Proceedings of the Fifth Symposium On Usable Privacy and Security (SOUPS-2009), Mountain View, CA

Serial Hook-ups: A Comparative Usability Study of Secure Device Pairing Methods

Alfred Kobsa
Dept. of Informatics
University of California, Irvine
kobsa@uci.edu

Rahim Sonawalla
Dept. of Informatics
University of California, Irvine
rsonawal@uci.edu

Gene Tsudik
Dept. of Computer Science
University of California, Irvine
gts@ics.uci.edu

Ersin Uzun*
Dept. of Computer Science
University of California, Irvine
euzun@ics.uci.edu

Yang Wang
Dept. of Informatics
University of California, Irvine
yangwang@uci.edu

ABSTRACT

Secure Device Pairing is the bootstrapping of secure communication between two previously unassociated devices over a wireless channel. The human-imperceptible nature of wireless communication, lack of any prior security context, and absence of a common trust infrastructure open the door for *Man-in-the-Middle* (aka *Evil Twin*) attacks. A number of methods have been proposed to mitigate these attacks, each requiring user assistance in authenticating information exchanged over the wireless channel via some human-perceptible auxiliary channels, e.g., visual, acoustic or tactile.

In this paper, we present results of the first *comprehensive* and comparative study of eleven notable secure device pairing methods. Usability measures include: task performance times, ratings on System Usability Scale (SUS), task completion rates, and perceived security. Study subjects were controlled for age, gender and prior experience with device pairing. We present overall results and identify problematic methods for certain classes of users as well as methods best-suited for various device configurations.

1. INTRODUCTION

Wireless communication is very popular and is likely to remain so in the future. In particular, medium- and short-range wireless communication methods (such as Bluetooth, WiFi, Zigbee and WUSB) are becoming ubiquitous on personal devices, such as cell-phones, headsets, cameras and memory sticks. In the past, wireless devices communicated mostly with the (wired) infrastructure, e.g., cell-phones with base stations or laptops with access points. However, modern devices increasingly need to communicate among themselves, e.g., a Bluetooth headset with a cell-phone, a memory stick with a PDA, a PDA with a wireless printer, or a wireless access point with a laptop.

The convenience of seamless mobility and ubiquitous connectivity that comes with personal wireless devices is tempered by increased security and privacy risks. Compared to its wired counterpart, wireless communication is subject to easier eavesdropping and other attacks. Specifically, the process of setting up an initial security context between wireless devices is prone to so-called *Man-in-the-Middle* (MiTM) attacks, also known as *Evil Twin* attacks. Countering MiTM attacks requires the communication channel to be authenticated, which in turn requires either a pre-shared secret or a common trust infrastructure, neither of which exists. Secure device pairing therefore aims at authenticating communication channels by using a human-perceptible auxiliary channel.

The main challenge in secure device pairing stems from two factors: (1) inherent exposure to attacks and (2) human imperceptibility of wireless channels. Traditional cryptographic means of establishing secure communication (such as authenticated key exchange protocols) are unsuitable for the problem at hand, since the communication channel is not authenticated and unfamiliar devices have no prior security context or common point of trust. Among a multitude of device types and their manufacturers, there is no common security infrastructure and none is likely to materialize in the near future. This is due in part to the diversity of devices, lack of standards, and glacial progress of standardization bodies. However, there is wide-spread acceptance on the part of device manufacturers and the research community that some form of human user involvement in secure device pairing is unavoidable [32].

One natural and well-explored research direction aimed at addressing the problem is the use of auxiliary "out-of-band" (OOB) channels, which are both perceivable and manageable by the human user. An OOB channel takes advantage of human sensory capabilities to authenticate human-imperceptible information exchanged over the wireless channel (which is subject to MiTM attacks). OOB channels can be realized using acoustic, visual and tactile senses. The main idea is that a human-perceivable OOB channel, unlike the main wireless channel, exposes MiTM attacks to the user.

*Corresponding author.

Symposium on Usable Privacy and Security (SOUPS) 2009, July 15–17, 2009, Mountain View, CA USA
.

Since some degree of user involvement is unavoidable, usability becomes a crucial issue. Also, since a typical OOB channel is low-bandwidth, the amount of information transferred over it needs to be minimized for reasons of both usability and efficiency. Most pairing methods (see Section 2) involve sending only a few bits (e.g., 15) over the OOB channel to achieve reasonable security. At the same time, some devices (e.g., Bluetooth headsets and wireless access points) have very limited hardware capacities and poor or very rudimentary user interfaces, making it a challenge to communicate even a few bits.

In the last decade, many secure device pairing methods have been proposed, each claiming certain advantages and exhibiting certain shortcomings. As described in Section 2.2, their fundamental distinguishing characteristics concern the nature of the OOB channel and assumptions regarding the user interface and device features.

Even though some methods have been field-tested on their own (e.g, [29, 22]), no comprehensive and comparative *usability* evaluation of these methods has been carried out[1]. Very recently, Kumar, et al. [18] reported on the first study comparing a number of prominent methods. However, since it focused mainly on security, [18] has not yielded notable *usability* results.[2] Furthermore, the study in [18] has the following issues:

1. The set of participants was narrow, comprising mostly young (and male) graduate students very familiar with the newest technology.
2. Test administrators were also the developers of some of the tested methods.This undermines the perceived neutrality of the study.
3. Although the sequence of methods tested by each subject was random, it was not controlled for overall uniform distribution.
4. Each method was tested multiple times with different errors that simulated MiTM attacks. Although necessary for security measurements, this undermines the accuracy of usability evaluation, due to subject fatigue, uneven number of test cases and varying degrees of learning effect among methods.[3]

In general, we observe that many (if not most) secure device pairing methods have been developed by security researchers who, not surprisingly, are experts in security and not usability or HCI. What seems simple and user-friendly to a seasoned security professional might not be either to an average user. Non-specialist users are often initially clueless about manipulating new devices and have insufficient understanding of security issues and the very meaning of user participation in secure device pairing. This disconnect between developers and average users as well as aforementioned issues with [18], serve as the chief motivation for the study presented in this paper.

[1]Concurrent with and independent of our work, similar research has been conducted by Kainda, et al. [14].

[2]The only usability insight was obtained by asking participants to rank the perceived difficulty of tested methods, from easy to hard within each pre-defined group. Participants' post-hoc perception is only one of several usability indicators though.

[3]The large number of error scenarios also resulted in the study being broken into three batches, each batch separated by several days and taking place in a different environment.

Organization: Section 2, reviews prominent secure device pairing methods. Section 3 discusses our criteria for selecting candidate methods. The rest of the paper describes the design of our study (Section 4), presents its results (Section 5) and discusses its implications.

2. BACKGROUND

As a background, we now overview relevant cryptographic protocols and secure device pairing methods that use these protocols. (Those familiar with secure device pairing may wish to skip this section with no lack of continuity). The term *cryptographic protocol* denotes the entire interaction involved, and information exchanged, in the course of the pairing method. The term *pairing method* refers to the pairing process as viewed by the user, i.e., user actions. As discussed below, a particular cryptographic protocol can be realized using many pairing methods.

2.1 Cryptographic Protocols

A very simple protocol for device pairing was first suggested in [2]: devices A and B exchange their respective public keys pk_A and pk_B over the insecure channel, and the corresponding hashes $H(pk_A)$ and $H(pk_B)$ over the OOB channel. Although non-interactive, this protocol requires $H()$ to be a (weakly) collision-resistant hash function and thus needs at least 80 bits of OOB data in each direction. MANA protocols [9] reduce the size of OOB messages to k bits while limiting attacker's success probability to 2^{-k}. However, they impose a stronger assumption on the OOB channel: the adversary cannot delay or replay any OOB messages.

An alternative approach involves Short Authenticated Strings (SAS). The first SAS protocol was proposed in [34]. It limits attack probability to 2^{-k} for a k-bit OOB channel, even when the adversary can delay and/or replay OOB messages. This protocol uses commitment schemes (which can be based on hash functions such as SHA-2) and requires four rounds of communication over the wireless channel. Subsequent work [19, 24] yielded 3-round SAS protocols.[4] Generally, SAS protocols are used in device pairing settings where either: (1) the OOB channel is used to transmit something (i.e., the SAS itself) from one device to another, or (2) the user is asked to compare two values emitted by the respective devices.

Some pairing methods require the user to generate a secret random value and somehow enter it into both devices. Devices then perform *authenticated* key exchange, using the user-generated secret as a means of one-time authentication. Cryptographic protocols used for this purpose are called Password-Authenticated Key Exchange (PAKE) protocols [4].

2.2 Device Pairing Methods

Based on cryptographic protocols described above, a number of pairing methods have been proposed. They operate over different OOB channels and offer varying degrees of security and usability.

"Resurrecting Duckling" [31] is the initial attempt to address the device pairing problem in the presence of MiTM attacks. It requires standardized physical interfaces and ca-

[4]Recently, [27, 28] proposed even more efficient SAS protocols which are used in several pairing methods we studied.

bles. Though appropriate in the 1990s, it is clearly obsolete today, due to the greatly increased diversity of devices. Requiring physical equipment (i.e., a cable) also defeats the purpose of using wireless connections. Another early method is "Talking to Strangers" [2], where infrared (IR) communication is used as the OOB channel. It requires almost no user involvement, except for the initial setup. Unlike many other methods, it has been extensively tested [1]. This method is deceptively simple: since IR is line-of-sight, set-up requires the user to find IR ports on both devices – not a trivial task for many – and align them. Also, despite its line-of-sight property, IR is not completely immune to MiTM attacks. The main drawback is that IR has been largely displaced by other wireless technologies, such as Bluetooth, and is available on very few modern devices.

Another early 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" [10], "Random Arts Visual Hash" [25] and "Colorful Flag" [7]. Such methods require both devices to have displays with sufficiently high resolution. Their applicability is thus limited to high-end devices, such as laptops, PDAs and certain cell-phones.

In [22], McCune, et al. proposed the "Seeing-is-Believing" (SiB) method. In its original form, SiB requires a bidirectional visual OOB channel: each device, one after the other, 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 both devices to have a camera and a display for bidirectional authentication. Thus, it is unsuitable for lower-end devices. Our study includes an SiB variant from [27, 28] which only requires one device to have a camera. We refer to it as "See-Believe".

A related approach, called "Visual authentication based on Integrity Checking" (VIC) was explored in [27]. Like SiB, it uses the visual OOB channel and requires one device to have a continuous 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 by 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. We refer to this method as "Video".

[26] proposed several pairing methods based on synchronized audio-visual patterns: "Blink-Blink", "Beep-Beep" and "Beep-Blink". All of them 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 only require devices to have two LEDs or a basic speaker.

Another recent method is "Loud-and-Clear" (L&C) [11]. It uses the audio (acoustic) OOB channel along with vocalized MadLib sentences which represent the digest of information exchanged over the main wireless channel. There are two L&C variants: "Display-Speaker" and "Speaker-Speaker". In the latter, the user compares two vocalized sentences and in the former – a displayed sentence with its vocalized counterpart. Minimal device requirements include a speaker (or audio-out port) on one device and a speaker or a display on the other. The user is required to compare two respective (vocalized and/or displayed) MadLib sentences and either accept or abort the protocol based on the outcome of the comparison. In this paper, we use the L&C variant based on SAS protocols [24, 19] to reduce the number of words in the MadLib sentences. Depending on the required user interaction, we call the two L&C variants as "Listen-Look" and "Listen-Listen".

As a follow-on to L&C, HAPADEP [30] considered pairing devices that have no common wireless channel, at least not at pairing time. HAPADEP uses pure audio to transmit cryptographic protocol messages and asks the user to merely monitor device interaction for any extraneous audio interference. It requires both devices to have speakers and microphones. To cater to very basic device settings, we include a HAPADEP variant that uses the wireless channel for cryptographic protocol messages, and the audio as the OOB channel. We call this variant "Over-Audio". It needs only one device to be equipped with a speaker, and the other with a microphone. Also, on the user's part, it involves no entry of data and no comparisons.

A small-scale experimental investigation [33] presented the results of a comparative usability study of four simple pairing methods for devices with displays capable of showing a few (4-8) decimal digits:

Compare-and-Confirm: The user compares two (4-, 6- or 8-digit) numbers displayed by respective devices.

Select-and-Confirm: One device displays a single number. The other displays a set of numbers and user selects one that matches the number displayed by the first device.

Copy-and-Confirm: The user copies a number from one device to the other.

Choose-and-Enter: The user picks a "random" 4-to-8-digit number and enters it into both devices.

Though all these methods are very simple, [33] shows that Select-and-Confirm and Copy-and-Confirm are slow and error-prone. Furthermore, Choose-and-Enter is insecure, since studies show that numbers selected by users exhibit very poor randomness.

Yet another approach – BEDA [29] – involves the user pressing device buttons, thus utilizing the tactile OOB channel. BEDA has several variants: *LED-Button*, *Beep-Button*, *Vibration-Button* and *Button-Button*. In the first two (based on the SAS protocol [27]), whenever the sending device blinks its LED (or vibrates or beeps), the user presses a button on the receiving device. Each 3-bit block of the SAS string is encoded as the delay between consecutive blinks (or vibrations or beeps). Thus, repeated button presses transmit the SAS from one device to another. In the Button-Button variant – which works with any Password-Authenticated Key Exchange (PAKE) protocol [4] – the user simultaneously presses buttons on both devices and random user-controlled inter-button-press delays are used as a means of establishing a common secret. In this paper, we refer to BEDA variants *LED-Button*, *Beep-Button* and *Vibration-Button* as "LED-Press", "Beep-Press" and "Vibrate-Press", respectively.

There are other methods involving technologies that are currently expensive and/or uncommon on commodity devices. We briefly summarize a few. [15] suggested using ultrasound as the OOB channel. A related technique uses laser and requires each device to have a laser transceiver

	Device/Equipment Requirements		User Actions			
Pairing Method	**Sending Device**	**Receiving Device**	**Phase I: Setup**	**Phase II: Exchange**	**Phase III: Outcome**	**OOB Channels**
Visual Comparison Based ▪Image-Compare ▪PIN-Compare ▪Sentence-Compare	Display + user-input on both		**NONE**	Compare: ▪two images ▪two numbers ▪two phrases	Abort or accept on both devices	Visual
Seeing is Believing (SiB) ▪See-Believe	Display + user-input	Photo camera + user-output	**NONE**	Align camera on receiving device with displayed barcode on sending device, take picture	Abort or accept on sending device based on receiving device decision	Visual
Visual Integrity Code (VIC) ▪Video	LED + user-input	User-output + Light detector or video camera	**NONE**	Initiate transmittal of OOB data by sending device, align camera or light detector on receiving device.	Abort or accept on sending device based on receiving device decision	Visual
Loud & Clear (L&C) ▪Listen-Look ▪Listen-Listen	User-input on both + ▪display on one & speaker on the other, or ▪speaker on both		**NONE**	Compare: ▪two vocalizations ▪Displayed phrase with vocalization	Abort or accept on both devices	▪Acoustic, or ▪Acoustic+ visual
Button-Enabled (BEDA) ▪Vibrate-Press ▪LED-Press ▪Beep-Press	User input + ▪vibration ▪LED ▪beeper	User output + One button	Touch or hold both devices	For each signal (display, sound or vibration) by sending device, press a button on receiving device	Abort or accept on sending device based receiving device decision	▪Tactile ▪Visual + tactile ▪Acoustic+ tactile
Audio Pairing (HAPADEP) ▪Over-Audio	Speaker + user-input	Microphone + user-output	**NONE**	Wait for signal from receiving device.	Abort or accept on sending device	Acoustic
Resurrecting Duckling	Hardware port (e.g., USB) on and a cable		Connect cable to devices	**NONE**	**NONE**	Cable
Talking to Strangers	IR port on both		Find, activate, align IR ports	**NONE**	**NONE**	IR
Copy–and-Confirm	Display + user-input	Keypad + user-output	**NONE**	Enter value displayed by sending device into receiving device	Abort or accept on sending device based on receiving device decision	Visual
Choose-and-Enter	User input on both devices		**NONE**	Select "random" value and enter it into each device	**NONE** **(unless synch. Error)**	Tactile
Audio/Visual Synch. ▪Beep-Beep ▪Blink-Blink ▪Blink-Beep	User-input on both + ▪Beeper on each ▪LED on each ▪Beeper on one & LED on the other		**NONE**	Monitor synchronized: ▪beeping, or ▪blinking, or ▪beeping & blinking	Abort on both devices if no synchrony	▪Visual ▪ Audio ▪Audio + visual
Smart-its-Friends, Shake-Well-Before-Use	2-axis accelerometers on both + user-output on one		Hold both devices	Shake/twirl devices together, until output signal	**NONE** **(unless synch. error)**	Tactile + motion

Figure 1: Feature Summary of Notable Device Pairing Methods

[21]. In "Smart-Its-Friends" [13], a common movement pattern is used to communicate a shared secret to both devices as they are twirled and shaken together by the user. A similar approach is developed in "Shake Well Before Use" [20]. Both techniques require devices to have 2-axis accelerometers. Although some new cell-phones (e.g., the iPhone) are thus equipped, accelerometers are rare on most other devices. Also, physical shaking/twirling is an activity unsuitable for delicate as well as stationary or large/bulky devices.

Methods Summary

Figure 1 summarizes our discussion of existing methods by comparing their salient features. The following terminology is used:

Sending/Receiving Device: applies to all methods where the OOB channel is used in one direction.

Phase I: Setup: user actions to bootstrap the method.

Phase II: Exchange: user actions as part of the protocol.

Phase III: Outcome: user actions finalizing the method.

User-input: any means of user input, e.g., a button.

User-output: any user-perceivable means of output, e.g., an LED.

3. SELECTION OF PAIRING METHODS

As follows from the above overview, there is a large body of prior research on secure device pairing and many proposed methods. As shown in Figure 1, there are about 20 notable methods (counting variations). In the course of performing extensive pilot tests, we determined that only about half of all methods ought to be included in a within-subject study, mainly to avoid user fatigue. We therefore eliminated the following methods (at the bottom of Figure 1):

Resurrecting-Duckling: obsolete, requires cables.

Talking-to-Strangers: obsolete, IR ports are uncommon.

Copy-and-Confirm: performed poorly in prior evaluations due to high user error rate.

Choose-and-Enter: performed poorly in prior evaluations due to low security.

Simple Audio/Visual Synchronization (Beep-Beep, Blink-Blink, Beep-Blink): performed poorly in prior evaluations due to user annoyance and high error rate.

Smart-its-Friends, Shake-Well-Before-Use, Ultrasound- and Laser-based methods: require interfaces that are uncommon on many current types of devices.

All remaining methods were included in our study.

4. EXPERIMENTAL DESIGN

4.1 Apparatus

We used the two Nokia cell-phone models E51 and E61[5] as test devices. Both models have been released for at least two years and do not represent the cutting edge. We selected these particular models to avoid devices with exotic or expensive features and faster-than-average processors. Another reason for choosing these devices is the number of *commonly available* interfaces, such as:

User-input: keypad (subsumes button), microphone, video camera (subsumes photo camera)

User-output: vibration, speaker (subsumes beeper), color screen (subsumes LED)

Wireless: Bluetooth, Wi-Fi and cellular (GSM)

In all our tests, Bluetooth was used as the (human-imperceptible) wireless channel; it is both inexpensive and widely available. For methods that involve beeping, the cell-phone speaker is trivial to use as a beeper. Whenever a button is needed, one of the keypad keys is easily configured for that purpose. An LED is simulated with a small LED-like image glowing (alternating between light and dark) on the cell-phone screen.[6]

In comparative usability studies, meaningful and fair results can only be achieved if all methods are tested under similar conditions. In our case, the fair comparison basis is formed by using (1) the same test devices, (2) consistent GUI design practices (e.g., safe defaults), and (3) the same targeted level of security for all methods. We also automated timing and logging to minimize administrator errors and biases.

In order to have a unified test platform, our implementation of the eleven selected device pairing methods was based upon the open-source comparative usability testing framework developed by Kostiainen, et al. [17]. It provides basic communication primitives as well as automated logging and timing functionality. However, we still had to implement separate user interfaces and simulated functionality for all tested methods in JAVA-MIDP. For all methods, we kept SAS string length (and secret OOB string length in Button-Button) constant at 15 bits. It is well-known that this size provides a reasonable level of security [34].

As implemented on our test platforms, each device pairing method very closely approximates user experience with a real implementation. One difference is that our versions of tested methods omit initial rounds of the underlying cryptographic protocol over the (human-imperceptible) wireless channel. However, this omission is completely transparent to users.

The only methods noticeably different from their real-world implementations were Seeing-is-Believing and Video. Due to the difficulty of implementing image and video processing on cell-phones, we chose to simulate their operations.[7] Specifically, we saved the captured barcode image in Seeing-is-Believing (and the recorded video of blinking screen in Video) on the test device and *manually* analyzed later whether their quality was sufficient for image recognition. From the user's perspective, the only difference is that these pairing methods do not fail, which is not problematic since each user only tests these methods once. Also, execution times of these two methods were penalized by a few seconds in our tested implementations, since a system security notification popped up each time the camera was activated by our third-party testing software.

4.2 Subjects

The 22 study participants were adults, mainly from Irvine, California. Most of them were University of California students and staff. They were balanced by age group (eight were between 18 and 25, seven between 26 and 40, and seven were 41 and over), and also separately by gender (i.e., eleven from each gender).

4.3 Procedures

We conducted a within-subjects experiment, in which all participants were subject to the following procedures:

Background Questionnaire: Subjects were polled on age, gender, ownership of mobile device, experience with device pairing, and experience with different functionality offered by mobile devices, e.g., messaging, gaming, music.

Scenario Presentation: Subjects were asked to imagine that they had just bought a new cell-phone and that a store employee had already set up everything for them. When they returned home they wanted to pair their new cell-phone with their old one.

Experiment with pairing methods: Subjects sequentially performed the following procedures for each of the eleven tested methods.

1. They were given brief and simple instructions on the next pairing method, both textually on one of the devices and orally by the test administrator.

2. They tried pairing the devices with one of the tested methods to establish a connection, and thereby performed one of the following actions:

   Beep-Press: When device A beeps, the user presses a key on device B. The user then accepts or rejects the outcome on A based on the output (green or red LED) on B.

   LED-Press: When an LED on A turns ON, the user presses a key on B. The user then accepts or rejects the outcome on A based on the output (green or red LED) on B.

   Image-Compare: Both A and B display a visual pattern. The user compares the two patterns and decides whether they match and enters the decision into both devices.

[5]See `http://www.nokiausa.com/A4579382` and `europe.nokia.com/A4142101` for their respective specifications.

[6]Even though both tested cell-phones have LEDs, there are unfortunately no system calls to access them via Java MIDP.

[7]The current CMU implementation of Seeing-is-Believing is supported on Nokia models N70 [23] and 6620 [22] as receiving devices. The current Nokia implementation of Video is supported only on Nokia 6630 [16] as the receiving device. Since we wanted to perform our tests on the same devices throughout, neither implementation could be used. Moreover, porting existing implementations onto our devices was not viable since characteristics of cameras on these cell-phones are quite different and each performs its own adjustments to images and video, at the operating system level.

Listen-Listen: Both phones "vocalize" a 3-word sentence. The user decides whether the two sentences match and enters the decision into both devices.

Listen-Look: A displays a 3-word sentence, while B vocalizes a 3-word sentence. The user decides whether the sentences match and enter the decision into both devices.

PIN-Compare: Both A and B display a 5-digit number. The user decides whether the numbers are identical and enters the decision into both devices.

Sentence-Compare: Both A and B display a 3-word sentence. The user decides whether the sentences match and enters the decision into both devices.

Over-Audio: A transmits data over audio and B receives the transmission (records it). User confirms that no other nearby source emits audio during the process.

Seeing-is-Believing (See-Believe): With device A, the user takes a photo of a barcode displayed by B. Based on the output by A, the user either accepts or rejects the outcome on B.

Vibrate-Press: Whenever A vibrates, the user presses a key on B. The user then accepts or rejects the outcome on A based on the output (green or red LED) of B.

Video: With device A, the user takes a video clip of a blinking pattern displayed by B. Based on the output by A, the user either accepts or rejects the outcome on B.

To avoid order effects (particularly due to training and fatigue), the sequence of performing the eleven pairing methods tasks was counter-balanced using a Latin Square design.

3. **Subjective Perceptions:** Subjects completed the System Usability Scale (SUS) questionnaire [5], a widely used and highly reliable 10-item Likert scale that polls subjects' satisfaction with computer systems [3]. We used the original questions from [5], but replaced "system" with "method". Subjects also rated the perceived security of each method, on the same scale.

4. **Observable usability indicators:** The following measures of observable usability indicators were taken for each device pairing: task performance time, errors (if any), and task completion (i.e., whether or not a connection was established). Subjects were videotaped during the pairing process (but were told beforehand that only their hands and the devices would be captured).

5. **Qualitative post-test questionnaire and interview:** Subjects completed a brief questionnaire that asked them to name the three easiest and the three hardest methods. It also asked them to pick two methods they would like to see on their personal device and to indicate why (the options given were "easy", "secure" and "fun"). Subjects could explain orally if they preferred a method that was not tested.

For each subject, the entire experiment lasted between 30 and 45 minutes.

5. RESULTS

As described above, the following measures were collected before and during the experiment, which form the within-subjects usability measures and between-subjects factors of our study:

Within-subjects usability measures: task performance time, SUS score, perceived security, and task completion (a categorical variable).

Between-subjects factors: age group, gender, and prior experience with device pairing.

Unless indicated otherwise, statistical significance will be reported at the 5% level (flagged as "*" or "significant", or "$<$" "or $>$" in comparisons), and at the 1% level (flagged as "**" or "highly significant", or "$\ll$" or "$\gg$" in comparisons).

5.1 Cross-correlation of usability measures

It is a common assumption in HCI that usability measures are typically not independent of each other, but rather correlated. For instance, user satisfaction is assumed to be negatively correlated to some degree with task performance times. A broad meta-study by Frøjær et al. [8] challenges this view though. The authors recommend that "unless domain specific studies suggest otherwise, effectiveness, efficiency, and satisfaction should be considered independent aspects of usability and all be included in usability testing."

We therefore performed linear cross-correlations of the four usability measures. Table 1 shows the correlation coefficients and their statistical significance.

	Task performance time	SUS	Perceived security
SUS	-0.383**	-	
Perceived Security	-0.211**	0.512**	-
Task completion	-0.248**	0.126	0.039

Table 1: Cross-Correlation of Usability Measures

In the Social Sciences, coefficients from -0.3 to -0.1 and 0.1 to 0.3 are generally regarded as small, and coefficients between -0.5 to -0.3 and 0.3 to 0.5 as medium [6]. In line with the findings of [8], we cannot regard any of our usability measures as sufficiently correlated with others that they could be justifiably omitted. On the other hand, since the measures *are* lowly correlated, it does make sense to also look at them as a whole. In the following, we are going to present an analysis of each usability measure individually, and thereafter perform a cluster analysis based on a principal component analysis globally for all usability measures.

5.2 Individual Usability Measures

A Repeated Measures Analysis of Variance was performed for each usability measure except for task completion (which is categorical). The between-subjects factors are age group, gender, and prior experience with device pairing, while the within-subjects factor is method. Pairwise (unpaired one-tailed) t-tests were performed between different levels of the between-subjects factors on each within-subjects measure, except for task completion that was subject to a Chi-square

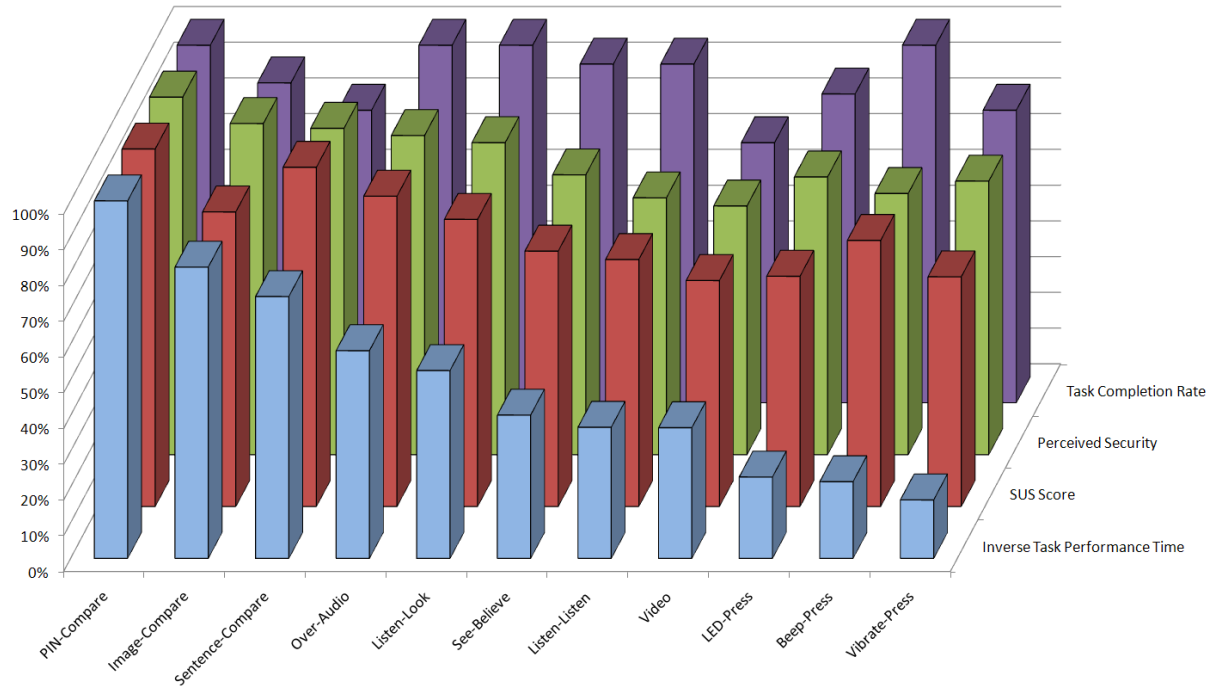

Figure 2: Effects of method on the four usability measures

test instead. Pairwise (paired one-tailed) t-tests between different methods were performed on each within-subjects measure. Pairwise (one-tailed) McNemar's Chi-squared tests were performed between different methods on task completion.

5.2.1 *Overview*

Figure 2 shows the averages of these usability measures for each method, normalized by the maximum average for the respective measure which equals 100%. Coincidentally, PIN-Compare fares best and thus represents 100% along *all* usability measures. To improve the visibility of the three-dimensional bar chart and to consistently associate greater height with "good", the *inverses* of the task performances times are plotted in the front layer. Again to improve visibility, the methods are sorted from left to right by inverse task performance time. This order should however not be interpreted as an overall ranking (except for PIN-Compare that tops all four measures).

A first impression from Figure 2 is that a few methods rank equally along all four usability measures, whereas considerable differences along the different measures exist for all other methods. This suggests that a ranking of methods should consider all four usability measures rather than only a single one. Section 5.3 will present such a ranking based on a Principal Component Analysis of all four measures. Another observation from Figure 2 is that the task performance times of the different methods vary considerably (ranging from an average of 15.7 sec for PIN-Compare to 93.4 sec for Vibrate-Press), while the other measures show a far lower variability. This suggests that more attention should be paid to the task performance times than to the other measures, since it is a very distinguishing characteristic.

5.2.2 *Effect of Method*

Repeated measures analysis of variance revealed that method has a highly significant effect on task performance time, SUS score, and perceived security ($p \ll 0.001$ in all cases). Below we will discuss the effects on these individual usability measures in more detail.

Task performance time: The following differences between the means of the pairing methods were statistically significant, including their transitive hulls (19 pairwise comparisons were made to arrive at these results)[8]:

PIN-Compare $\ll$ Image-Compare $<$ Listen-Look
PIN-Compare $<$ Sentence-Compare $<$ Over-Audio
Over-Audio $<$ Listen-Listen $<$ LED-Press
Listen-Look $\ll$ See-Believe $<$ Video-Compare
Listen-Look $<$ Listen-Listen $<$ LED-Press
Video-Compare $<$ LED-Press $<$ Beep-Press
Video-Compare $<$ Vibrate-Press

Particularly noteworthy is the high significance of the difference between the means of two fastest methods: PIN-Compare and Sentence-Compare.

[8]In the case of multiple comparisons, researchers can assign an acceptable type I error α (false positives) either to each individual comparison or jointly across all comparisons. Hochberg and Tamhane [12] advise that if "inferences are unrelated in terms of their content or intended use (although they may be statistically dependent), then they should be treated separately and not jointly." This position has been adopted in this study. To judge the results along a joint α, a significance level α/n can be chosen for each test, with n being the number of pairwise comparisons performed (Bonferroni correction).

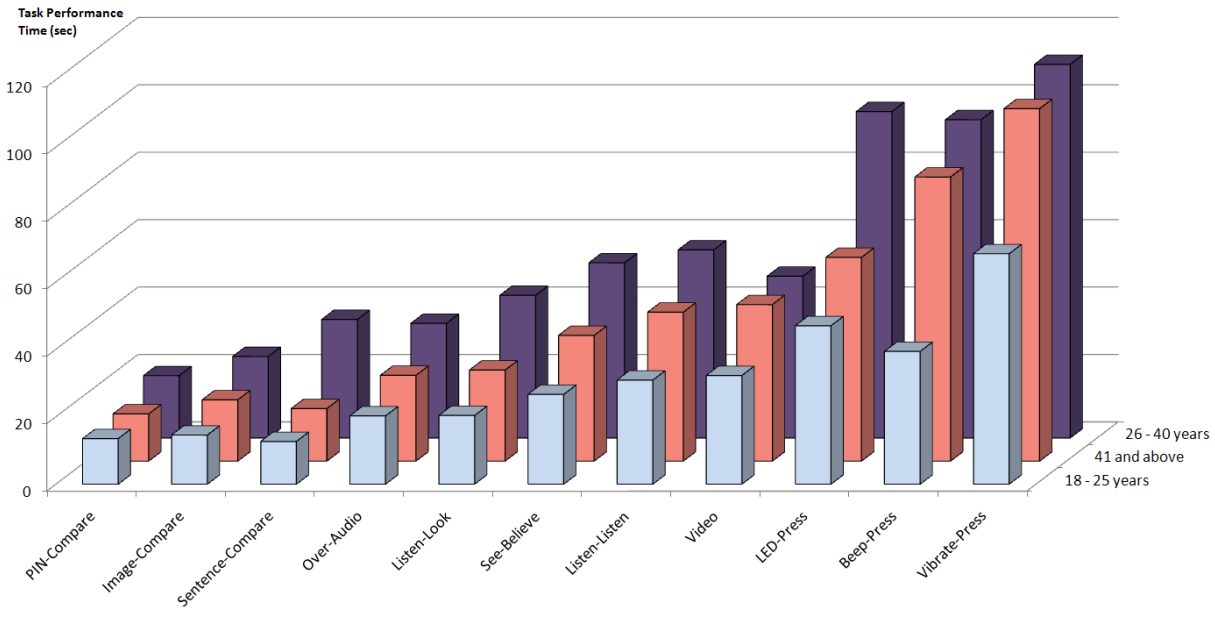

Figure 3: Average task performance time by age group

SUS score: The following significant differences between means were observed, including their transitive hulls (22 pairwise comparisons were performed to arrive at these results):

PIN-Compare > Sentence-Compare > Over-Audio
Sentence-Compare ≫ Image-Compare > See-Believe
Over-Audio > Listen-Look > Listen-Listen
Listen-Look > LED-Press, Vibrate-Press
See-Believe > Video-Compare > Beep-Press

Perceived Security: The following groups of methods can be distinguished, in decreasing order of perceived security (40 pairwise comparisons were made to arrive at these results):

1. PIN-Compare
2. Sentence-Compare, Over-Audio, Listen-Look
3. See-Believe, LED-Press, Vibrate-Press, Beep-Press, Listen-Listen, Video

Differences in means between these groups were found to be (highly) significant, while difference in means within the groups were not.

Task completion: Task completion rates were generally very high, and no noteworthy statistically significant pairwise difference could be found.

5.2.3 *Effect of Age*

As one would expect, the 18-25 year age group exhibited the shortest task performance time (30.6 sec on average). Surprisingly, though, it was not the oldest age group but rather the middle age group that had the longest task performance time (26-40 years: 57.1 sec; 41 and above: 44.2 sec).[9] Figure 3 shows the task performance times of the different pairing methods listed in the same sequence as in Figure 2 (in contrast to this prior figure though, the task performance times and not their inverses are plotted, and hence short height is "good"). The youngest age group can be seen in the front plane, the oldest age group in the middle plane, and the middle age group in the rear plane. The time differences between ages were particularly stark for the following methods:

Beep-Press: means of age group 18-25: 39.4 sec; means of pooled age groups 26-40 and 41 and above: 89.4 sec (the difference is highly significant).

Vibrate-Press: means of age group 18-25: 68.4 sec; means of pooled age groups 26-40 and 41 and above: 107.7 sec (the difference is approaching significance, p=0.089).

LED-Press: means of pooled age groups 18-25 and 41 and above:[10] 53.3 sec; means of age group 26-40: 96.9 sec (the difference is significant).

Different age groups also had somewhat different task completion rates: 94.4% on average for age group 18-25, 88.7% for the age group 26-40, and 86.3% for the age group 41 and above. The difference between the young and the old age group is significant, and the difference between the young and middle age group approaches significance (p= 0.074). Different age groups also exhibit differences with regard to perceived security. The security rating of the 26-40 year olds (7.1 out of 10 on average) is significantly higher than that of the 41 and above group (5.9 out of 10). The 18-25 year olds perceived the security somewhat in between (6.3 out of 10).

[9]The difference between the young and the middle group is significant at the p≪0.001 level, and the difference between the young and the old group significant at the p=0.019 level.

[10]Note that this is a pool of all participants but for the middle age group.

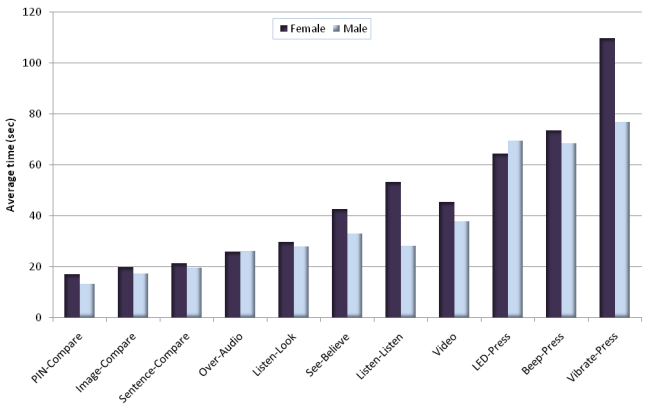

Figure 4: Average task performance time by gender

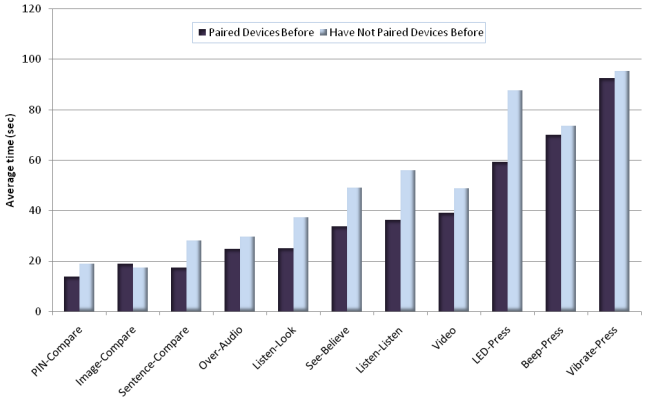

Figure 5: Average task performance time by experience

5.2.4 *Effect of Gender*

Males generally assigned higher SUS scores than females (average for males = 76.5, average for females = 66.07, $p < 0.001$), and also perceived the security of the pairing methods higher than females (average for males = 7.0, average for females = 5.9, $p < 0.001$). Figure 4 shows that females were generally also somewhat slower than males (average for males = 39.4 sec, average for females = 47.1 sec, p=0.07). The differences in time were particularly stark for Listen-Listen (average for males = 28.4 sec, average for females = 53.5 sec, significant) and for Vibrate-Press (average for males = 76.9 sec, average for females = 109.9 sec, not significant).

5.2.5 *Effect of Experience*

Figure 5 shows the average task performance time per method, split by subjects who had prior experience with device pairing and those who had not. No overall significant effect of experience could be observed. This can probably be regarded as an indicator that the pairing methods were easy enough and the experimental instructions effective enough that all subjects attained roughly the same skill level, independent of prior experience. The sizable task performance time difference for LED-Press was not statistically significant due to enormous within-group variability.

5.3 Cluster Analysis

A cluster analysis based on principal components was performed to determine methods that are closely related with regard on our usability measures. Table 2 lists the four principal components that explain 100% of the variance in the data. The first component PC1 explains nearly 75% of the variance, and the second component adds just 16.6% to this.

	PC1	PC2	PC3	PC4
Standard deviation	1.7292	0.8137	0.5307	0.2575
Proportion of Variance	0.7475	0.1655	0.0704	0.0166
Cumulative Proportion	0.7475	0.9130	0.9834	1.0000

Table 2: Principle Components of Usability Measures

For all practical purposes, we may thus disregard PC2 through PC4 since they contribute little. Table 3 shows the factor loadings of PC1. Not surprisingly, task performance time loads negatively while all other factors show a positive loading.

	x
Task performance time	-0.47
SUS	0.56
Perceived security	0.53
Task completion	0.42

Table 3: Factor Loadings of PC1

Figure 6 shows the result of a cluster analysis on principal components. Three clusters with six, two and three connection methods, respectively, can be distinguished (an alternative two-cluster solution would have merged clusters 2 and 3, but it makes less sense conceptually). Since Component 2 can be largely disregarded, methods towards the left side of Figure 6 can be regarded as "good" and methods towards the right as "bad".

5.4 Post-experimental ranking of easiest and hardest methods

As part of the exit questionnaire, subjects were asked to rank-order the three easiest and the three hardest methods in their view. The rank-order average across all subjects on a 1 (easiest) to 11 (hardest) scale can be seen in Table 4.

Easiest	1. PIN-Compare	1.8
	2. Sentence-Compare	2.1
	3. Over-Audio	2.9
↓	.....	...
	9. Seeing-is-Believing	9.2
	10. LED-Press	9.6
Hardest	11. Video	10.3

Table 4: Post-Experimental Ranking of Easiest and Hardest Methods

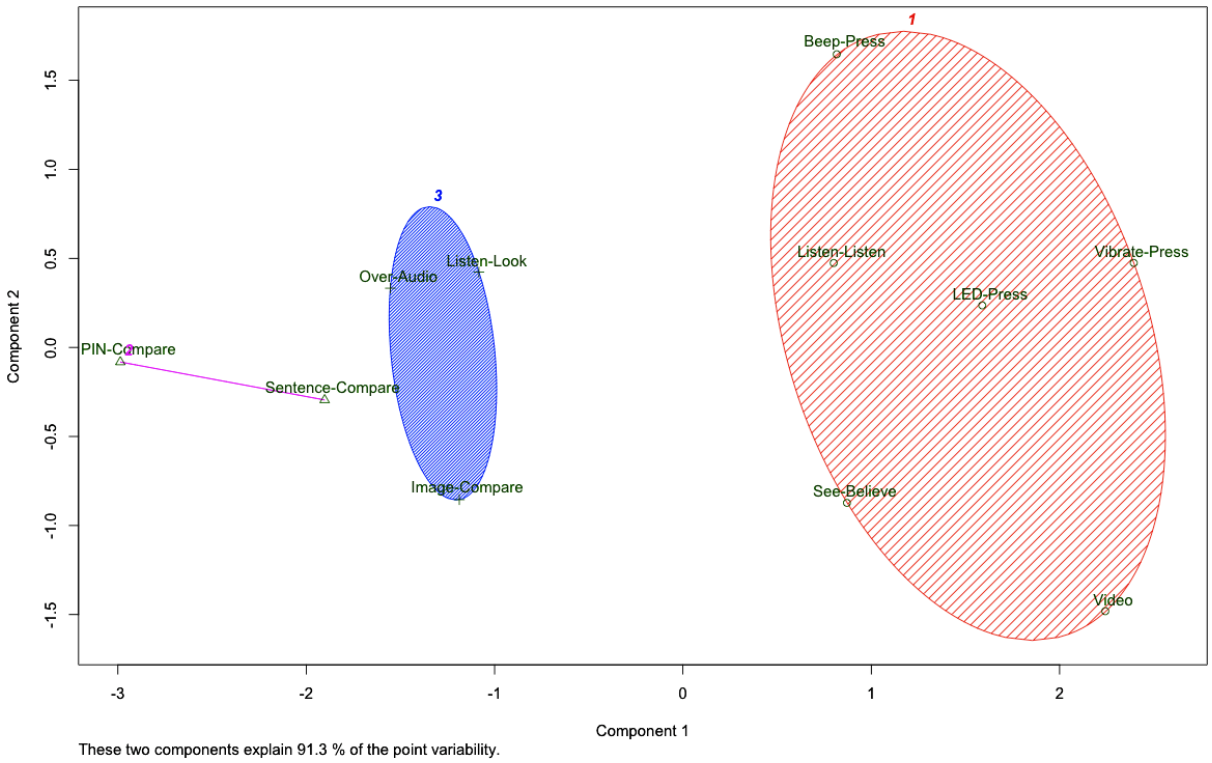

Figure 6: Result of cluster analysis based on principal components

6. DISCUSSION

As shown in Table 1 further above, several highly significant correlations can be observed between our usability measures ($p<0.001$). Particularly noteworthy is the medium-strong positive correlation between perceived security and SUS score. Few participants had a technical background sufficient to objectively assess the security of different device pairing methods. It seems that they partly relied on their usability rating of the methods instead.

We believe that Figure 6 is the clearest representation of our study's overall results. In it, the two methods in Cluster 2 (PIN- and Sentence-Compare) perform best overall, and the three methods in Cluster 3 (Over-Audio, Image-Compare and Listen-Look) come in as close second. However, viewed in isolation, PIN-Compare stands out against all others.

The main common feature of all methods in the two top clusters is that each requires a user comparison and a decision based on presented visual information, except Over-Audio which does not present any information to the user and is thus least taxing. In the remaining cases, the comparison is only between visual information, except for Listen-Look where it is between visual and audio information. As expected, the comparison of limited visual information (short PINs in PIN-Compare) ranks higher in usability than methods that require comparing more extensive visual information (Sentence- and Image-Compare[11]). In contrast, methods in the lower-ranked Cluster 1 require users to perform manual actions (press buttons, take pictures or video clips) or to listen to two successively spoken sentences, which is generally more taxing. Such added requirements seem to have a negative effect on the usability measures in our study.

It is heartening that subjects' post-experimental ranking of the easiest and hardiest methods (see Table 4) matches exactly the ranking along the first principal component of our usability measures (see Figure 6). Subjects reported that ease-of-use is by far the most important reason for them to favor a method. They praised methods which involved "no guesswork," and they liked "comparisons that require little effort". Only a few participants listed perceived security as a preference criterion (mostly in tandem with "ease"), and only one person cited "fun".[12]

One direct and practical consequence of our cluster analysis is the following set of design guidelines, based on the capabilities of the two devices involved:

1. If both devices have (even rudimentary) displays, the advisable methods are, in order of preference: PIN-Compare, Sentence-Compare and Image-Compare.

2. If only one device has a display, and the other audio output, then Listen-Look is the best choice.

3. If neither device has a display, but one has audio output and the other audio input (microphone), then Over-Audio is recommended.

These guidelines can help manufacturers to implement methods best-suited for specific pairs of devices. On a powerful

[11]The images generated by the Image-Compare method are patterns and do not contain recognizable objects.

[12]This might be interpreted as indicating that tasks involving security should not be perceived as "fun".

mobile device (e.g., a high-end cell-phone or a PDA) with rich I/O (and user) interfaces, the guidelines can also be used to determine the optimal pairing method based on the capabilities of the other device.[13]

A practical consequence of our study of between-subjects differences is the following set of population-specific guidelines:

- Listen-Listen should be avoided, particularly for female users, who took nearly twice as long to use this methods compared with males.
- Beep-Press is suitable for the younger age group only (if at all), since the other age groups took more than twice as long.
- LED-Press should be avoided, particularly for the middle age group, since this group took nearly twice as long as other age groups.

7. CONCLUSIONS AND FUTURE WORK

The study described in this paper sheds some much needed light on usability factors of many secure device pairing methods. Our experiment yielded numerous interesting results. In particular, the study clearly points to some methods that should be avoided altogether and several others (especially, those based on visual comparisons) that are well-suited for most users. It helps spot methods that are not well-suited for certain subgroups of the user population with regard to age, gender, and possibly also prior experience with device pairing. It also helps identify methods best-suited for settings where one or both device(s) lack displays.

However, there remain a number of issues for future work, such as:

- Since each secure device pairing method aims to protect the user(s) against MiTM attacks, a comparative evaluation of all methods under such attacks needs to be performed. Individual methods vary in terms of fragility and specifics of applicable attacks.
- On a related note, it would be useful to investigate various pairing methods in non-ideal settings, i.e., when the environment is not conducive to a specific method. Examples include performing visual comparisons with insufficient light, or using the audio channel in the presence of ambient noise.
- Our study was conducted with the population of healthy (physically unimpaired) adults. It would be valuable to perform similar studies with handicapped users, e.g., vision- or hearing-impaired as well as those with limited manual dexterity.
- All methods in our study were new to the participants. However, the effect of learning over time may significantly change the security and usability results. A long-term study is needed to investigate this effect.
- Finally, our study only considered a situation where *one* user pairs two devices. When two users need to pair their respective devices, the setting changes and a separate effort must be made to evaluate usability factors of various methods.

[13]To do so, either the user would have to be involved, or devices would exchange their respective capabilities over the wireless channel.

8. ACKNOWLEDGMENTS

This research was supported by NSF grant CNS-0831526. We thank Stacey Arnold and Nazia Chorwadwala for their help in conducting usability experiments, and Sameer Patil for his assistance in evaluating the results. Finally, we are very grateful to the participants who made this study possible.

9. REFERENCES

[1] D. Balfanz, G. Durfee, R. Grinter, D. Smetters, and P. Stewart. Network-in-a-Box: how to set up a secure wireless network in under a minute. In *USENIX Security*, pages 207–222, 2004.

[2] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Network and Distributed System Security Symposium (NDSS)*, 2002.

[3] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008. DOI 10.1080/10447310802205776.

[4] V. Boyko, P. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using diffie-hellman. In *Advances in Cryptology-Eurocrypt*, pages 156–171. Springer, 2000.

[5] J. Brooke. SUS: a "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, editors, *Usability Evaluation in Industry*. Taylor and Francis, London, 1996.

[6] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

[7] 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.

[8] E. Frøkjær, M. Hertzum, and K. Hornbæk. Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 345–352, 2000.

[9] C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *RSA CryptoBytes*, 7(1):29–37, 2004.

[10] I. Goldberg. Visual key fingerprint code. `http://www.cs.berkeley.edu/iang/visprint.c`, 1996.

[11] 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*, page 10, 2006.

[12] Y. Hochberg and A. C. Tamhane. *Multiple Comparison Procedures*. Wiley, New York, 1987.

[13] L. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Ubiquitous Computing (UbiComp)*, pages 116–122, London, UK, 2001. Springer-Verlag.

[14] R. Kainda, I. Flechais, and A. W. Roscoe. Usability and security of out-of-band channels in secure device pairing protocols. In *2009 Symposium On Usable Privacy and Security (SOUPS)*, Mountain View, CA (this volume), 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] K. Kostiainen. Personal Communication, Mar 2008.

[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, N. Saxena, G. Tsudik, and E. Uzun. Caveat Emptor: A Comparative Study of Secure Device Pairing Methods. In *IEEE International Conference on Pervasive Computing and Communications (IEEE PerCom'09)*, 2009.

[19] S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. In *International Conference on Cryptology and Network Security (CANS)*, volume 4301, pages 90–107, 2006.

[20] R. Mayrhofer and H. Gellersen. Shake well before use: Authentication based on accelerometer data. In *Pervasive Computing (PERVASIVE)*, pages 144–161.

[21] R. Mayrhofer and M. Welch. A human-verifiable authentication protocol using visible laser light. In *International Conference on Availability, Reliability and Security (ARES)*, pages 1143–1148, 2007.

[22] J. McCune, A. Perrig, and M. Reiter. Seeing-Is-Believing: using camera phones for human-verifiable authentication. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 110–124, 2005.

[23] J. M. McCune. Personal Communication, Mar 2008.

[24] S. Pasini and S. Vaudenay. SAS-Based Authenticated Key Agreement. In *Public key cryptography-PKC 2006: 9th International Conference on Theory And Practice in Public-Key Cryptography*, pages 395–409, 2006.

[25] 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.

[26] R. Prasad and N. Saxena. Efficient device pairing using "human-comparable" synchronized audiovisual patterns. In *Conference on Applied Cryptography and Network Security (ACNS)*, pages 328–345, 2008.

[27] N. Saxena, J. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *2006 IEEE Symposium on Security and Privacy*, pages 306–313, 2006.

[28] N. Saxena and M. B. Uddin. Automated device pairing for asymmetric pairing scenarios. In *Information and Communications Security (ICICS)*, pages 311–327, 2008.

[29] C. Soriente, G. Tsudik, and E. Uzun. BEDA: button-enabled device association. In *UbiComp Workshop Proceedings: International Workshop on Security for Spontaneous Interaction (IWSSI)*, 2007.

[30] C. Soriente, G. Tsudik, and E. Uzun. HAPADEP: human-assisted pure audio device pairing. In *Information Security*, pages 385–400, 2008.

[31] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols Workshop*, 1999.

[32] J. Suomalainen, J. Valkonen, and N. Asokan. Security associations in personal networks: A comparative analysis. In F. Stajano, C. Meadows, S. Capkun, and T. Moore, editors, *Security and Privacy in Ad-hoc and Sensor Networks Workshop (ESAS)*, pages 43–57, 2007.

[33] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Financial Cryptography and Data Security (FC'07) & Usable Security (USEC'07)*, pages 307–324, 2007.

[34] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Advances in Cryptology-CRYPTO*, pages 309–326, 2005.