gaps. The receiver device indicates success/failure to the user who, in turn, informs the other to accept or abort.

Quite recently, [28] developed a pairing method based on synchronized audio-visual patterns. Three proposed methods, "Blink-Blink", "Beep-Beep" and "Beep-Blink", involve users comparing very simple audiovisual patterns, e.g., in the form of "beeping" and "blinking", transmitted as simultaneous streams, forming two synchronized channels. One advantage of these methods is that they require devices to only have two LEDs (one of which is to ensure synchronization) or a basic speaker.

Another recent method is "Loud-and-Clear" (L&C) [7]. It uses the audio and/or visual OOB channels along with MadLib phrases which represent the digest of information exchanged over the main wireless channel. There are three L&C variants: "Phrase-SS", "Phrase-DS" and "Phrase-SS". In the first one, user compares two displayed phrases and in the last one, two vocalized ones. The middle one requires the user to compare a displayed phrase with its vocalized counterpart. Minimal device requirements include a speaker or a display on each device and the user either accept or abort the pairing based on the outcome of the comparison. i.e., whether the phrases are the same. As described in [7], L&C is based on the protocol of [1]. In this paper, to reduce the number of words in the MadLib sentences, we use the L&C variant based on SAS protocols [26, 19]. The third variant of L&C, "Phrase-DD," simply involves displaying the sentences on two devices, which the user is asked to compare.

Some follow-on work (HAPADEP [32]) considered pairing devices that – at least at pairing time – have no common wireless channel. HAPADEP uses pure audio to transmit cryptographic protocol messages and requires the user to merely monitor device interaction for any extraneous sounds or interference. It requires both devices to have speakers and microphones. To appeal to more common setting (one where a common wireless channel is available), we employ a HAPADEP variant, we call "Over-Audio." This variant uses the wireless channel for cryptographic protocol messages and the audio – as the OOB channel. In it, only one device needs a speaker and the other – a microphone. Also, the user is not involved in any comparisons.

An experimental investigation [34] presented the results of a comparative usability study of simple pairing methods for devices with displays capable of showing a few (4-8) decimal digits of OOB data. In the "Compare-Confirm" or "Numeric-Compare" approach, the user simply compares two 4-, 6- or 8-digit numbers displayed by devices. In the "Select-Confirm" approach, one device displays to the user a set of (4-, 6- or 8-digit) numbers, the user selects the one that matches a single such number displayed by the other device. In the "Copy-Confirm" approach, the user copies a number from one device to the other. The last variant is "Choose-Enter" which asks the user to pick a "random" 4-to-8-digit number and enter it into both devices. All of these methods are undoubtedly simple, however, as [34] indicates, Select-Confirm and Copy-Confirm are slow and error-prone. Furthermore, "Choose-Enter" is insecure since studies show that the quality of numbers (in terms of randomness) picked by the average user is very low.

The approach Button-Enabled Device Authentication (BEDA) [31] suggests pairing devices with the help of button pressing, thus utilizing the tactile OOB channel. It has has several variants: "BEDA-Blink", "BEDA-Beep", "BEDA-Vibrate" and "BEDA-Buttons". In the first three variants, respectively, the sending device blinks its LED (or beeps or vibrates) and the user synchronously presses a button on the receiving device. Each 3-bit block of the SAS string is encoded as the delay between consecutive blinks (or beeps or vibrations). As the sending device blinks (or beeps or vibrates), the user presses the button on the other device thereby transmitting the SAS between the devices. In the BEDA-Buttons variant, which can work with any PAKE protocol (e.g., [3]), the user simultaneously presses buttons on both devices and random user-controlled inter-button pressing delays are used as the means of establishing a common secret.

A very different OOB channel was considered in "Smart-Its-Friends" [8]: a common movement pattern is used to communicate a shared secret to both devices as they are shaken together by the user. A similar approach is taken in "Shake Well Before Use" [22]. Both techniques require devices to be equipped with 2-axis accelerometers. Although some recent mobile phones (e.g., iPhone) are equipped with it, accelerometers are not common on other devices.

There are also other methods involving technologies that are relatively expensive and uncommon. To summarize a few. [15] suggested using ultrasound as the OOB channel. A related technique uses laser as the OOB and requires each device to have a laser transceiver [23]. Other methods require Near Field Communication technology and devices to be touched with each other. However, the hardware needed for these methods are not readily available in many current devices and are not expected to be ubiquitous soon.

Recently, comprehensive and comparative studies of different personal pairing methods have been introduced in [12, 16] and [18]. In [18], authors selected 13 pairing methods that they deemed practical and comparatively investigated the security and usability of them. [16, 12] also conducted similar studies and all these studies indicate that Numeric-Compare out-performed other methods in terms of efficiency, security and usability. They also show that all methods involving manual comparison of SAS data yielded non-zero error rates in most cases. Unlike the work we present in this paper, these studies evaluated methods suitable for personal pairing. As we discuss later, our results stand in contrast to the results of all these prior studies. Most recently, [11] and [25] looked into the group pairing setting. However, [11] included only one group consisting of 2 users and [25] experimented with groups with at least 4 users. Both papers evaluated different sets of methods from the current paper and their focus was pairing methods geared for larger group sizes (more than 2 users). Another study [10] examines user attitudes and behaviors while pairing devices under different contexts, and is complementary to the current paper.

STUDY PRELIMINARIES

Methods Tested

There is a large body of prior research on secure device pairing. All of these methods were proposed in the context of a personal pairing setting.

There are more than twenty methods, counting variations, in the literature. However, some of them have very limited use cases due to requiring both devices to be controlled by the same user during the pairing (e.g., accelerometer-based methods such as [22]) or requiring hardware not ubiquitous among wireless devices. Some methods have stronger assumptions about the OOB channel and require it to be confidential (e.g., BEDA-Buttons variant of [31]). Notice that secret OOB channels are hard to achieve in real-life since a close-by attacker can easily eavesdrop on any human perceptible channel (e.g., by shoulder surfing).

We believe that it is very difficult to test all available methods in one single study and hope that our results yield meaningful comparative usability metrics. Obstacles, such as varying security assumptions about the OOB channel among different methods and possible user fatigue from including too many methods would undermine study results. Consequently, we have to cull the number of methods down to a more manageable number, eliminating those that are obsolete, deprecated based on prior evaluations or unrealistic due to their OOB assumptions. Of course, we also eliminated any methods that are limited to personal pairing only. The following methods are excluded from our study:

- Resurrecting-Duckling [33]: obsolete due to physical equipment, i.e., cable, requirement.
- Talking-to-Strangers [1]: obsolete since IR ports are not secure against MitM attacks and IR has become uncommon.
- Choose-and-Enter [34], Copy [34], BEDA-Buttons [31]: requires a secret OOB channel and/or performed poorly in prior evaluations.
- Beep-Beep [28]: performed poorly in prior evaluations due to user annoyance and high error rate.
- Blink-Blink [28], Image Comparison [6, 27, 5]: do not extend well to the social pairing setting, since two devices need to be placed adjacent to each other or temporarily exchanged between users.
- Seeing-is-Believing [24], Blinking Lights [30]: require photo or video cameras on devices and do not extend well to the social pairing setting due to the need for close proximity between the devices; also cameras are not ubiquitous interfaces except for mobile phones.
- BEDA-Vibrate [31]: vibration is not a common interface, except for mobile phones; also it is hard for one user to sense the vibration on another user's device, making this method unusable in a social pairing setting.
- Smart-its-Friends [8], Shake-Well-Before-Use [22]: requires one user to hold and control both devices and thus do not extend to social pairing scenarios.

- Ultrasound- [15] and laser-based [23] methods: requires hardware capabilities not common across devices.

Remaining methods, namely **BEDA-Beep**, **BEDA-Blink**, **Beep-Blink**, **Over-Audio**, **Numeric-Compare**, **Phrase-DD**, **Phrase-DS**, **Phrase-SS** and **Copy-Confirm**, have been included in the study. These were selected based on their suitability for social pairing scenarios. We had to slightly modify certain methods to standardize OOB assumptions and security level. In particular, we updated all methods to be based on a SAS protocol for better efficiency and unified security assumptions. This resulted in a slightly changed user interaction in BEDA-Beep, BEDA-Blink and Over-Audio methods.

Test Devices and Implementation

In a personal pairing setting, one of the devices might be interface-constrained. For example, a headset, being paired with a cell phone or an access point being paired with a laptop, are constrained devices (with no display, keypad). On the other hand, both devices participating in a social pairing scenario are "personal" devices (such as PDAs, cell phones, laptops) and are usually not constrained. These devices are generally equipped with at least a display and a keypad.

We wanted to simulate, as closely as possible, common social pairing scenarios. To this end, for our entire study, we used two Nokia cellphones models:[3] N73 and E61, as test devices. Our test devices have all the features and interfaces needed for the tested methods, such as a display, keypad, speaker, microphone and Bluetooth.

For methods that involve beeping, we configured a general-purpose speaker for use as a beeper. A picture of a bright LED displayed on the screen to simulate a blinking LED (Utilizing the whole screen rather than an LED is an obvious choice for social pairing).

To achieve a unified software platform, we used the open-source comparative usability testing framework developed by Kostiainen, et al. [17][4] that provides basic communication primitives between devices as well as automated logging and timing functionality. However, we implemented separate user interfaces and simulated functionality for all tested methods.

For all methods, we kept the theoretical security level constant [5]. We also tried to keep all user interfaces similar, while applying same design practices, i.e., safe-default selection prompts, clear instructions, simple language and so on. In all tests, Bluetooth was used as the wireless radio channel and the initial rounds of the underlying cryptographic protocol running over the Bluetooth channel is omitted. Instead, our implementation supplies devices with synthetic SAS strings to realistically simulate normal and MiTM attack scenarios.

[3]For N73 specs, see: `www.nokiausa.com/A4409012`, and for E61 – `europe.nokia.com/A4142101`.

[4]The same framework was also utilized by [34, 18, 16] to implement and test various pairing methods.

[5]we used SAS string length of 15 bits for all methods, which provides reasonable security for many applications[35]

USABILITY TESTING DETAILS

Having implemented all selected social pairing methods on a common platform, we set out to evaluate and compare social pairing methods (identified previously) with respect to the following factors:

1. *Efficiency*: time to complete each method
2. *Robustness*: how often each method yields false positives (rejection of a successful pairing instance) and false negatives (acceptance of an unsuccessful pairing instance). Following the terminology introduced in [34], we refer to the former category as *safe errors* and the latter – as *fatal errors*.
3. *Usability*: how each method fares in terms of user burden (i.e., ease-of-use perception), successful task completion and personal preference.
4. *User Interactions*: how two users interact in order to perform steps involved in each method.

Study Participants

We recruited 40 participants. At any given point, two participants had to be present to complete the tests. The study lasted over a period of more than two months. Each pair of participants was chosen very carefully as we required them to have varying trust relationships with each other, ranging from being strangers, to acquaintances to close friends. Each pair was briefed on the estimated amount of time required to complete the tests. Participants were mostly young (18-29 years old) university students both at undergraduate and graduate level. Thus, our study represents only the first step towards identifying methods suitable for the broad cross-section of user population.

We prepared two questionnaires: *background* – to obtain user demographics and *post-test* – for user feedback on methods tested. None of the participants reported any physical impairments that could interfere with their ability to complete given tasks. The gender split was: 65% male and 35% female. Also, prior to testing, we collected information on whether the participants knew each other and if so, how well.

Trust between participants: Among 20 subject pairs, 5 have not met before (were complete strangers), 5 were close friends, and the remaining 10 were friends or colleagues who did not consider each other as close friends. In order to gain some insight into the trust relations and acceptable interaction between subject pairs, we asked them whether they would consider temporarily handing their device to the other person in order to initiate a secure connection that they can later use to exchange files, messages or play games. We also asked their reasoning and concerns related to answers.

Not surprisingly, all 5 pairs that have not met before said they would not consider any physical exchange of devices as part of an acceptable interaction. The two main concerns were: the security of the device and the data it stores as well as the unpleasant social situation it may create. On the other hand, 4 out of 5 pairs of close friends did not state any privacy concerns and indicated that they do not mind exchanging their devices during pairing. Among 10 pairs acquain-

tances, 6 expressed serious concerns about any physical device exchange and considered it unacceptable; their reasonings were similar to those of the first group.

Based on observed trust relations and concerns expressed by our subjects, we conclude that any method that needs physical exchange of devices is unacceptable in many scenarios where the owners *do not know each other very well*. Moreover, it may still be problematic in some situations where owners know each other Among 8 pairs who were not reluctant to exchange devices, the relationship between the users played a strong role. Surprisingly, 5 among 8 pairs only considered friends as the acceptable social group to temporarily exchange devices and even excluded family members. The remaining 3 considered both family and friends as acceptable. However, we believe that the observed strong tendency to share devices with friends (rather than with the family members) was perhaps due to a somewhat biased sampling of our subjects, i.e., mostly single college students.

Testing Process

We created 9 test cases. They were designed such that the attacks occur probabilistically, meaning that the user does not encounter both "no-attack" and "under-attack" scenario for each method but encounters either with a 50% chance. This prevented users from expecting one no-attack and one under-attack test case for each method and reduced the number of tests. The order of tests presented to the user was counterbalanced for learning effect using the Latin Square design. During the experiments, test devices are set to have their keypad lights, Bluetooth interfaces and screen-backlights always on and their screensaver functionalities were disabled.

Our study was conducted in a variety of campus venues including: student laboratories, cafés, student dorms, classrooms, office spaces and outdoor terraces. This was possible since test devices were mobile, test set-up was more-or-less automated and only minimal involvement from the test administrator was required.

After giving a brief overview of our study goals, we asked the participants to fill out the background questionnaire to collect demographic information. Both participants in each test jointly filled out the questionnaire. After a short interview about the relationship between each pair of subjects, they were given a brief introduction to cell-phone devices.

Each pair of users was given two devices (one per user) and asked to follow on-screen instructions in completing each task. Users were closely watched to observe whether they exchanged devices during the tests.

User interactions were observed by the test administrator and timings were logged automatically by the testing framework. After completing the tasks, each user-pair jointly filled out the post-test questionnaire, where they provided feedback on tested methods and also indicated whether they found any particular test to be difficult or problematic. They were also given a few minutes of free discussion time, in order to offer comments to the test administrator.

Test Results

For each method, completion times, errors, actions and the playcount, i.e., number of trials before successful pairing was established, were automatically logged by the software. Collected data is summarized in Table 1.

Method	Avg. time* (in seconds)	Fatal error rate	Safe error rate	Avg. # of trials until success
BEDA-Beep	40.43 (se^{+}=4.57)	**0.00**	0.14	1.14
BEDA-Blink	96.00 (se=21.9)	**0.00**	0.10	2.20
Beep-Blink	45.10 (se=5.46)	0.09	0.11	1.20
Over-Audio	18.75 (se=2.74)	**0.00**	**0.00**	1.13
Numeric-Compare	12.50 (se=4.75)	**0.00**	0.10	N/A
Phrase-DD	11.44 (se=1.43)	**0.00**	**0.00**	N/A
Phrase-DS	21.45 (se=5.96)	**0.00**	**0.00**	N/A
Phrase-SS	38.71 (se=16.0)	**0.00**	**0.00**	N/A
Copy-Confirm	17.00 (se=2.72)	0.17	**0.00**	N/A

*Completion time in normal (non-attack) test cases

$^{+}$se=standard error of the mean

Table 1. Summary of Logged Data

We also observed user interactions while each pair of users was performing the various steps involved in each method. In general, we observed that subjects often decided the outcome of pairing based on mutual agreement, which, we believe, may have helped reduce errors in most comparison-based methods. We did not, however, observe any significant effect of the closeness of the relationship between the participants on their interactions during the pairing process. Observed interactions for each method are summarized below. (Assume Alice is pairing her device A with Bob's device B)

• BEDA-Beep: The user responsible for pressing the button (Bob) would listen carefully the beeping on the other device (A) and synchronously press any button on B. In most cases, the user of the beeping device (Alice) moved closer to Bob within a distance of about 1-2 feet so that he could clearly hear the beeping sound. Alice was also noticing if Bob was synchronizing the beeping with the button press. Once finished with this phase, Bob verbally notified Alice to accept or reject the pairing, based on the result shown on B.

• BEDA-Blink: The user of the blinking device (Alice) would show her device to the other user (Bob) who would press the button in synchronization with blinking. Users were again 1-2 feet apart. Once finished with this phase, Bob verbally notified Alice to accept or reject the pairing, based on the result shown on B.

• Beep-Blink: After starting the pairing process, both Alice and Bob compared the blinking/beeping on their own device with the beeping/blinking on the other device. This required the two users to be within touching distance of each other so that both could watch the flashing screen and listen to beeping of respective devices. At the end, both users accepted or rejected the pairing, based on their mutual judgement of whether blinking/beeping was synchronized.

• Over-Audio: In this method, the role of the two users was "passive." Alice's device would start to play an audio clip that encoded a bit-string and Bob's device would automatically record it. After the audio transfer, Bob would (ver-

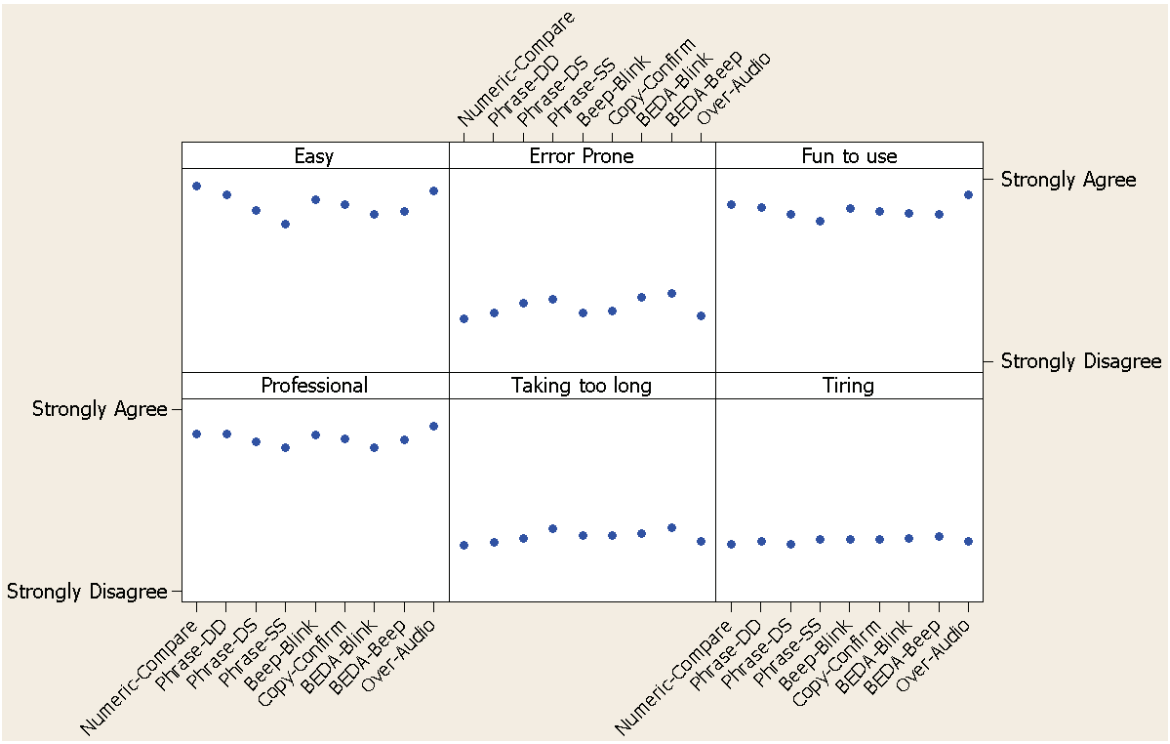

Figure 2. Means of all user ratings for the tested methods

bally) tell Alice to accept or reject depending on what his screen indicated.

• Numeric-Compare: In this method, both Alice and Bob either spell out or show the number displayed on the screens of their devices, compare the two and mutually accept or reject the pairing. Most subjects in our pool preferred to spell out displayed numbers.

• Phrase-DD: Similar to Numeric-Compare, both Alice and Bob either spell out or show the sentence displayed by their devices, compare the two and accept or reject the pairing, based on mutual agreement. Again, most test participants preferred to spell out the displayed phrase.

• Phrase-DS: This method involves the user (Alice) listening carefully to the sentence spelled out by the device of other user (Bob) and then comparing it with the sentence displayed on the screen of her device, and vice versa. For this to take place, Bob would bring his device near (about 1-2 feet) Alice's device, in order for her to be able to hear the spoken sentence. Both participants mutually accepted or rejected the pairing, following a short discussion.

• Phrase-SS: This method involves both devices vocalizing a sentence. When the devices "speak", both users would lean toward them in order to hear clearly. After listening, they would determine whether the sentences matched. Participants then accepted or rejected the pairing on their respective devices, after verbally confirming with each other.

• Copy-Confirm: Alice would either read out the number displayed on A or directly show the screen displaying the number to Bob. After inputting the number into his device, Bob would verbally notify Alice to accept or reject, depending instructions on B's screen. We observed that participants preferred to read out the number and a few also showed it to the other user after spelling out.

Finally, through the post-test questionnaire, we solicited user opinions about all tested methods. Each user-pair rated each method on a 6-level Likert scale[20]. Ratings included: *easy to use, professional, fun to use, tiring, takes too long to complete, and error prone*. Means of all user ratings are graphed in Figure 2.

INTERPRETING RESULTS

We now attempt to interpret the results of our study. We first consider various mechanical data, i.e., time to completion and error rates. Then, we analyze perceived qualitative aspects of the methods based on collected user ratings. We finalize our interpretations by looking at all measures combined with principal component and cluster analyses.

Interpreting Time and Error Results

Our results, summarized in Table 1, prompt a number of observations

Completion time: The logical way to interpret the completion time is by looking at it under normal circumstances, i.e., when no active or passive attacks occur. Thus, we only considered the no-attack test cases while calculating average completion time. Based on this performance metric, tested methods fall into three speed categories: fast (less than 20 secs), moderate (between 20 and 30 secs) and slow (more than 30 secs). The fastest method is Phrase-DD at 11.44 secs

for a successful outcome, closely followed by the Numeric-Compare at 12.5 secs. Copy-Confirm and Over-Audio are next, taking 17 and 18.75 secs, respectively. Phrase-DS comes in at 21.45 secs; its performance is considered moderate and acceptable. The slow category includes the rest, ranging from Phrase-SS (38.71 secs) to BEDA-Blink which takes a whooping 96 secs. The differences between completion times for BEDA-Blink and all other methods were found to be statistically significant at 5% level.

Error Rates: As discussed in earlier, *fatal errors* have a significant impact on security of pairing since they can result in successful MiTM attacks. On the other hand, *safe errors* do not constitute an immediate security threat but can cause user annoyance as the pairing process has to be repeated. However, in addition to poor usability, safe errors may eventually influence security because high levels of user annoyance may result in careless behavior that can, in turn, cause fatal errors.

Most methods, fare well reporting no fatal errors. The two exceptions are Copy-Confirm and Beep-Blink that exhibited fatal error rates of 17% and 9%, respectively, which constitutes a significant vulnerability in the context of security applications. BEDA-Beep, BEDA-Blink, Beep-Blink and Numeric-Compare methods all yield higher than 10% error rates. However, this was for safe errors considered not security-relevant. However, as discussed above, safe errors are indication of poor usability and may eventually adversely impact security.

Timing and Error Rates: Looking at timing and error results together, it (fortunately) appears that the fastest method is also one of the error-free methods. Taking both factors into account, the best overall method is clearly Phrase-DD, followed by Over-Audio and Phrase-DS. Although Numeric-Compare is also quite fast, there is little motivation for using it over Phrase-DD. The reason is simple: both require the same hardware (basic displays) and Phrase-DD offers lower error rates and takes about the same time as Numeric-Compare. (This also confirms our intuition that users are better at interpreting phrases than numbers). Thus, if both devices are equipped with decent quality displays, Phrase-DD is a clear winner.

Using similar reasoning, Over-Audio and Phrase-DS appear to be the best choices if the audio channel can be utilized. However, Over-Audio needs a microphone on one device and a speaker on the other. Phrase-DS can also be used in scenarios where one device has a speaker. Although Phrase-SS is also error-free, it is relatively slow compared to Phrase-DS. Thus, there is no good motivation for Phrase-SS over Phrase-DD or Phrase-DS.

Although BEDA-Beep, BEDA-Blink and Beep-Blink have lower hardware requirements and work on devices with most basic interfaces, they take too long to complete. They usually require more than one trial to achieve successful pairing and Beep-Blink also yields very high fatal error rates. Considering that devices taking part in social scenarios have reasonably good interfaces, BEDA-Beep, BEDA-Blink and Beep-Blink can be safely ruled out, in favor of Phrase-DD or Phrase-DS.

Interpreting User Ratings

We now turn our attention to the graph in Figure 2, that summarizes user opinions collected via post-test questionnaires. Users are asked to rate six different statements for each method on a 6-point Likert scale ranging between "Strongly Disagree" and "Strongly Agree." The rated statements for each method were based on the criteria: {*Easy, Professional, Fun to Use, Tiring, Taking Too Long, Error Prone*}.

Not surprisingly, Numeric-Compare is ranked among the easiest methods due to its fast timing and familiarity to most users as it has already been deployed in many personal devices. As expected, Phrase-DD and Over-Audio also received very high ratings and are ranked among the easiest, most fun to use and professional methods. Numeric-Compare, Phrase-DD and Over-Audio are among user favorites.

Despite their poor timing and/or high error rates, Beep-Blink and Copy-Confirm are ranked surprisingly positively. Both methods were perceived as easy and error-resistant. Considering Copy-Confirm had a very high fatal error rate and rated as one of the least error-prone methods, we conclude that users who committed a fatal error were clearly not aware of it. Also, judging from the high error rates of both Copy-Confirm and Beep-Blink, users' perception of security may be far from reality. This contradiction can also be easily observed in Phrase-DS and Phrase-SS, although in the other direction. These two methods are ranked among the most error-prone, however, they yield 0% error-rate in our tests.

BEDA-Beep, BEDA-Blink, Phrase-DS and Phrase-SS are considered relatively hard, error prone, taking long time to complete, less professional and less fun to use. Relatively low user ratings for BEDA-Beep and BEDA-Blink agree with the long completion timings and high error rates observed in our tests. However, user perception was deceptive about Phrase-DS and Phrase-SS, especially, in terms of how error prone they are.

	Easy	Tiring	Professional	Long	Fun
Tiring	-0.343				
Professional	0.693	-0.266			
Long	-0.445	0.817	-0.293		
Fun	0.666	-0.318	0.737	-0.378	
Error-Prone	-0.425	0.722	-0.358	0.749	-0.361

$p < 0.01$, for all correlations

Table 2. Cross-Correlation of User Ratings

Observed Correlations: As can be seen from Figure 2, there is some observable, and statistically significant, correlation among user ratings for various usability measures. The cross correlation of user ratings for usability measures is given in Table 2. Correlation coefficients ranging from -0.3 to -0.1 and 0.1 to 0.3 are generally regarded as small, -0.5 to -0.3 and 0.3 to 0.5 as medium, and coefficients larger than 0.5 and smaller than -0.5 as high [4]. In line with observed high coefficients, our results show that methods rated easy are generally also rated as fun to use and professional.

Moreover, methods rated as taking too long were perceived as tiring and error-prone. Observed medium correlation coefficients also show that methods perceived to be easy, professional or fun to use are unlikely to be rated as error-prone, tiring or taking too long.

Principal Component and Cluster Analysis

Since most of our usability measures are correlated, looking at them as a whole is important. To this end, we performed principal component analysis for all usability measures.

Our analysis showed the first two components, i.e., PC1 and PC2, have eigenvalues more than 1, i.e., explaining more variance than one original variable [13], and they collectively explicate more than %64 of the variance. Factor loadings of PC1 and PC2 are given in Table 3.

PC1 factors positive usability measures (such as, user ratings for easiness, fun to use or professionalism; higher values indicating better usability) as negative and other measures as positive. Thus, lower PC1 scores for a method indicate better usability. On the other hand, PC2 factors in time and safe error rates much more than PC1. Thus, a very high PC2 score may be an indication of usability problems.

	PC1	PC2
Task Completion Time	0.088	0.484
Safe Error	0.033	0.461
"Easy" Rating	-0.422	0.216
"Error Prone" Rating	0.407	0.183
"Fun to use" Rating	-0.365	0.414
"Professional" Rating	-0.402	0.346
"Taking too long" Rating	0.442	0.262
"Tiring" Rating	0.396	0.336

Table 3. Factor Loadings of PC1 and PC2

Figure 3 shows mean values of PC1 and PC2 scores for all methods. Three observed clusters (using the Euclidian distance and average linkage method) based on principal components are also superimposed on *Figure 3*. The figure indicates that two methods, BEDA-Blink and Phrase-SS, are different from other methods and form their own clusters. This is mainly due to the significantly longer completion times of BEDA-Blink and lower user ratings for Phrase-SS. Overall, methods can be partitioned into two clusters, with good and poor usability. The former (shown with red colors in Figure 3) include Numeric-Compare, Over-Audio, Phrase-DD, Beep-Blink and BEDA-Beep. The rest (highlighted green, blue or orange in Figure 3) exhibit poor overall usability.

Among methods in the "good usability" group, BEDA-Beep takes at least twice as much time to complete and thus shows least similarity to all other methods in this group. Despite its good usability, Beep-Blink exhibits high fatal rates and suffers from robustness problems. Among the remaining three, Numeric-Compare had the highest "safe error" rate and Over-Audio took the longest.

Final Inferences and Recommendations

Our overall conclusions are as follows:

- Comparison-based pairing methods over the visual channel are preferred by users. Among those, we recommend Phrase-DD over Numeric-Compare as the former yields lower (in fact, no) errors. Since displays are ubiquitous on personal devices in social scenarios, Phrase-DD is a clear winner in terms of speed, robustness and user preference, as well as universal deployability.
- Among methods over the audio channel, Over-Audio was the user favorite. Phrase-DS yielded low completion timing and no errors while Phrase-SS was slow and error-free, however, both performed poorly compared to Over-Audio when all usability measures are taken into account. On the other hand, Over-Audio needs a microphone on one device and a speaker on the other. If devices lack required hardware for Over-Audio, we believe that Phrase-DS is still an acceptable choice.
- Beep-Blink and Copy-Confirm produced high fatal error rates in our tests and thus we do not recommend using them.
- BEDA-Beep and BEDA-Blink demonstrated poor completion time performance. They usually take more than one trial for successful completion and too long to complete. User ratings for these methods are also relatively low and we do not recommend using them.
- In general, we recommend Phrase-DD for social pairing. Phrase-DD can be complemented with Over-Audio or Phrase-DS, depending on available hardware.

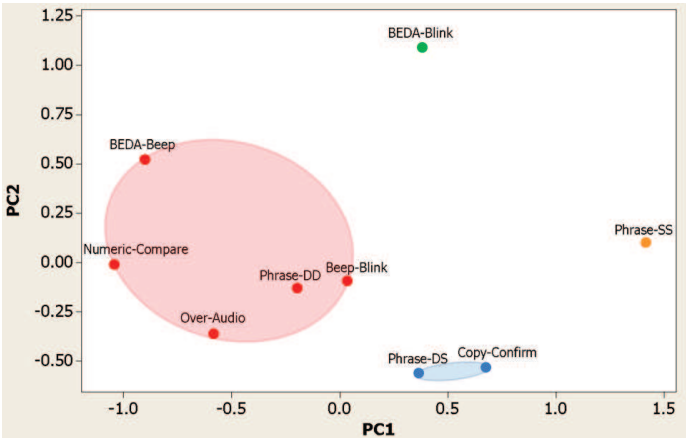

Figure 3. Method scorings and clusters based on PC1 and PC2

CONCLUSIONS

In this paper, we presented an experimental evaluation of prominent device pairing methods that can be used in social pairing scenarios. First and foremost, our survey results confirmed our belief that a majority of users are considerate about the security and privacy of their personal devices and they may not be willing to hand-in their devices to other users, even temporarily to perform the pairing process. This means that a social pairing method can not be simply reduced to personal pairing.

The results of our usability study show that one simple method, Phrase-DD, is quite attractive overall, being both fast and error-tolerant as well as user-friendly. It naturally appeals to social pairing scenarios where devices have appropriate quality and size displays. Slightly slower methods,

Over-Audio or Phrase-DS, can seamlessly inter-operate with Phrase-DD, for wider deployment and for scenarios where one device has a speaker. The fact that phrase comparison turned out to be the most suitable method for social pairing is in contrast to personal pairing where numeric comparison was a winner (as shown by the studies of [18, 16, 12]). We also observed that, in general, the test subjects often decided the outcome of social pairing based on mutual agreement, which, we believe, may have helped to reduce errors in most of our comparison-based pairing methods.

Acknowledgments: We thank A.J. Brush, Amy Karlson and Stuart Schechter for discussion related to their work on sharing of phones [14]. This work is supported in part by NSF Cybertrust grants #0831397 and #0831526.

REFERENCES

1. D. Balfanz et al. Talking to strangers: Authentication in ad-hoc wireless networks. In *Network and Distributed System Security Symposium (NDSS)*, 2002.
2. D. Balfanz et al. Network-in-a-box: How to set up a secure wireless network in under a minute. In *USENIX Security*, pages 207–222, 2004.
3. V. Boyko, P. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using diffie-heilman. *Eurocrypt*, 2000.
4. J. Cohen et al. *Applied multiple regression/correlation analysis for the behavioral sciences*. Erlbaum Hillsdale, NJ, 1983.
5. C. M. Ellison and S. Dohrmann. Public-key support for group collaboration. *ACM Transactions on Information and System Security (TISSEC)*, 6(4):547–565, 2003.
6. I. Goldberg. Visual key fingerprint code. `http://www.cs.berkeley.edu/iang/visprint.c`, 1996.
7. M. Goodrich et al. Loud and clear: Human-verifiable authentication based on audio. In *International Conference on Distributed Computing Systems (ICDCS)*, 2006.
8. L. Holmquist et al. Smart-its friends: A technique for users to easily establish connections between smart artifacts. In *Ubicomp*, 2001.
9. S.-J. Hong, K. Y. Tam, and J. Kim. Mobile data service fuels the desire for uniqueness. *Communications of the ACM*, 49(9), 2006.
10. I. Ion et al. Influence of user perception, security needs, and social factors on device pairing method choices. In *SOUPS: Symposium on Usable Privacy and Security*, 2010.
11. R. Kainda, I. Flechais, and A. Roscoe. Two heads are better than one: Security and usability of device associations in group scenarios. In *SOUPS: Symposium on Usable Privacy and Security*, pages 1–13, 2010.
12. R. Kainda, I. Flechais, and A. W. Roscoe. Usability and security of out-of-band channels in secure device pairing protocols. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.
13. H. Kaiser. The application of electronic computers to factor analysis. *Educational and psychological measurement*, 20(1):141–151, 1960.
14. A. K. Karlson, A. B. Brush, and S. Schechter. Can i borrow your phone?: understanding concerns when sharing mobile phones. In *CHI '09: Conference on Human factors in computing systems*, 2009.
15. T. Kindberg and K. Zhang. Validating and securing spontaneous associations between wireless devices. In *Information Security Conference (ISC)*, pages 44–53, 2003.
16. A. Kobsa, R. Sonawalla, G. Tsudik, E. Uzun, and Y. Wang. Serial hook-ups: A comparative usability study of secure device pairing methods. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.
17. K. Kostiainen and E. Uzun. Framework for comparative usability testing of distributed applications. In *Security User Studies: Methodologies and Best Practices Workshop*, 2007.
18. A. Kumar et al. Caveat emptor: A comparative study of secure device pairing methods. In *IEEE Pervasive Computing and Communications (PerCom)*, 2009.
19. S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. *Conference on Cryptology and Network Security (CANS)*, 2006.
20. R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 66:68–71, 1932.
21. Y. Liu, A. Rahmati, Y. Huang, H. Jang, L. Zhong, Y. Zhang, and S. Zhang. xshare: supporting impromptu sharing of mobile phones. In *MobiSys '09: Conference on Mobile systems, applications, and services*, 2009.
22. R. Mayrhofer and H. Gellersen. Shake well before use: Authentication based on accelerometer data. *Conference on Pervasive Computing (Pervasive)*, 2007.
23. R. Mayrhofer and M. Welch. A human-verifiable authentication protocol using visible laser light. In *IEEE Conference on Availability, Reliability and Security*, 2007.
24. J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *IEEE Symposium on Security and Privacy*, 2005.
25. R. Nithyanand et al. Groupthink: usability of secure group association for wireless devices. In *Ubicomp*, pages 331–340, 2010.
26. S. Pasini and S. Vaudenay. Sas-based authenticated key agreement. In *International Conference on Theory and Practice of Public-Key Cryptography (PKC)*, 2006.
27. A. Perrig and D. Song. Hash visualization: a new technique to improve real-world security. In *International Workshop on Cryptographic Techniques and E-Commerce*, 1999.
28. R. Prasad and N. Saxena. Efficient device pairing using "human-comparable" synchronized audiovisual patterns. In *Applied Cryptography and Network Security (ACNS)*, 2008.
29. V. Roth et al. Simple and effective defense against evil twin access points. In *ACM Conference on Wireless Network Security (WiSec)*, pages 220–235, 2008.
30. N. Saxena et al. Extended abstract: Secure device pairing based on a visual channel. In *IEEE Symposium on Security and Privacy*, 2006.
31. C. Soriente, G. Tsudik, and E. Uzun. Beda: Button-enabled device association. In *International Workshop on Security and Privacy in Spontaneous Interaction (IWSSI)*, 2007.
32. C. Soriente, G. Tsudik, and E. Uzun. Hapadep: Human-assisted pure audio device pairing. In *Information Security Conference (ISC)*, pages 385–400, 2008.
33. F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols Workshop*, 1999.
34. E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Financial Cryptography and Data Security*, pages 307–324.
35. S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *International Cryptology Conference (CRYPTO)*, 2005.