Usability and Security of Out-Of-Band Channels in Secure Device Pairing Protocols

Ronald Kainda
Oxford University Computing Laboratory
Parks Road, OX1 3QD, UK
ronald.kainda@comlab.ox.ac.uk

Ivan Flechais
Oxford University Computing Laboratory
Parks Road, OX1 3QD, UK
ivan.flechais@comlab.ox.ac.uk

A.W. Roscoe
Oxford University Computing Laboratory
Parks Road, OX1 3QD, UK
bill.roscoe@comlab.ox.ac.uk

ABSTRACT

Initiating and bootstrapping secure, yet low-cost, *ad-hoc* transactions is an important challenge that needs to be overcome if the promise of mobile and pervasive computing is to be fulfilled. For example, mobile payment applications would benefit from the ability to pair devices securely without resorting to conventional mechanisms such as shared secrets, a Public Key Infrastructure (PKI), or trusted third parties. A number of methods have been proposed for doing this based on the use of a secondary out-of-band (OOB) channel that either authenticates information passed over the normal communication channel or otherwise establishes an authenticated shared secret which can be used for subsequent secure communication. A key element of the success of these methods is dependent on the performance and effectiveness of the OOB channel, which usually depends on people performing certain critical tasks correctly.

In this paper, we present the results of a comparative usability study on methods that propose using humans to implement the OOB channel and argue that most of these proposals fail to take into account factors that may seriously harm the security and usability of a protocol. Our work builds on previous research in the usability of pairing methods and the accompanying recommendations for designing user interfaces that minimise human mistakes. Our findings show that the traditional methods of comparing and typing short strings into mobile devices are still preferable despite claims that new methods are more usable and secure, and that user interface design alone is not sufficient in mitigating human mistakes in OOB channels.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*Human Factors*

General Terms

Experimentation, Security, Human Factors

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

Keywords

Security Protocols, Usability, pairing devices

1. INTRODUCTION

Secure systems have a habit of being broken by their own users. Whether this is deliberate (like insider attacks) or accidental (such as social engineering, innocent mistakes or unusable security mechanisms), it is now commonly accepted that *"people are the weakest link in the security chain"* [26].

This is of particular importance in the field of mobile computing. The growing power and connectivity of mobile devices – coupled with their near ubiquity – makes them both a threat and a valuable asset. Companies are increasingly becoming aware of the threats these devices pose to their internal networks (such as virus infection, trojans or other malware) [11]. But these devices are also valuable – they frequently hold personal, financial and private information – and are thus in need of protection. The trouble is that these devices tend to be managed and administered by untrained users, who do not understand the threats or necessarily care about security.

One particular area of mobile device security concerns key exchange in the secure pairing of mobile devices, which has recently attracted a lot of attention from both researchers and practioners. Because of the security weaknesses of the Bluetooth protocol for device pairing [10], new protocols [2, 4, 7, 18, 28, 32] have been proposed which potentially may replace the Bluetooth protocol or be used in scenarios where greater security is required.

Many of these protocols propose the use of an out-of-band (OOB) channel and expect humans to implement the channel. In this paper, we will refer to such protocols as *Human-Interactive Security Protocols* (HISPs).

The main characteristics of the OOB channel are that it has low bandwidth and is not vulnerable to typical Man-in-The-Middle (MiTM) attacks. One interesting example of the OOB channel consists of direct human communication, which naturally allows certain levels of trust to be established among the communicants. With the right security protocol, this trust can be transferred to devices that belong to the users – enabling two devices to establish a trusted communication that reflects the existing trust their users place on one another.

Proposals on how humans could implement the OOB channel include direct comparison of small pieces of information, transferring one piece of information from one device to another, and using an auxiliary device. These types of information include for example numeric combinations, alphanumeric strings, words, and sentences. An example of an auxiliary device is a camera phone that can be used to capture a picture of a barcode displayed on another device, allowing the camera phone to determine whether the barcode is correct. Each of the proposed methods aims to improve the usability of HISPs while maintaining a high level of security.

A closer examination of these methods is important in establishing which methods provide the best security and usability tradeoff. In order to investigate this, we conducted a study that builds on the findings of Uzun et al. in their *Usability Analysis of Secure Pairing Methods* [31]. While their research focused on basic methods for comparing and typing short numeric strings into mobile devices, a number of new pairing methods have since been proposed that claim to provide additional usability. In order to provide a comparable baseline for all these methods, both the basic methods and the newer ones were included in our own study. The methods included in our study consist of:

- comparing and confirming a match for images, melodies, sound, alphanumeric strings, sentences, and country/city names,
- matching a piece of information displayed on one device with the correct item in a list displayed on another device,
- typing alphanumeric strings from one device into another,
- using one device to take pictures of 2D barcodes displayed on another device.

In this paper, we present the results of the study and argue that many of the proposed methods fail to take into account factors that undermine the effective security of the protocol when used by humans. The results show that currently proposed methods are either subject to fatal security failures (e.g comparing and confirming a match is susceptible to a user mistakenly confirming a match), restricted to devices with specific capabilities (e.g. cameras are required for the barcode method), or restricted to specific usage scenarios (e.g. devices in close physical proximity, or pairing only two devices). In terms of relative usability, comparing strings ranked first, followed by typing strings, then comparing and selecting strings, and last was the barcode. When looking at the combined usability and security tradeoff, typing strings ranked first, followed by barcode, comparing and confirming and lastly comparing and selecting. In addition, based on our findings, we propose a number of factors that are crucial to designing secure and usable OOB channels that satisfy human and contextual needs.

The paper is organised as follows: in Section 2, we present background and related work; in Section 3, we discuss the experimental design and present the results in Section 4. We analyse and discuss the results in Section 5 and conclude with Section 6.

2. BACKGROUND AND RELATED WORK

In this section, we present a background to Human-Interactive Security Protocols and discuss challenges to their usability and security. To put the discussion of these challenges into context, we use the Symmetrised Hash Commitment Before Knowledge (SHCBK) protocol [18]. We also discuss some of the related work.

2.1 Human-interactive security protocols

A graphical representation of HISPs is shown in Figure 1, where N is the normal Dolev-Yao channel [5], which refers to an environment in which an attacker can overhear, delete, or modify messages while OOB is the low-bandwidth out-of-band channel where an attacker has no control over the messages but can overhear them.

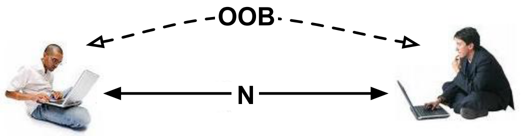

Figure 1: Graphical representation of HISP

The major strength of the OOB channel is authentication and message integrity, as opposed to secrecy. In other words, two people interacting over the OOB channel (such as a face to face discussion) have good assurances of each other's identity and obvious guarantees that their conversation is not modified or otherwise tampered with, but no assurance that they are not being overheard. Thus, users can exchange messages over the OOB channel by conversation in the presence of others who are not part of the protocol (that might include a potential attacker) but the latter have no control over these messages as they cannot block or modify them. The authenticity and integrity of the messages exchanged over the normal channel is verified by the message exchanged over the OOB channel.

The messages exchanged on the normal channel are public and can include public keys that may be long term or ephemeral. For example, devices can exchange RSA public keys together with other data using the normal channel and independently calculate a cryptographic hash (or digest[1]) value from this information. This value is then transmitted from one device to the other over the OOB channel E in order to verify that both have the same value, indicating that the information exchanged on the normal channel has not been altered.

Comparison of hash (or fingerprint) values relies on human users. For example, both devices display a short string (representing the hash value) which users can then compare, or one device displays a string and the user copies it to the other device which then checks the value. Should the values match, the public keys that were exchanged over the normal channel are thus authenticated, allowing the devices to set up secure communications (such as SSL/TLS for example) if desired.

The use of an OOB channel to bootstrap security in ad hoc networks seems most promising, as can be seen by the amount of research in this area [8, 9, 18, 32]. However, the benefits of the OOB channel are tempered by the necessity of involving human users and creating extra work for them. This may not be easy to implement in practice, given the problems that users generally experience when using security

[1] A digest [18] is a cryptographic function related to a *universal hash function*. It has two arguments, namely a key and data to be digested. It should be designed so that *inter alia* the likelihood (as the key k varies) that $digest(k, A) = digest(k, B)$ is minimised for all $A \neq B$.

mechanisms, such as passwords [3, 33] or encryption [34].

Another issue is users' understanding of the need to have a more secure protocol. For example, current Bluetooth pairing is relatively straightforward, and requires one user to initially set a Personal Identification Number (PIN) on their device. Once another device attempts to communicate with it, this PIN is used as part of the authentication protocol. Given that attacks have been found against this [10], users need to understand the importance of the information they want to protect in order to buy into the need for additional security. Lacking this incentive to adopt stronger security, should a new security mechanism require more of an effort than currently is being expended on the PIN mechanism, it is likely that serious adoption issues may arise. In essence, unless users have a need for or motivation to perform security and understand the weakness of current mechanisms, they are not likely to accept a security mechanism that is harder to use.

2.2 SHCBK protocol

In order to demonstrate the potential sources of attacks on HISPs, we use the Symmetrised Hash Commitment Before Knowledge (SHCBK) protocol presented in [18]. We present the formal description of the protocol below.

1. $\forall A \longrightarrow_N \forall A' : A, INFO_A, longhash(A, k_A)$
2. $\forall A \longrightarrow_N \forall A' : k_A$
3. $\forall A \longrightarrow_E \forall A'$:users compare *Digest(k*,INFOS)* where k* is the XOR of all the $k'_A s$ for A $\in$ G

In the SHCBK, participating devices exchange information as follows:

- Message 1: each device sends its identity A, any information it wants to be authenticated including its public key $INFO_A$ and a *longhash*[2] of its identity and a key k_A.
- Message 2: After all devices have received message one, each device then discloses its key by sending it to all the other devices.
- Message 3: Devices independently calculate a *digest* of the XOR of all the keys k_As (k*) and *INFOs* and users compare the digest.

The SHCBK protocols (and a few others) make no assumptions on the secrecy of the information exchanged during the run of the protocol. As a result, the messages exchanged during the protocol may be overhead by the intruder without affecting the results. Abstracting away from the details of the protocol above and assuming that the protocol is resistant to all kinds of attacks on messages 1 and 2, message 3 becomes of interest to usability researchers. All protocols proposing the use of the normal channel and an OOB channel can be represented as in Figure 1. The figure helps to visualise what is going on clearly without focusing on the details of the normal channel as different protocols implement the normal channel differently.

Someone wishing to attack the SHCBK protocol would need to be able to inject his own messages to one of the devices such that the hash value of these messages matches that of other participants. This means making different messages hash to the same value. However, this is equivalent to a random guess which has a probability of 1 in 2^b to succeed [18], where b is the number of bits of the hash. The structure of the protocol means that it is impossible for an attacker to perform any combinatorial searching to improve this chance.

Another potential attack on this and similar protocols is to convince the users to agree that the hash values are matching when they are not. This means when users compare hash values, they have to ensure that they do it accurately as any mistakes may result in a successful attack.

The attack on technical security[3] can easily be addressed by using a hash function that produces a large number of bits that makes an attack infeasible given available technology. The larger the size of the hash value, the stronger or more secure (theoretically) the protocol is.

However, increasing the size of hash value can have a direct impact on the performance of the protocol, and can even increase the chances of an attack of the second sort succeeding. Depending on human users to handle large amounts of data with any degree of accuracy is risky, as they are both unreliable and slow [22].

2.3 Factors affecting HISPs

The security of a protocol depends on the correct behaviour of all the different components involved. In many instances, a protocol is said to be secure if the theoretical security set by mathematical proofs asserts that it cannot be broken under certain conditions. However, history has shown that protocols have failed—not because they were theoretically insecure—but because of other factors that where not considered to play a major role in the implementation or use of the protocol. For example, the Russian army in World War I found their cypher system too hard to use and reverted to simpler systems that were then easily broken by the Germans [1].

Schneier [27] argues that security is not just mathematics as it also involves people and as such, secure systems could be broken due to improper use rather than just mathematical incorrectness. In other words, security operates within a social-technical system [6].

In HISPs, the technical factors (i.e. the need for large numbers) seem to run directly against human factors (e.g. limited accuracy, memory constraints, and other issues of motivation). The larger the number of bits we use for the hash value, the more difficult it becomes for human users to compare them. This can lead to errors that may potentially compromise effective security[4] of the protocol, and certainly affect its performance. The critical success factor of any HISP design is finding the best compromise between the two given a specific usage scenario. For example, in military applications – where we can expect trained soldiers together with an understanding of the need for high security – it may be possible to use longer hash values than in other situations.

We require a trade off that can find the best compromise between the theoretical needs of security and the human

[2] *longhash* is a strong collision-resistant and inversion-resistant hash function

[3] This is the theoretical security of a protocol specified by protocol designers based on mathematical proofs. In HISPs, it is essentially the size of the fingerprint.

[4] This is the security of a protocol having considered its technical security, usability, as well as context of use.

factors of usability. In these protocols, usability plays a major role in deciding the size of the fingerprint (theoretical security). This is in contrast to methods that do not require human users to compare the values manually, and where the limit on the number of bits used depends generally on the bandwidth, the mode of comparison and the processing power of the devices under consideration.

Protocols that propose OOB channels that do not require human interaction have other weaknesses, however, such as a failure to provide security guarantees unless users *know* that these channels connect exactly the parties that are to be paired.

2.4 Related work

There has been very limited work on the usability of pairing methods. The most notable work, as stated earlier, is by Uzun et al. [31] where a comparative usability analysis of the traditional secure pairing methods was conducted. In their study, participants were asked to compare strings displayed on mobile devices, copy a PIN displayed on one device and enter it onto another, and select a PIN from among 4 numeric values that matched a string displayed on another device. Their findings were that participants regarded *copying and entering* as both secure and professional while *comparing* was regarded as easy to use. They recommend using a PIN of not more than 7 digits and that the user interface should be designed in such a way that the default option is the most secure. However, recently proposed HISPs require a wide range of sizes of information that needs to be compared, ranging from 16 bits [18, 25] to 68 bits [15]. This means that there is a need for methods that enable and facilitate the comparison of longer hash values than the recommended 7 digits (in addition protocols that currently only require 6 digits may change this requirement in time). Unknown to us until this paper was accepted, Kobsa et al. [13] have conducted similar work which has been presented at the same conference.

Our aim is to compare the usability and security of recently proposed OOB channels together with the traditional ones. In addition to increasing the number of methods compared, there are two significant differences between the original study and ours. Firstly, other than pairing devices, this study did not provide participants with a realistic goal, and we felt that this may have forced participants to focus on the pairing process alone. Secondly, in everyday use of mobile devices, users are likely to encounter scenarios where the compared strings are nearly similar–different by only one or two digits. We took this into account during the design of our study to specifically evaluate how such scenarios may affect the usability and security of OOB channels.

3. EXPERIMENTAL DESIGN

Given the range of secure pairing protocols currently available, we believe it is timely to compare the proposed methods to each other, in order to gain a better understanding of how these methods can be improved, adapted or modified to achieve optimum effective security. In addition, this gives us the opportunity to identify fundamental factors that affect the usability and security of OOB channels.

3.1 Definitions

The following definitions will be used throughout the rest of this paper.

- *Method*: a method is a way of comparing or transferring hash values independently computed by two or more devices. In this paper we focus on four methods: Comparing (*compare & confirm*), Selecting (*compare & select*), Entering (*copy & enter*), and Taking a picture of a barcode using a camera (*barcode*).
- *Representation*: a format in which the hash value is displayed to the user. This includes numeric and alphanumeric strings, words, barcodes, images, etc.
- *Method-representation*: Some of the methods are capable of using more than one representation and hence method-representation will be used to associate a particular representation with a particular method. For example, *compare & confirm*-alphanumeric describes the *compare & confirm* method using alphanumeric strings.

3.2 Participants

Participants in the study were respondents to an online advertisement. A total of 30 paid participants were recruited. Table 1 shows the participants' demographics.

Gender	Male: 47% Female: 53%
Age	18 - 25 40% 26 - 35 27% 36 - 45 13% 46 - 55 3% 56 - 65 13% 66 - 75 4%
Education	High School: 27% College: 27% Graduate: 26% Postgraduate: 20%

Table 1: Participant demographics

3.3 Material and apparatus

In real world device associations, devices are paired in order to achieve a specific goal such as exchanging files. Therefore, the process of association is merely a means to an end. Unlike other studies conducted before, we took this into account to evaluate whether participants are likely to pay less attention to the process of pairing devices. We simulated a *Peer-to-Peer (P2P) payment system*, in which one uses a personal device to make an electronic payment to another. The participants' goal was thus to carry out a successful payment transaction rather than merely associating devices.

The P2P payment system was developed using Java Microedition (J2ME) for portability and Bluetooth support on mobile devices. The study was conducted using two mobile phones; Nokia N95 with 2.6 inch screen, 240x320 pixels resolution, 332MHz CPU, 160MB memory capacity, and running Symbian OS v9.2 and Nokia N73 with a 2.4 inch screen, 240x320 pixels resolution, 220 MHz CPU, 42MB memory capacity, and running S60 operating system. Both devices support Mobile Information Device Profile version 2.0 (MIDP 2.0), a specification for use of Java on embedded devices [30], and both had cameras.

One of the mobile phones kept a log of participants' activities:

- time to complete the association process
- number of errors committed. Errors (or failures) were categorised as either security or non-security: security failures are those that may result in a user pairing her device with an unintended device, while non-security failures only result in a failure to successfully pair the two devices.

Other information was collected using three sets of questionnaires and interviews. An Enrolment Questionnaire (EQ) provided information on participants' demographic data, while After Scenario Questionnaires (ASQ) provided subjective data on three main components of a particular method:

- satisfaction with the ease with which a method-representation was used,
- satisfaction with the amount of time spent on a method-representation,
- whether a participant felt they could effectively carry out a transaction using a particular method-representation.

The ASQ was a rating scale type questionnaire consisting of 3 questions with answers based on a scale of 1 to 7, with 1 corresponding to *strongly agree* and 7 to *strongly disagree*. Many rating scales use a scale of 5 intervals rather than 7. However, it has been found that reliability of rating scales increases with the number of items and also the number of interval points for each item, and levels off at about 7 intervals with no significant increase after 11 intervals [14], hence the use of a 7 interval scale.

An End of Experiment (EoE) questionnaire gave participants an opportunity to identify methods they felt were easy, difficult and which ones they preferred or would avoid. Interviews gathered participants' views and comments on what they felt about the method-representations and what their experience in general was. In order to maintain consistency across participants but also be flexible enough to discuss issues that were to be raised, the interview was semi-structured.

3.4 Tasks

The study was conducted in a laboratory environment. Upon arrival, a participant was taken to the room where the study was conducted. A summary of what was to be done was given verbally, and, where this had not been received in advance, participants were asked to fill in an EQ. The participant then moved to a desk where she/he was provided with an instruction sheet, ASQ questionnaires and two mobile phones. The instructions were provided in written form to achieve consistency across all participants.

Part of the instructions included informing a participant about which of the two devices was to be assumed her personal device (in this case Nokia N73) and which one was the payee's. For most of the tasks, participants interacted only with the personal device while only observing the payee's, except for a few cases where a participant was required to press a button on the payee's device. However, participants had the freedom of holding the payee's device for their convenience.

Each participant carried 33 transactions, aimed at testing 14 different methods and method-representations. Since certain methods and method-representations require a user to decide whether the hash values match or not, additional scenarios were used to test a user's ability to correctly identify a match or lack of one (where appropriate, details of these additional scenarios are described in detail with each method).

The system presented these scenarios in a random order to increase internal validity [16]. Each participant completed 14 After Scenario Questionnaires (ASQ), for each of the 14 different methods and method-representations. After completing all 33 transactions, a participant filled in an EoE questionnaire and was interviewed. Participants required between 35 and 60 minutes to complete the study.

3.5 Methods

There are several proposed OOB channels for HISPs. However, our focus was on methods that require human attention and diligence, but also methods that did not require hardware modification of any of the devices we used in the study. The methods studied may be grouped into four broad categories:

3.5.1 Compare & Confirm (CC)

With this method, a user compares strings, sounds, melodies or images displayed on both devices and presses a button to indicate a match or disparity. In this study, the system simulated one-way authentication in which participants were required to press a button on their personal device only. Several hash value representations were used with this method: numeric, alphanumeric, *numeric & sound, alphanumeric & sound*, words, sentences, melodies, names of countries/cities, and images. These representations were used in the study because they have been proposed before.

The length of both numeric and alphanumeric strings was 6 characters rather than 4 as is common with PINs used for bank cards. Some protocols [7, 19, 23] propose using hash values of between 16 to 20 bits. However, a number of protocols propose using longer values, and as a result we decided to set the minimum number of bits for our study to be 20 bits, which corresponds to 6 decimal digits.

We used three different scenarios wherever possible:

- one where the two values matched
- one where they were significantly different
- one where they were different but nearly matching. Near matching means that the strings differ by a single digit for numeric, a single character for alphanumeric, a single word for words and sentences and a single country/city name for countries/cities.

This was done to draw attention to the potentially problematic near match case, since we suspected that participants might well make more security errors in these cases.

- **Numeric**: Each device displayed a 6 digit value and a participant had to compare the values on both devices with the instruction: *"Compare the two numbers. Are they DIFFERENT?"*. The participant then pressed *'SAME'* or *'DIFFERENT'* depending on whether they perceived the values to be same or different respectively. This wording is the same as that recommended by Uzun et al. [31], which they found improved the usability of the method. The values were displayed in two blocks of three digits. This separation was used when displaying numeric and alphanumeric values with a view that it might help users to split the comparison into two rather than the full string at once.

- **Alphanumeric**: With alphanumeric characters, a 32 character set was used. This includes all the numeric characters (0-9) and all characters in the English alphabet with the exception of 'I', 'O', 'Q', and 'U', since these could cause confusion. For example, the letter 'O' could be confused with the number 0 or the letter 'Q', 'I' with 1, and 'U' with 'V'. Thus each character in the set could represent 5 bits of the digest, and thus the complete string represented 30 bits. Despite alphanumeric representing more bits than numeric, it was felt necessary to display both types of strings in equal length to the user for comparative analysis.

- **Words**: Words were constructed from a dictionary of 1024 English verbs. Each word represented 10 bits and a set of four words was used in the study. For the experiment, a hash value was calculated from a randomly generated string and each segment of 10 bits was used to look up a specific word in the dictionary. With 10 bits per word, it would have been sufficient to compare two words. However, some mobile phones may use dictionaries that are much smaller than this for reasons of memory, and two or three words my not be sufficient for a 20 bit digest in such cases.

- **Sentences**: It has been suggested that people find it easier to deal with meaningful strings such as words than meaningless ones like alphanumeric [8]. Sentences were generated from the hash value based on MadLib [8] puzzles. A total of 32 sentences were stored which had at most 7 words of which 3 were missing. During the test, a sentence was selected and the missing words were queried from the dictionary using values from the digest.

- **Images**: People have been found to be good at dealing with images compared to strings [17] and proposals for users comparing images in HISPs have been made based on this finding. In the study, images were stored locally on the devices and only two scenarios were tested: matching and non-matching images. It was difficult to simulate near matching images as this is subjective as opposed to other representations such as numeric. A participant compared images displayed on both devices and pressed 'SAME' or 'DIFFERENT' on their personal device.

- **Melodies**: While all the above representations try to utilise human visual abilities, melodies try to utilise the ability to distinguish two audio sequences. In the study, melodies were generated by playing a note based on each digit of the 6 digit digest. In this test, only two scenarios were tested: matching and non-matching melodies. Participants played one melody after another, but it was also possible to play melodies on both devices simultaneously by pressing buttons on both devices.

- **Sound**: Another variation is to utilise both the visual and audio abilities of people by having a hash value displayed on one device while the other device reads out its value. The user listens to the string read on one device and compares it with the one displayed on the other device. Only numeric and alphanumeric strings were tested in this manner for three variations: matching, near matching and non-matching. Words and sentences were not used for this method as this would require text to speech capabilities which most mobile phones do not possess.

3.5.2 Compare & Select (CS)

Unlike *Compare & Confirm*, one string was displayed on one device (the payee's) and four strings of the same format were displayed on the other device. Due to limitations on the size of the displays, only numeric and alphanumeric values were used. For each of these, 4 variations were constructed:

- *Match* where one of the strings among the 4 displayed on the personal device matched the string displayed on the payee's,

- *Match and near-match* where one string matches and at least one other is near-matching the string displayed on the payee's device,

- *Different* where none of the 4 strings displayed on the personal device matches one on the payee's,

- *Different and near-matching* where there was no string among the 4 matching the one displayed on the payee's but at least one is near-matching.

Participants were asked to choose a string from the payer's device that matched the string on the payee's device or press "NOT FOUND" if no matching string was present.

3.5.3 Copy & Enter (CE)

With this method, one device displayed a string while the other asked the user to enter the same string. The device then compared the entered string with a locally generated one. If the two strings match, the device accepts the association otherwise rejects. The major difference with the other methods above is that the user does not do the comparison but rather only enters what she sees displayed on the other device. This reduces security failures to a single random guess, and makes it hard to simulate then and only left us to focus on non-security failures of the method. Due to limitations of the keypads on mobile phones, only numeric and alphanumeric strings were used for this method.

3.5.4 Barcode

This method differs completely from all the other methods in that it, firstly, requires a mobile phone with a camera and, secondly, the hash value is not displayed in a human readable format. The barcode reader used in this study was ZXing [21], and the barcode used in this study was in qrcode format [29]. One device displayed a qrcode barcode (in this case the payee's) while the other automatically activated the camera function and asked the user to point the camera at the other device and take a snapshot of the displayed image. The image used was 162x162 pixels and contained all letters of the English alphabet.

This method was included in the study because of its ability to accommodate more bits than the other methods and it

	Matching %		Non-matching %		Near-matching %		Total %	
	Security	Non-security	Security	Non-security	Security	Non-security	Security	Non-security
Numeric	0	3.3	0	0	0	0	0	3.3
Alphanumeric	0	16.7	3.3	0	10	0	13.3	16.7
Words	0	16.7	3.3	0	0	0	3.3	16.7
Sentences	0	16.7	0	0	0	0	0	16.7
Images	0	3.3	0	0	n/a	n/a	0	3.3
Melodies	0	36.7	6.7	0	n/a	n/a	6.7	36.7
Numeric & sound	0	0	0	0	3.3	0	3.3	0
Alphanumeric & sound	0	20	0	0	3.3	0	3.3	20
Country/City names	0	3.3	0	0	0	0	0	3.3

Table 2: Compare & confirm: Security and non-security failures

also presented a complete diversion from all the other methods in that a user only needs to point her mobile phone at the image displayed by another device and press the "CAPTURE" button. The image size used in the study was chosen to be big enough for easy focusing while the size of hash value encoded in it was consistent with the proposal that a barcode may be required to contain a fingerprint of a device's public key and other information such as name and address of device [15]

4. RESULTS

Each participant generated a separate log file for completion times and errors for all the method-representations and their variants. This constituted the main source of objective data. In addition, each participant produced 14 completed ASQs, 1 completed EoE, and 1 completed EQ. All this data was later compiled into Microsoft Excel worksheet in readiness for analysis using statistical tools provided by various packages. An audio recording of the interview for each participant was also a product of the study.

4.1 Objective results

This data revealed errors that various method-representations are prone to. The study was a repeated measure in which each participant was tested on all the scenarios. For 30 participants with 33 scenarios, a total of 990 data items were available for analysis.

4.1.1 Compare & Confirm

Table 2 shows a summary of results for *compare & confirm.* For each representation, the number of security and non-security failures according to three categories simulated in the study is shown. For images and melodies, no simulation was done for near-matching as explained above, hence n/a.

In *compare & confirm*, security failures are only possible in non-matching and near-matching scenarios while non-security failures are only possible in matching strings. The table shows that non-security failures ranged from 0% for *numeric & sound* to 36.7% for melodies while security failures ranged from 0% to 13.3%. It is worthy noting that security failures in this method are too high for a security application and they are just as likely to happen in a non-matching scenario as in a near-matching one.

Table 3 shows completion times for each representation. The results show that numeric and alphanumeric had the fastest completion times while melodies had the slowest. The table also shows that there were a number of outliers in completion times for each representation. For example, while numeric had a maximum completion time of 62 seconds, the mean was only 6 and the mode 3. These outliers could be explained in terms of participants getting distracted as they carried out a task. Outliers, however, were few in the data and their influence on the calculated means was minimal.

	Time - seconds					
	Mean	Mode	Median	SD	Min	Max
Numeric	6	3	5	7	1	62
Alphanumeric	6	2	5	4	1	25
Words	7	6	6	4	1	20
Sentences	11	6	8	10	2	56
Images	8	2	5	12	1	85
Melodies	24	15	20	16	4	88
Numeric & sound	14	10	11	11	4	76
Alphanumeric & sound	12	10	10	7	4	48
Country/ City names	9	4	8	5	2	26

Table 3: Compare & confirm: Completion times

In order to evaluate the significance of the differences in the dependant variable (time) and within-subjects variable, a one-way repeated measure analysis of variance (ANOVA) was perfomed. The results showed statistical significance in both factors, with F(8, 472) = 1.776 and p = .0000 for the dependent variable and F(59, 472) = 23.393) with p = .0007 for within-subjects. The variations in time is apparent from the means in Table 3; some methods took longer than others. The variation in the within-subjects variable could be attributed to the observation that younger people performed better (in completion times) than older ones.

4.1.2 Compare & Select

With *compare & select*, four scenarios were simulated as

	Matching %		Non matching %		Near matching %		Matching and near matching %		Total %	
	Security	Non-security	Security	Non-Security	Security	Non-security	Security	Non-security	Security	Non-security
Numeric	0	0	0	0	10	0	0	10	10	10
Alphanumeric	6.7	13.3	6.7	0	6.7	0	0	16.7	20	30

Table 4: Compare & select: Security and non-security failures

discussed above giving a total of 120 data items to analyse for 30 participants. Half of these had matching strings while the other half had non-matching strings. With this method, a user indicating that there is no match when there is results in a non-security failure. However, when a user selects a value and indicates that it is a match when there is none has two possible outcomes; either non-security or security failure. We, however, took the worst case scenario and regarded all errors resulting from selecting a non-matching value as security failures even though there is a chance that they might not be.

Table 4 shows a summary of errors for *compare & select.* Alphanumeric had a higher rate of both security and non-security failures. Despite the differences in both types of errors, there were no significant differences in terms of completion times as summarised in Table 5. A statistical test using one-way repeated ANOVA on completion times showed that the result was significant for the within-subjects variable with p = 0.0000 (F (119, 119)= 2.207) while it was not significant for the dependant variable (time) with p = 0.9255 (F(1, 119) = 0.009). The significance of the within-subjects variable could be explained in terms of the differences between younger and older participants and also participants' familiarity with the models of the mobile phones used in the study.

	Time - seconds					
	Mean	Mode	Median	SD	Min	Max
Numeric	9	4	7	7	2	52
Alphan-umeric	9	5	8	7	2	54

Table 5: Compare & select: Completion times

4.1.3 Copy & enter

Copy & enter had no variants resulting in 30 data items to analyse. It is clear from Table 6 that participants took longer to enter alphanumeric compared to numeric values. Entering alphanumeric also produced more errors than numeric. Of these errors, 75% of numeric errors were as a result of the confusion between copying the displayed digit and typing a four digit PIN for the payment transaction while 43% of alphanumeric errors were as a result of unfamiliarity with the model of the mobile phone.

	Time - seconds				Errors %
	Mean	Mode	Median	SD	Non-security
Numeric	17	10	14	14	13
Alphan-umeric	40	30	34	26	23

Table 6: Copy & enter: Completion times and errors

A one-way repeated measure ANOVA on completion times showed that there was no significance for the within-subjects variable with F(29,29) = .774 and p = .7531 while the results were significant for the dependent variable with F (1, 29) = 15.78 and p = .0004.

4.1.4 Barcode

The results of the *barcode* indicate that participants spent a significant amount of time focusing the camera on the displayed image. It also shows a very high percentage of non-security errors. These errors were as a result of not taking a clear shot of the image resulting in a failure by the decoding algorithm to reconstruct the image in order to decode it. Part of the problem is failure by participants to get a clear shot of the image, but more so was the implementation itself. The software worked quiet robustly when an image was displayed on a laptop screen. This was not the case, however, when the same image was displayed on the phones used mainly because of the size and resolution of the mobile phone screen.

Time - seconds						Errors %
Mean	Mode	Median	SD	Min	Max	Non-security
37	33	33	14	15	79	53

Table 7: Barcode: Completion times and errors

A source of concern for this method is not the number of errors (the results could be different with a more robust implementation) but what the users thought of it. Users seemed to be confused that the method, unlike other methods, was not intuitive. They could not figure out the purpose for taking a snapshot of an image displayed on another mobile phone.

Security failures can only be as a result of taking a snapshot of an unintended barcode. For example, a barcode displayed on a bogus cash machine or a fixed barcode on an access point that has been replaced by another from an intruder. However, in this study, such scenarios were not covered.

4.2 Subjective results

Objective results above show each method-representation in terms of two dependant variables; errors and time. There was a need, however, to gather participants' views on each method-representation in view of the possibility that despite a method-representation performing objectively well in the two dependant variables, participants may not necessarily favour such.

4.2.1 ASQ (rating scores)

Participants gave their rating scores to each of the three items on the ASQ. These ratings were summed and averaged to calculate a single score for each participant for each method-representation. The raw data was inverted before presentation so that a high score represents a high agreement from the participant rather than what was in the questionnaire where a high score indicated a disagreement (low score) from the participant.

The results, summarised in Figure 2, show that most methods had a score higher than 5 except melodies, *barcode*, and *copy & enter* alphanumeric. *Compare & confirm*-numeric and *compare & select*-numeric had the highest scores of 6.3 followed by *copy & enter*-numeric at 6.1.

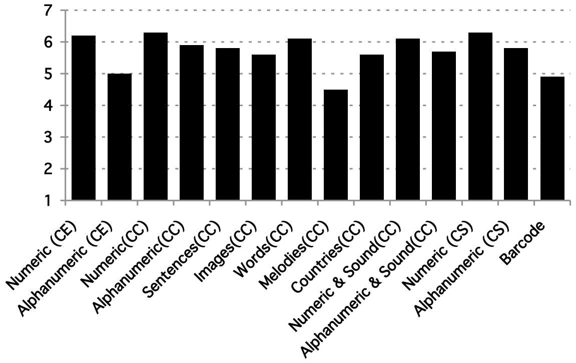

Figure 2: Participants ratings based on ASQ

On a scale of 7 intervals, a method-representation was regarded as usable if it had a score of 5.6 or more. This was based on the results of [20] which indicated that a system is usable if it has a score of 4 or more on a scale of 1 to 5 or a score of 5.6 on a scale of 1 to 7. Based on this result, then *copy & enter* alphanumeric, melodies, *alphanumeric & sound*, and *barcode* are less usable.

4.2.2 Preferred methods

In addition to assigning rating scores to each method-representation, participants were asked to indicate all the method-representations that they felt were easy and also to indicate their preferred one. The results are summarised in Figure 3.

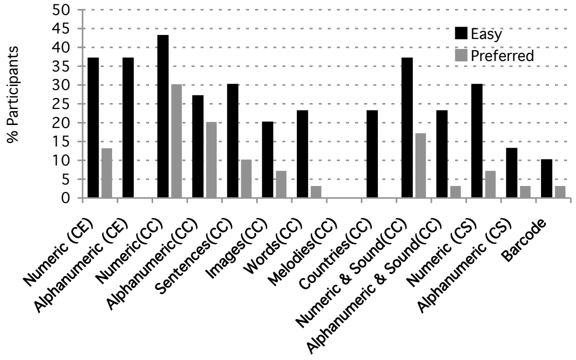

Figure 3: Participant ratings: Easy of use

4.2.3 Unpreferred methods

Despite participants indicating which method-representations they felt were easy, it was also necessary for participants to explicitly indicate which method-representation they felt were difficult and which one they would avoid, given a choice. The results, summarised in Figure 4, correlate with those in Figure 3; melodies had the lowest score in Table 3 and they had the highest score in Figure 4. Generally, method-representations that had a high score in Figure 3 had low scores in Figure 4 and vice versa indicating that the results correlate.

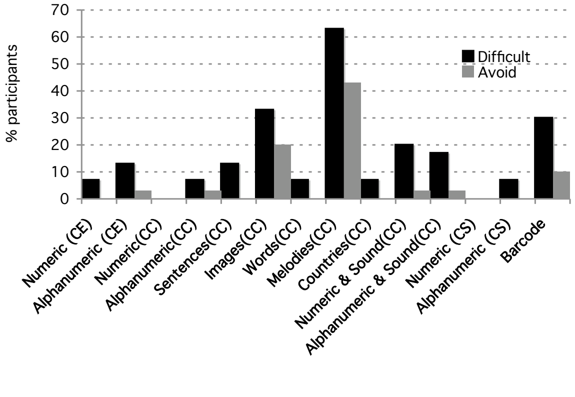

Figure 4: Participant ratings: difficult of use.

5. ANALYSIS AND DISCUSSION

In order to analyse the relative performance of each method-representation in terms of all the parameters measured in the study, it was necessary to have a single overall score for each method-representation. While trying to find a method to quantify the measurements and calculate a single score, only Single Usability Metric (SUM) [24] was found to suit this purpose. This is a method of calculating a Single Usability Score by standardising raw data for each of the parameters under the study. Standardising scores allows comparison of values of different variables regardless of their original unit.

In order to calculate standardised scores, however, it was necessary to come up with specification (limit) scores [24] for some of the parameters measured. Specification scores are the acceptable values below which a method is regarded as less usable. There are various ways by which specification scores may be determined including using an existing system, a prototype, a user's earlier performance, or an absolute scale [12]. However, because of lack of previous data on task completion times for the methods tested, twice the modal score was used as the specification score. On a rating scale of 1 to 7 (7 indicating best), an average score of 5.6 is the minimum for a product that is deemed usable [20], and as such this value was used as a specification score for the ASQ scores.

With the specification scores determined, Table 8 shows the ranking of the method-representations in order of their single usability scores. The scores shown under each column are quality [24] scores rather than defects. For example, the table shows for *compare & confirm* melodies that despite a potential for non-security failures, 63.4% of cases resulted in a successful accurate comparison.

Based on the SUM scores, Table 8 shows that *compare & confirm* (numeric, alphanumeric and words, *numeric &*

	Non-security failures	Time %	ASQ %	Preference	Easy %	SUM score
Numeric(CC)	98.4	50.6	74.8	30	100	73.7
Alphanumeric(CC)	91.7	83.1	58.7	20	93	72.5
Words(CC)	91.7	88.1	63.7	3	93	70.6
Numeric & sound (CC)	100	63	68.9	17	80	69.2
Numeric(CE)	97.5	59	68.1	13	93	69
Numeric(CS)	95.8	44.6	80.1	7	100	68.3
Alphanumeric &sound (CC)	90	84.8	52.7	3	83	65.8
Alphanumeric(CS)	98	56.4	55.4	3	93	64.2
Sentences(CC)	91.7	54.9	55.5	10	87	62.9
Alphanumeric(CE)	87	78	34.5	0	87	60.4
Countries(CC)	96.7	41	50	0	93	59.1
Images(CC)	96.7	37.1	50	7	67	54.3
Barcode	47	97.2	33.6	3	70	53
Melodies(CC)	63.4	63.8	27.2	0	37	40.7

Table 8: Ranking based on SUM Scores

sound) ranked top ranging from 69.2% to 73.7%, followed by *copy & enter* (numeric) at 69%, *compare & select* (numeric) at 68.3% and finally *barcode* at 53%. In comparison with the results of Uzun et al. [31], *compare & confirm* was ranked highest in both cases while *compare & select* ranked higher than *copy and enter* in their case. In our study, participants felt that *compare & select* was a complication of *compare & confirm*, hence a lower ranking than *copy & enter*.

Among the methods tested for *compare & confirm* that received very low ratings were melodies (40.7%), images (54.3%), country/city names (59.1%), and alphanumeric & sound (65.8%). Sixty percent of participants said that *'not being a musical person'* made comparing melodies hard for them. Country/city names and sentences were lowly rated because they were *'too long'* for most participants while 10% of participants felt that it was *'strange'* for a mobile phone to be *'talking'* to them (alphanumeric & sound, numeric & sound). Participants also found comparing images challenging especially those that were meant to be similar. This may be attributed to the way the questions on the devices were phrased "are they DIFFERENT?" – a key recommendation from Uzun et al. [31]. For similar images, participants spent considerable amount of time looking for differences in the images but this was not the case with non-similar images since the differences were quite apparent. It appears therefore that this phrasing helps reduce the number of security failures at the expense of performance.

In *copy & enter*, it was expected that alphanumeric would receive a low rating because of the difficult in entering text on a mobile keypad especially where one is required to switch between numeric and text. *Barcode* was the lowest ranked method. This was because, first, participants did not understand how the method fitted into the simulated payment system. Second, they were not sure to what details or how clear the image should be. Third, 67% of older participants (> 45 years) had difficulties because they are *'not used to taking pictures using a mobile phone'*.

While the rankings in Table 8 provide the relative usability of the methods tested, they do not give us a complete means by which we may make an informed decision on a method suitable for the OOB channel. This is because security also needs to be considered. Security failures were not included in Table 8 because we consider them to be critical and they cannot carry the same weight as other factors analysed. To this regard, it was necessary to re-rank the methods in a manner that gave security failures more weight compared to other factors. Table 9 shows this ranking. The method-representations are ranked according to their susceptibility to security failures, followed by the number of actual security failures observed in the study and finally their SUM scores.

	Subject to security failures	Security failures	SUM score
Numeric(CE)	No	0	69
Alphanumeric(CE)	No	0	60.4
Barcode	No	0	53
Numeric(CC)	Yes	0	73.7
Sentences(CC)	Yes	0	62.9
Countries(CC)	Yes	0	59.1
Images(CC)	Yes	0	54.3
Words(CC)	Yes	3.3	70.6
Numeric & sound	Yes	3.3	69.2
Alphanumeric & sound	Yes	3.3	65.8
Melodies(CC)	Yes	6.7	40.7
Numeric(CS)	Yes	10	68.3
Alphanumeric(CC)	Yes	13.3	72.5
Alphanumeric(CS)	Yes	20	64.2

Table 9: Ranking based on security failures

Copy & enter was ranked first because it is not susceptible to security failures and it had a relatively high SUM score. *Barcode* was ranked second despite a relatively very low SUM score, followed by *compare & confirm* and *compare & select.*

An analysis of interview data revealed two distinct groups of participants. One group was only concerned with the ease of use of the method-representations tested. For this group, the speed, mental, and physical workload of completing the comparison was of the utmost importance. This most likely explains why *compare & confirm* numeric and alphanumeric had high preferences, and ASQ scores. The other group wanted a method that was usable but at the same time reassured them that the correct devices were being paired through the accurate comparison of hash values. These participants were found to favour *copy & enter*, firstly, because they felt it was secure since they *'can double check the string entered'* and hence were more likely to copy correctly. Secondly, they indicated that they were used to typing short strings such as PINs and short text messages on mobile phones and cash machines. Thirdly, they were afraid that it was *'easy for one to be distracted in compare & confirm'* or *'only compare the first few digits and think that the rest are matching'* resulting in pairing with the wrong device.

While *compare & confirm* was ranked first in terms of usability, it is subject to security failures. It does not compel users to compare the strings accurately. Users may cause security failures deliberately (by choosing not to compare), or because they are distracted, stressed, or conditioned to seeing matching values. Though these situations cannot easily be captured in a lab environment, they do exist in the real world and cannot be ignored. In fact, we expect method-representations to perform better in a lab environment than in the real world. Our study was designed according to the interface design recommendations in [31], however the number of security failures observed in the various representations of the *compare & confirm* method indicate that the user interface design is not sufficient in eliminating their occurrence.

Despite *copy & enter* numeric being ranked below *compare & confirm* in terms of usability, it is not subject to security failures and users cannot bypass it, otherwise the association will fail. While it is recognised that it may be difficult for some users to enter text on mobile devices such as a phone, the popularity of Short Message Service (SMS) means that a growing number of users will be familiar with this means of interaction. The results do not indicate that this method is unusable, but its relative usability compared to other methods, combined with its inherently stronger security, makes it the best candidate among the methods tested.

Barcode's resistance to security failures, together with its ability to accommodate more bits than other methods in this study, makes it a very interesting candidate for an OOB channel. However, this method is limited to devices with cameras, which most laptops or PDAs generally lack. Moreover, most participants did not understand the process intuitively, and a substantial number of them felt it was an *'added complexity'*. It may be possible to overcome this problem through education and exposure to the technology, however, given the necessity of using a camera, this method is somewhat limited in its scope of application.

The results of this study provide an insight into these method-representations, but further investigation is needed. The findings show that an OOB channel should not be selected only based on its perceived usability, but on a number of other factors too. Based on the observations from the study, we outline some of the factors that may affect usability, security, and eventual success of OOB channels.

- User conditioning: repetitive security tasks to which users can predict the outcome should be avoided. For example, an OOB channel using *compare & confirm* used in an everyday application such as mobile payments may result in users anticipating matching hash values and getting used to pressing “SAME”.
- User motivation: users have different levels of motivation to perform security tasks in different circumstances. In the study, a good number of participants indicated that they would prefer typing digits longer than 6 digits for financial transactions exceeding a certain monetary value. A system designer may want to exploit this additional motivation in cases where a higher level of security is desirable.
- Security failures: susceptibility to security failures may not be acceptable in high security applications. The fact that a good number of representations for *compare & confirm* had no security failures for a sample size of 30 in a lab environment does not rule out their possibility. Users may operate devices under different conditions (such as stressful, noisy or distracting situations) which may have an impact on how well they compare hash values.
- Attentiveness: Users can easily be distracted causing them to shift their attention from the pairing process. OOB channels must not demand undivided attention throughout the pairing process as this is likely to cause frustrations in scenarios where the user is distracted. For example, comparing melodies requires users to be attentive while the melody plays – any distraction requires restarting from the beginning.

6. CONCLUSION

Based on the results, we conclude that currently proposed methods require rethinking not only from a usability perspective but also from a security standpoint: *compare & confirm* and *compare & select* are not suitable methods for the out-of-band channel because of security failures. While it is important to ensure the usability of these protocols, it is also important to ensure that users are compelled to carry out their roles in a diligent way. It is thus also our conclusion from this work that there is need for methods that encourage and compel users to carry out their role diligently, and also adopting measures that suit human and contextual needs. We firmly believe that the findings in this paper show evidence of this and that it will motivate researchers to rethink the methods for the OOB channel in device associations.

7. REFERENCES

[1] R. Anderson. Why cryptosystems fail. *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 215–227, 1993.

[2] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *In Symposium on Network and Distributed Systems Security (NDSS '02), San Diego, California*, 2002.

[3] S. Brostoff and M. A. Sasse. Are passfaces more usable than passwords? a field trial investigation. In *Proceedings of HCI 2000*, 2000.

[4] M. Čagalj, S. Čapkun, and J. Hubaux. Key agreement in peer-to-peer wireless networks. In *Proceedings of the IEEE (Special Issue on Cryptography and Security)*. IEEE, 2006.

[5] D. Dolev and A. Yao. On the security of public key protocols. In *Information Theory, IEEE Transactions on*, volume 29(2), pages 198–208, 1983.

[6] I. Flechais. *Designing Secure and Usable System*. PhD thesis, University of London, 2005.

[7] C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. In *RSA Cryptobytes*, volume 7(1), pages 29–37. RSA Security, Spring 2004.

[8] M. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human-verifiable authentication based on audio. In *Proc. 26th IEEE International Conference on Distributed Computing Systems ICDCS 2006*, pages 10–10, 04–07 July 2006.

[9] B. S. I. Group. Simple pairing white paper. www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/ SimplePairing_WP_V10r00.pdf.

[10] M. Jakobsson and S. Wetzel. Security weaknesses in bluetooth. In *Lecture Notes in Computer Science*, volume 2020, pages 176+, 2001.

[11] J. Jamaluddin, N. Zotou, and P. Coulton. Mobile phone vulnerabilities: a new generation of malware. *Consumer Electronics, 2004 IEEE International Symposium on*, pages 199–202, Sept. 1-3, 2004.

[12] S. Jeff. and E. Kindlund. How long should a task take? identifying specification limits for task times in usability tests. In *In Proceeding of the Human Computer Interaction International Conference HCII 2005), Las Vegas*, 2005.

[13] A. Kobsa, R. Sonawalla, and G. Tsudik. Serial hook-ups: A comparative usability study of secure device pairing methods. In *SOUPS '09: Proceedings of the 5th symposium on Usable privacy and security*, 2009.

[14] J. R. Lewis. Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *Int. J. Hum.-Comput. Interact.*, 7(1):57–78, 1995.

[15] J. McCune, A. Perrig, and M. Reiter. Seeing-is-believing: using camera phones for human-verifiable authentication. In *Proc. IEEE Symposium on Security and Privacy*, pages 110–124, 8–11 May 2005.

[16] A. Minke. Conducting repeated measures analyses: Experimental design considerations. Technical report, Annual Meeting of the Southwest Educational Research Association (Austin, TX, January 23-25, 1997), 1997.

[17] W. Moncur and G. Leplâtre. Pictures at the atm: exploring the usability of multiple graphical passwords. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 887–894, New York, NY, USA, 2007. ACM.

[18] L. H. Nguyen and A. W. Roscoe. Efficient group authentication protocol based on human interaction. In *Proceedings of the Workshop on Foundation of Computer Security and Automated Reasoning Protocol Security Analysis (FCS-ARSPA)*, pages 9–33, 2006.

[19] L. H. Nguyen and A. W. Roscoe. Authenticating ad hoc networks by comparison of short digests. In *Journal of Information and Computation. Special Issue of Information and Computation on Computer Security: Foundations and Automated Reasoning*, 2007.

[20] J. Nielsen and J. Levy. Measuring usability: preference vs. performance. *Commun. ACM*, 37(4):66–75, April 1994.

[21] Owen. Zxing: Multi-format 1d/2d barcode image processing library with clients for android, java, and iphone project: http://code.google.com/p/zxing/.

[22] A. Perrig and D. Song. Hash visualization: a new technique to improve real-world security. In *International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, pages 131–138, 1999.

[23] A. W. Roscoe and L. H. Nguyen. Authenticating ad hoc networks by comparison of short digests. *Information and Computation*, 206:250–271, 2008.

[24] J. Sauro and E. Kindlund. A method to standardize usability metrics into a single score. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–409, New York, NY, USA, 2005. ACM.

[25] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel (short paper). In *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 306–313, Washington, DC, USA, 2006. IEEE Computer Society.

[26] B. Schneier. Biometrics: Truths and fictions. Crypto-Gram Newsletter, August 15, 1998.

[27] B. Schneier. Security in the real-world: How to evaluate security technology, 1999.

[28] F. Stajano and R. Anderson. The resurrecting duckling: security issues for ubiquitous computing. *Computer*, 35(4):22–26, Part Supplement,&pril 2002.

[29] I. Standards. Qrcode standard: Iso/iec18004.

[30] S. Systems. Midp specification: http://java.sun.com/products/midp/.

[31] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Financial Cryptography and Data Security*, pages 307–324, 2007.

[32] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Lecture Notes in Computer Science*, volume 3621, pages 309–326, November 2005.

[33] D. Weirich and M. A. Sasse. Pretty good persuasion: a first step towards effective password security in the real world. In *NSPW '01: Proceedings of the 2001 workshop on New security paradigms*, pages 137–143, New York, NY, USA, 2001. ACM.

[34] A. Whitten and J. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th USENIX Security Symposium, August 1999, Washington*, pages 169–183, 1999.