HUMAN FACTORS IN HCBK PROTOCOL

By

Ronald Kainda

University College

Supervisor: Professor Bill Roscoe

University of Oxford Computing Laboratory

2007

Thesis submitted in partial fulfilment of the requirement for the degree of Master of Science in Computer Science

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to my supervisor Professor Andrew William Roscoe for his strong support and guidance through out the project. I would also want to thank Andrew Warr and Dr. Marina Jirotka for providing me with information and guidance on the subject of Human Computer Interaction. Thanks also to Long Nguyen for helping me understand the (S)HCBK protocols and for his valuable comments before and during my thesis write up.

I would also like to thank my parents and siblings for their continued support through out the study. Thanks to all my Oxford friends for making the year enjoyable and shorter.

ABSTRACT

Device authentication in ad hoc networks is a challenge as no PKI infrastructure can cover all mobile devices as they daily become pervasive in our lives. Neither is a centralized server solution workable in these environments. As a result, the viable solution that researchers are looking into is to use a human mediated medium through which these devices may be authenticated. This is based on the fact that human users can easily identify objects by seeing, touching or hearing. Human users also establish some kind of trust to those they want to communicate to. Based on this trust, human users can ensure that their device is communicating to the device of the person they want to communicate with. However, this approach has come with challenges of usability. The major challenge is minimizing human effort involved to make these systems highly usable and at the same time maximizing achievable security. In this project, the focus is on assessing the usability of the proposed methods in relation to the HCBK protocol.

TABLE OF CONTENTS

Page

Acknowledgements ………………………………………………… ii

Abstract ……………………………………………………………… iii

Contents …………………………………………………………….. iv

List of tables …………………………………………………….…. vi

List of figures …………………………………………………… vii

CHAPTERS

1 Introduction.. 1
1.1 Bootstrapping security in ad hoc networks 1
1.2 HCBK Protocol 2
1.3 SHCBK Protocol 5
1.4 Project overview.. 6
1.5 Project outline.. 6
2 Related Work on Empirical channels 8
2.1 Seeing is Believing (SiB) 8
2.2 Loud and Clear... 9
2.3 Human-Assisted Pure Audio Device Pairing (HAPADEP) 10
2.4 Wireless direct Channels.. 12
2.5 Hash visualization... 12
2.6 Wired channel 13
2.7 Manual Method.. 13
2.8 Integrity Regions 14
2.9 Integrity Codes 15
2.10 Usability analysis 15
3 CLASSIFICATION OF SCENARIOS 17
3.1 Devices adjacent to each other.. 17
3.1.1 Two devices single user.. 17
3.1.2 Multiple devices with single user.. 17
3.1.3 Multiple devices with multiple users.. 18
3.2 Devices not adjacent to each other 18
3.2.1 Two devices... 18

3.2.2 Multiple devices ... 19
3.3 Level of security ... 19
3.3.1 Little or no Security required ... 19
3.3.2 Medium security ... 19
3.3.3 Maximum security ... 20
3.4 Modes of comparison ... 20
3.4.1 Audio channel ... 20
3.4.2 Digital cameras ... 20
3.4.3 Compare and Confirm ... 21
3.4.4 Copy and enter ... 22
3.4.5 Compare and select ... 22
3.4.6 Direct connection ... 24
3.5 Scenario-mode of comparison matrix ... 25
3.6 Material compared ... 26
4 Maximising the security of the protocol ... 27
4.1 Technological factors ... 27
4.2 Human factors ... 28
4.2.1 Retry mode ... 29
4.3 Reliability ... 31
4.4 Technological factors versus human factors ... 32
5 Experimental results ... 34
5.1 Details of the experiment ... 34
5.2 Summary Results ... 39
5.3 Compare and confirm ... 41
5.4 Copy and enter ... 42
5.5 Compare and select ... 43
6 Implementing the 1-bit empirical channel ... 45
6.1 direct human interaction ... 45
6.2 Summary ... 47
7 Summary and Conclusion ... 48
7.1 Recommendations ... 48
7.3 Conclusion ... 50
8 References ... 52

LIST OF TABLES

Table 1: Scenarios versus modes of comparison …………………….... 25

Table 2: Modes of comparison versus material compared ....……….. 26

Table 3: Retry mode scenario ……….…………….....………………. 30

Table 4: Average time and error rates ……………………………….. 40

Table 4: User ratings …………………………………………………… 41

LIST OF FIGURES

Figure 1: Graphical representation of the HCBK protocol ………… 4

Figure 2: Simulated device ……………………………………….………… 35

Figure 3: Compare and confirm of hexadecimals…………………… 37

Figure 4: Method rating ………………………………………….………… 38

CHAPTER 1

Introduction

1.1 Bootstrapping security in ad hoc networks

Bootstrapping security, from first principles, without the existence of a Public Key Infrastructure (PKI) or pre-shared information between pairing devices is a difficult, if not an impossible, task. This is because, under the Dolev-Yao attack model[1], an intruder is able to modify, spoof, and intercept messages. Ad hoc or ubiquitous computing environments need such bootstrapping because in most cases the devices authenticating each other have no pre-shared information. The use of PKI or centralised servers does not seem to be the solution to these environments as no server or PKI infrastructure can hold up-to-date information about all pervasive devices as these devices frequently change their identities. Goodrich et al [5] have shown why PKI or centralised server infrastructure is not suitable in this environment.

Given the above scenario, how then do we bootstrap security in ad hoc networks? Many researchers have proposed the use of an extra channel with characteristics that are different from the normal Dolev-Yao channel. On this channel, an intruder cannot intercept, modify, or fake messages.

[1] *The Dolev-Yao threat model represents an attacker that can overhear, intercept and synthesise any message and is only limited by the constraints of the cryptographic methods used. [29]*

This channel should not be vulnerable to Man-In-the-Middle attacks. Bill Roscoe named this channel the Empirical channel [11, 26]. On this channel, exchange of information between authenticating devices is secure.

A question remains on how we implement the empirical channel so that the channel remains secure and not vulnerable to attacks? Bill Roscoe and Long Nguyen proposed creating this channel by shifting trust from authenticating devices to the humans using these devices. This approach has been called human-centric computing, human centred computer security, human assisted authentication, and human verifiable authentication [1, 5, 6, 12, 25, 26]. Humans interacting can physically see each other. They may also physically see the devices that they need to have authenticated. They can also recognise each other's voice in a case where they are far apart. In these scenarios, it is impossible for an intruder to convince a user that the person sitting across the table is someone else and not who s/he thinks to be or the device s/he is looking at or touching is not the device s/he thinks it is.

1.2 HCBK Protocol

The Hash Commitment Before Knowledge (HCBK) protocol [11] was discovered by Bill Roscoe with the aim of bootstrapping security in ad hoc networks. As pointed out earlier, it is common in these environments that these devices have no pre-shared information. The HCBK protocol aims to achieve a high level of authenticity of the information that these devices exchange. Unlike most of the protocols that have been proposed for ad hoc networks, HCBK protocol covers a wide range of scenarios. The scenarios range from two devices being paired by a single or two users to an arbitrary sized group of devices, say a lecture theatre full of people.

The HCBK protocol achieves its authentication goal through the use of a combination of the normal Dolev-Yao channel and Empirical channel or human mediated channel where an intruder cannot spoof messages as opposed to the normal high bandwidth channel which follows the Dolev-Yao attack model.

0. $I \xrightarrow{}_N \forall S$: I

1. $\forall A \xrightarrow{}_N \forall A'$: (A, INFO_A)

2a. $I \xrightarrow{}_N \forall S$: longhash(N_I)

2b. $\forall S \xrightarrow{}_E I$: committed

3. $I \xrightarrow{}_N \forall S$: N_I

4a. $\forall A$ displays : digest (N_I, all messages 1), init (I, A)

4b. $\forall A \xrightarrow{}_E \forall A'$: users compare and check presence of I.

HCBK protocol

The notations used in the above protocol are as follows:

- *I* is the Initiator; s/he may be any member of the group who is trusted because of his/her status or position or s/he may be someone who requires to authenticate the other members of the group.
- *A* is any group member. It may be *I* or any other member.
- *Longhash* is a strongly collision-resistant and inversion-resistant hash function.
- *digest* is a digest function producing as many bits as we expect the humans to compare.
- $\xrightarrow{}_N$ is the Normal Dolev-Yao channel where messages can be overhead, modified or deleted.
- $\xrightarrow{}_E$ is the empirical channel where messages cannot be spoofed.

The meanings of these messages are as follows:

- Message 1 publishes the information that all the nodes want to have attached to them, via the insecure channel.
- Message 2a has *I* devise a nonce N_I with sufficient entropy that longhash(N_I), which it publishes here, has no more than an infinitesimal likelihood of any combinatorial attack on it succeeding.
- Message 2b has all the slaves communicate to *I* that they have received Message 2a and are therefore committed to their final digest value (though none of them know it yet).

– Message 3 has I publish the nonce N_I after it has received commitments from all members of the group over the empirical channels. All slaves now have the duty to check if the values of Messages 2a and 3 are consistent.
– Message 4a has all the nodes compute what should be the same digest value.
– Message 4b has them compare these values.

In the above protocol, we see that messages 1, 2a and 3 are transmitted on the normal Dolev-Yao channel ($\longrightarrow_N$) where messages can be faked, modified and intercepted. Messages 2b and 4b use the secure empirical channel ($\longrightarrow_E$). Figure 1 below shows a 2-party or pair-wise graphical representation of the HCBK protocol. N is the normal Dolev-Yao channel and E is the Empirical channel that bypasses an intruder.

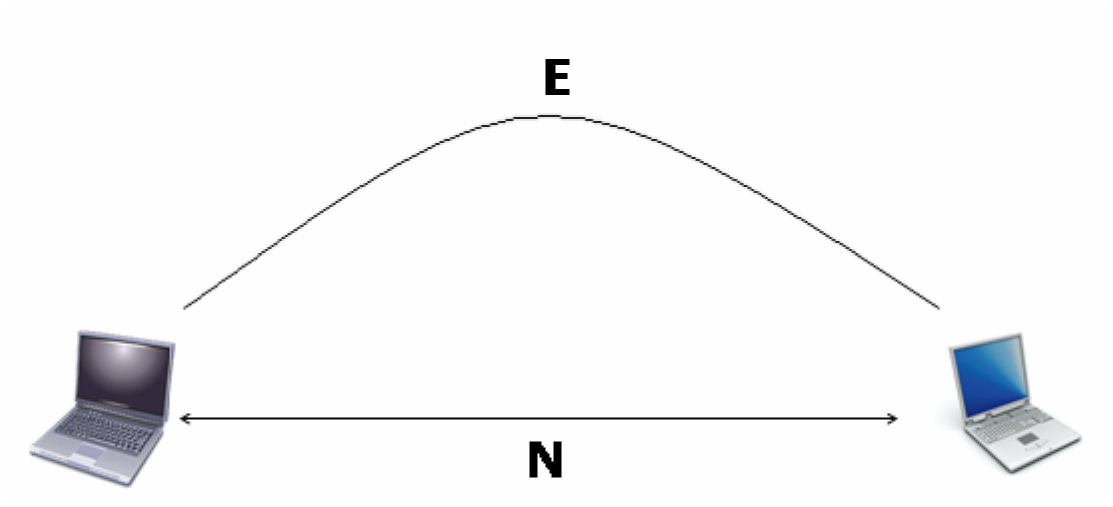

Figure 1. Graphical representation of HCBK

In message 2b, users have to confirm to I that their devices have received message 2a while in message 4b they have to exchange/compare the output of the digest function using the empirical channel.

The goal of the HCBK protocol is to ensure that devices authenticate each other securely. Since an intruder has control over the high bandwidth channel, information exchanged over this channel is difficult to authenticate without the use of a trusted or an empirical channel. The HCBK achieves its goal by using a combination of the normal high bandwidth channel and the human mediated channel. Human users use the human mediated channel to verify the integrity of the messages exchanged over the normal channel.

To carry out a successful attack on the HCBK protocol, an intruder has 1/H chance, where H is equal to 2^b and b is the number of bits of the digest value compared by human users. The value of b is a trade off between human effort and required security. Users cannot compare large strings of data and very short strings are vulnerable to attacks. As a result, we assume b is never less than 16 bits but it can be as large as is required to attain the desired level of security but also the protocol should remain highly usable and acceptable to human users.

1.3 SHCBK Protocol

A variant of the original HCBK protocol was discovered by Bill Roscoe and Long Nguyen and was presented also in [11]. The protocol named SHCBK (Symmetrised HCBK) was discovered with a view of avoiding the requirement of the HCBK protocol that there must be a trusted participant (*I* in the above representation) among the group running the protocol. The SHCBK protocol does not require a trusted participant.

1. $\forall A \longrightarrow_N \forall A'$: A, $\mathrm{INFO'}_{A,}$ $longhash(hk_A)$
2. $\forall A \longrightarrow_N \forall A'$: hk_A
3. $\forall A \longrightarrow_E \forall A'$: users compare *digest* $(hk^*, \{INFO'_A \mid A \in G\})$

Where hk^* is the XOR of all the hk_A*'s* for $A \in G$

SHCBK protocol

The details of the protocol are presented in [11]. From a usability point of view, message 4b of the HCBK protocol is the same as message 3 of the SHCBK protocol, since in both messages human users have to compare the output of a digest function. As a result, we need to mention that most of the discussion concerning the HCBK protocol also apply to

SHCBK and, hence, we will be using the notation (S)HCBK to refer to both the HCBK and SHCBK protocol where applicable.

1.4 Project overview

Shifting trust to human users creates a burden on the users. It is extra work for them. They have to make sure that the devices they think they are connecting to are the ones they are connecting to only. This may not be easy to implement, taking into account human weaknesses. People may need to understand the importance of security as well as the importance of playing their roles correctly if we have to achieve the desired level of security.

In this research our focus is on accessing what approaches are workable and efficient for real human users. We focus on implementing this channel without the use of extra devices, or demanding any standardised ports on the devices as this will limit the application of the protocol. We focus on assessing what types of values human users work well with, and the effects of varying the lengths of such values. We make these assessments in light of other work related to usability. We also analyse and recommend ways in which the protocol can handle the various situations that may occur. In particular, we discuss how to react to a failed attack.

1.5 Project outline

Chapter 2 gives an overview of previous work done on security protocols that involve human user action to accomplish secure exchange of cryptographic information. It describes specific methods that have been proposed to implement the empirical channel together with their advantages and disadvantages.

Chapter 3 discusses the various scenarios that may occur in ad hoc networks. It also discusses the various modes of comparison of digest values. The modes of comparison include both manual and non-manual. Manual methods are those that directly involve the human users in comparing strings or images, while non-manual methods are those that do not. Different kinds of material compared using each of the modes of comparison are discussed as well.

Chapter 4 discusses the main factors that contribute to the security of a protocol and how we can maximise this. These factors are not only in relation to the (S)HCBK protocols but to all security protocols that involve human users at large. The chapter discusses mainly how technological and human factors affect the security of the protocol and how, in most cases, these two factors contradict each other. In relation to the (S)HCBK protocols, the chapter also discusses reliability of the different kinds of empirical channels as they are also a major factor since a weak empirical channel compromises the security of these protocols. A discussion on how the protocol technologically responds to a perceived failed attack is also presented.

Chapter 5 presents the results of an experiment carried out on the different modes of comparison that directly involve human users. These results shed more light on the modes of comparison as to how usable and secure they are. The chapter analyses these results and draws some conclusion on the usability and security of each method.

Chapter 6 discusses the implementation of the 1-bit empirical channel of message 2b in the HCBK protocol. It proposes a secure method that counters possible attacks on the channel.

Chapter 7 outlines recommendations and draws some conclusions on this project. It also points out what this project has achieved and makes some suggestions on future work.

CHAPTER 2

Related Work on Empirical channels

In this chapter, we present some of the proposed methods of implementing the empirical channel. Previous work mostly focuses on pairing two devices that are physically adjacent to each other and very few of these methods can easily extend to other scenarios. Some of the proposed methods attempt to authenticate exchanged information without human involvement. This has an advantage on the human users but mostly offers less assurance that the information a device received is from the expected device. Some proposals employ expensive methods by either requiring extra hardware on the devices or demanding that all devices have some kind of common interface. This chapter gives a brief overview of these methods and also discusses the strengths and weaknesses of each of these methods.

2.1 Seeing is Believing (SiB)

McCune et al [1] proposed "Seeing is Believing." The authors present a human assisted device authentication method that uses mobile camera-phones as a means of exchanging secret information between the devices authenticating each other. Seeing is Believing uses a 2D barcode that constitutes a total of 83 bits with 68 bits for data and 15 bits for error correction. Each of the paired devices encodes its hash value into a 2D barcode and then the device authenticating the other captures the barcode on the other device using the camera on the mobile phone and decodes that information. If the information decoded from the barcode matches the one received from the target device, then the device is successfully authenticated. For mutual authentication, each device has to capture and decode the barcode displayed on the other device.

Based on the assumption that the camera phones themselves are not compromised, there are no attacks on SiB. With this method, the number of bits may easily be increased by employing multiple 2D barcodes that can be read in sequence by a digital camera [1]. This means multiple barcodes may be used to get as many bits as are necessary to achieve the desired level of security.

However, most devices do not have cameras as part of their default design. Apart from mobile phones, Personal Digital Assistants (PDAs) do not have a camera embedded in them by default and neither do laptops. Even for mobile phones, those with cameras may have a higher financial cost compared to same models without a camera [5]. Camera equipped devices are also not allowed in high security premises such as military bases and nuclear plants [5]. SiB is also not appropriate when the two devices are not physically adjacent. This is because the two devices need to be physically close to each other in order for one or both devices to capture the other device's barcode.

It is also clear that using this method, human effort is directly proportion to the number of devices involved. For example if we have two devices, the amount of human effort is doubled if each device has to authenticate the other as opposed to only one device being authenticated. The authentication process becomes impractical when we consider more than two devices. For example, if we have three devices and each device needs to authenticate the other two devices, then we will need 6 barcode-capturing actions given two from each device. This will definitely be unbearable to the users as the number of devices increase.

2.2 Loud and Clear

Goodrich et al [5] present "Loud and Clear." Loud and Clear (L&C) uses the audio channel to attain human assisted device authentication. This method divides the hash value into segments of ten bits and each segment is converted to an equivalent integer value which will be matched to an index in a locally stored dictionary of words. The retrieved words are processed to produce a Madlib-like sentence that is syntactically correct but not necessarily sensible. Either one of the devices reads out the sentence while the user checks on the other device or both devices read out the sentences and the user has to detect if there is any difference.

This method relieves the user from the hard work of reading sentences that are syntactically correct but usually not sensible. However, we do not want to assume that the devices under consideration will have speakers that produce good sound quality and that these devices have screens big enough to display complete sentences that have enough entropy. We choose not to expect a lot from these devices in terms of display size or quality of speakers.

The other challenge with this approach is that it is more likely that the user will play the role of just clicking a YES or NO to a prompt like "Are the sentences/sounds the same?" Users are prone to err in such situations because nothing binds them to make sure that they listen attentively and make an informed choice on the presented prompt. It is also possible that a user who does not take security seriously may pick the differences in these sentences but will go on to click "YES" in response to the above question. We feel that, as much as human effort has to be minimised, more human effort is required rather than just selecting YES/NO to ensure no fatal errors [13] occur.

As these sentences are read by devices that cannot adjust automatically in response to the environment, using Loud & Clear in a noisy environment may be challenging and, hence human readable sentences are more desirable.

2.3 Human-Assisted Pure Audio Device Pairing (HAPADEP)

Soriente et al [12] proposed HAPADEP. They proposed a method that enables exchange of keys between devices using the audio channel. The first device encodes its key into an audio codec and plays the resulting audio sequence while the second device records this audio sequence and decodes it to extract the key. The process is repeated so that the second device can send its key to the first device as well. This key exchange is the “transfer phase” [12] in the pairing process. In the “verification phase” [12], each device computes a digest of the cryptographic material received and encodes it into an audio codec. The user is expected to listen carefully to the sounds played on the devices and to indicate a match or lack of it by pressing a button on both devices.

HAPADEP does not require any other channel or communication link apart from the audio channel in order to exchange cryptographic data between two devices. All cryptographic data exchange happens over the audio channel. The devices can exchange this data even when they have no intention of communication immediately.

HAPADEP has two approaches; one where it deals away with the use of a stored dictionary proposed by Goodrich [5] (hence the devices do not require additional memory) and another where it uses a locally stored dictionary similar to Loud and Clear [5]. In the event of an intruder attempting to play Man in The Middle (MiTM) attack, he will not escape detection by serious legitimate users who pay attention to the sound played and the source of it. If we assume that the users involved in the pairing of devices do understand the importance and significance of security and they pay close attention to the sound played on both devices then there is no attack on HAPADEP.

However, just like Loud and Clear [5], this method suffers from the fact that it does not force the user to listen to the audio played. This means that the security of this method entirely depends on the users’ security consciousness and their diligence to listening carefully to the sounds played. This means that a user who does not understand the importance of security or the value of it may not pay attention to the sound played by the devices. Therefore, using this method, it is clear that the user can easily skip the verification stage and merely confirm that there is a match when there is none. Having the users not compelled to play their roles (and play them properly) poses a great danger by which security may be compromised. In this regard, both HAPADEP [12] and Loud and Clear [5] face similar challenges. It is also difficult to use this method in noisy or crowded environments like airports.

The other difficulty (in fact pointed out by the authors of HAPADEP too) is that the playing of melodies approach is neither fully secure nor very usable. This prompted them to experiment with the alternative method of reading out sentences constructed from a locally stored dictionary. This approach was found to be secure and very usable. However, they also pointed out that not many devices are capable of handling text-to-speech tasks.

2.4 Wireless direct Channels

Several methods [3, 4, 6, 16, 23] have been proposed that try to utilise wireless channels. These include the use of Infrared [3, 4, 23], laser light [6] or a Light Emitting Diode (LED) [24] and Near Field Channel [16]. These methods lack human verification which is more critical in ensuring that the devices that are targeted are the only ones exchanging the pairing information. Because users cannot verify the devices communicating, an attack on these methods is possible for an intruder who is in line of sight especially in a device crowded environment [5]. These methods also suffer from device proximity, i.e. the devices authenticating each other can only be at most 20ft (6 metres) apart [12]. The complexity of the pairing process also increases with increase in the number of devices taking part.

2.5 Hash visualization

Perig et al [2] proposed a way of representing meaningless strings into abstract images. The method converts bits into a randomly generated fixed-size image which human users can easily compare. Humans are known to have great skill in remembering images [2]. The proposed variance of this method uses a locally stored fixed database of real photographs instead of randomly generated images.

In the scenarios under our consideration, the hash visualization still poses a danger because the user is not compelled to compare the images. The user can easily skip the comparison stage. It will also have limited ways of exchanging these images between the involved parties since it is difficult to pass images using the audio channel. This means that the user should have this image provided in advance by the source if the source device is not adjacent to the device trying to authentic it.

The other difficult with this method is producing images that are not “near similar” [2], i.e. images that are different but too similar to be easily differentiated by human eye. This will increase the rate of attacks on the protocol since it is possible for two non similar images to appear similar to the human eye. Using a locally fixed database of images is something we think not to be feasible considering the kind of devices we are looking at, i.e. they have limited memory.

Hash visualization may also only be suitable for devices with high resolution displays which we do not want to assume that all devices under consideration in our research possess.

2.6 Wired channel

Other proposals include the use of the wired channel where two devices exchange data through a cable attached to both devices authenticating each other. This method is secure against all possible MITM attacks since data is exchanged only between the connected devices. However, the wired channel can only work if manufacturers of mobile devices agreed to standardise the ports to be used for such. However, even if that was done, we still have problems of dealing with more than two devices. If the use of wired channels was that easy, then we would better just communicate over the wires instead of the insecure wireless channels. It also means that only devices that are close together can authenticate each other.

2.7 Manual Method

Gehrmann et al [23] proposed “manual authentication.” In Manual Authentication, users either compare short strings (between 16 and 20 bits) displayed on the devices or type the short string displayed on one device onto the other device. There are available applications making use of the 4-digit PIN where a user has to enter a 4-digit number into a device. Applications include ATM cash machines, Mobile phone security codes, and Bluetooth devices. While for cash machines and mobile phones the user has to memorise the 4-digit number, we are more interested in the PIN used in pairing Bluetooth devices.

When pairing Bluetooth devices, both devices display a decimal number and the user(s) confirm if the numbers are matching, one device displays the decimal number and the user enters this number into the other device or the user(s) generates and enters the decimal number into both devices. Although users do not encounter significant problems with this approach, it is a well established fact that the Bluetooth protocol is subject to offline guessing attacks [30]. Another weakness with the Bluetooth protocol is that the PIN has to be kept secret.

2.8 Integrity Regions

Cagalj et al [26] proposed an authentication method that depends on integrity regions. They propose that devices authenticating each other measure the average distance of the other device and display the distance on the screen which human users can easily verify. This is achieved by sending authentication information bit by bit and measuring the distance each bit takes to travel from the sender to the receiver. After all information has been sent, the average distance from the sender to the receiver is calculated.

For the exchanged information to be authentic, there is not supposed to be any other devices (apart from those authenticating each other) within the integrity region. This means that, for example, if the distance measure is 30cm then there is not supposed to be any other device within a radius of 30cm from the sending device apart from the intended device.

As long as it is not a device (or generally) crowded environment and the users are able to easily verify that there are no other devices within the integrity region, the method is secure provided the assumption that an intruder cannot pretend to be closer than he is, is held true. In terms of usability, users only need to check the average distance displayed on both devices.

With integrity regions, users are only sure that the information exchanged is authentic by checking and making sure that there are no other devices within the integrity region other than the authenticating devices. The closer the authenticating devices are the easier it is for human users to verify authenticity of exchanged information. However, if the devices are far apart or when there are other devices within the integrity region, users cannot verify the authenticity of the exchanged information. This renders the integrity region method inappropriate for crowded environments and when the devices authenticating each other are far apart.

2.9 Integrity Codes

Cagalj et al [27] also proposed exchange of authentication information using integrity check codes. The strength of this method is on the difficult that an intruder faces to change the codes without being detected. The proposed codes have to be designed in way such that should a single bit be changed, the receiver will reject that information and thereby sensing the presence of an intruder.

The authors of [27] propose a way in which integrity codes can be designed to achieve the property that guarantees that whenever an intruder modifies the code, the receiver will reject it. If this property can be guaranteed, then integrity codes are secure. However, human users cannot verify authenticity of the exchanged information as they have no control on the high bandwidth channel.

2.10 Usability analysis

The only research that focused on the usability of proposed empirical-channel methods was conducted by Uzun et al [13]. Their focus was on methods that require human users to manually compare short strings. In their analysis, they carried out experiments to analyse user experience with 4 digit strings. They tested various methods such compare and confirm, select and confirm, copy and confirm, copy and choose and enter. Their results show the average time it took participants to accomplish their actions for each method. They also highlight a number of interface design issues that affect usability in such scenarios. However, with a four digit key, an intruder has greater than 2^{-14} chance of carrying out a successful attack on such methods. The search space for the digest value is not wide enough to avoid combinatorial attacks.

2.11 Summary of weaknesses and strengths of above methods

The methods presented above can be grouped into two broad categories; those that directly involve human users in the comparison of the digest value and those that do not. The methods that directly involve human users in the comparison of the digest value have an obvious disadvantage of requiring human effort in the process. As a result they require proper analysis of how human users will be involved in order to minimise the occurrence of fatal errors. However, these methods allow human users to verify the integrity of the exchanged information between devices. Human users can also know if there is an intruder interfering with the initial devices' communication.

The methods that do not directly involve human users in the device authentication process have an advantage of not putting any extra burden on the human users. This is a welcome approach but it is at a cost of what we want to ultimately achieve in any security protocol; security. Since human users are not directly involved, they cannot verify the integrity of the exchanged information. It is difficult for human users to detect any anomalies during the run of the protocol. Some of these methods also employ expensive ways by which they want to achieve authentication without directly involving human users. Using digital cameras, for example, will mean having a camera somehow embedded into a laptop or PDA. Others are not suitable for devices with limited capabilities. These include methods that require high processing power, high quality displays, and high quality speakers.

CHAPTER 3

CLASSIFICATION OF SCENARIOS

We discuss the different types of scenarios that may occur in ad hoc networks. We also discuss what modes of comparison are applicable to each of the scenarios considered here. We begin by discussing scenarios that may occur with devices that are adjacent to each other and then with devices that are not adjacent to each other.

3.1 Devices adjacent to each other

3.1.1 Two devices single user

In this scenario, a single user is pairing two devices that are close to each other. The user has to carry out the process of making sure that he compares the digest values independently calculated by each of the devices.

3.1.2 Multiple devices with single user

In a single user with multiple devices (more than two) scenario, a user has to ensure that the digest values independently computed by these devices are matching. For some modes of comparison, the amount of human effort increases with the number of devices involved. For example, if the human user has to copy a digest value displayed on one of the devices to the other devices, the number of times the user has to copy is n-1 where n is the total number of devices.

3.1.3 Multiple devices with multiple users

With each device controlled by a separate user, some modes of comparison are as good as having a single user controlling two devices while others are as bad as having a single user controlling multiple devices. For example, using a digital camera requires that a device captures the barcode displayed on each of the other devices. This means a single user will be active capturing barcodes displayed on the other users’ devices while the other users watch. For methods such as one employing the use of barcodes, the number of comparisons required will be N-1 where N is the size of the group.

For method like compare and confirm, copy and enter, and compare and select, a user can read out his value to the group while each member of the group enters that value into his device (for copy and enter) or compares and selects (or confirms).

3.2 Devices not adjacent to each other

With devices not adjacent to each other, some scenarios do not apply just like some modes of comparison do not apply. For example, we cannot have a single user scenario when devices are far apart and we cannot use the direct connection as a mode of comparison.

3.2.1 Two devices

With two devices not adjacent to each other, the modes of comparison applicable rely much on the users controlling these devices. A more reliable mode is the audio/video channel when one user can read the value displayed on his device while the other user(s) listens and compares with, or enters, the value onto his device.

Using video, one user can see the screen of the remote user and copy or compare the value displayed on that device. The user also needs to see the user controlling the device to make sure that s/he is comparing or copying a value displayed on the intended device. With the video channel, it is also possible to capture a barcode using a digital camera from the image of the other device.

3.2.2 Multiple devices

With multiple devices we assume a broadcast channel, say conference call, a direct channel from each member to each of the other members or from each member to *I* only.

The audio and or video channels are more reliable modes of comparison in this scenario. The audio channel limits the material compared mostly to readable strings where as the video channel can also make barcodes usable. We can also transmit melodies generated from the digest value over the audio channel. Implementation details may differ but the most efficient one may be where one user reads or plays a sound to the other users either to a single user at a time or to all users at the same time (if a broadcast channel is available).

3.3 Level of security

The level of security required provides a different level of scenario classification. We consider the following situation:

3.3.1 Little or no Security required

There are certain situations where little or no security at all is needed by the human user. For example, wanting to print a BBC article from a PDA to a public printer at an airport or pairing mobile phones to play a casual game. This security categorisation is mentioned only to appreciate its existence but has no relevance to any security protocol as no security is required at all.

3.3.2 Medium security

At this level of security, we can tolerate a certain probability of an intruder carrying out a successful attack. These are mainly situations where the cost of carrying out an attack is much higher than the value of information gained and the information gained causes no damage to the human users or equipment involved. Essentially the risks involved in leaking out information are minimal. If a lecturer, for example, has assignments on his laptop or PDA that s/he wants only legitimate students to download through an ad hoc network between his device and students' devices. If an intruder manages to carry out a successful attack on this network, no harm will be done to either the lecturer or students involved since the intruder will only get assignments.

3.3.3 Maximum security

With this level of security, leaking out any information may result in high risks on the people involved, equipment, or a business enterprise. For example, a team of directors discussing top secrets of their organisation are well aware of the implications of leaking this information to outsiders. Outsiders may include people from within the organisation who have no right to have access to this information. Most financial transactions also require this level of security.

3.4 Modes of comparison

The mode of comparison limits the type of material that can be employed to compare such material. The following are the main modes of comparison:

3.4.1 Audio channel

The audio channel mode uses sound as a medium of comparison. This means anything that can be verbally read either by human beings or machines can be transmitted using this mode. This includes melodies, digits, alphanumeric, hexadecimals, sentences, and any strings which may be words, names of people, cities, countries etc.

However, usability puts more limits on the type of material to be transmitted over the audio channel. If human beings are to read these strings so that others could listen and compare with what they have on their devices, then we do not expect users to find it easy reading strings that are difficult to read/pronounce or that have no meaning to them at all.

3.4.2 Digital cameras

If we are to use a digital camera attached to devices as a comparison mode, then we can only compare barcodes. Barcodes have an advantage in that they can encode more data hence giving us a wide range of the size of the digest value that we can use. Digital cameras make use of 2-D because they encode more data as compared to 1-D barcodes. As 3 dimensional (3D) barcodes [31] are being developed, they will provide even more opportunities for encoding more data in a small space.

3.4.3 Compare and Confirm

The compare and confirm mode of comparison involves a manual comparison of strings each displayed on a separate device. It is a manual comparison because human users have to compare these strings without the use of say digital cameras.

The manual comparison mode gives us a wider choice of materials that we can compare. We can compare strings (digits, hexadecimal, alphanumeric, sentences, words, names) and images. Careful selection of strings to be compared is required if we have to reduce the burden on the human users carrying out the comparison. The length and format of the compared string is an important consideration so that the method is usable.

This method eliminates the need for human users to copy any string from one device to another but rather just requires them to visually check whether two strings, images or sounds are matching and then either confirm or reject. Images have an advantage over strings that are not associated to anything (have no meaning). Because the user is only required to visually or auditorially compare two things and press one button for a confirmation or rejection, human effort is highly minimised.

However, this mode suffers from the fact that it entirely puts trust on the users' capability to accurately compare the digest values and diligently press the right button for confirmation or rejection of a match. As experimental results will show, this method is prone to fatal errors meaning that if a intruder is present then his attack may be successful not because of a flow in the protocol but in its use by the human users. Users become accustomed to success and may just press OK on this assumption or absent mindedly.

3.4.4 Copy and enter

This mode is similar to compare and confirm described above except that a string is displayed on one device and then copied to the other device by the user. This method increases the human effort required compared to the above method. However, it greatly reduces the possibility of fatal errors as will be discussed later. The increased amount of work required of the human users, however, is rewarded by the high security the method offers.

While the method seems to be free of fatal errors, we show that it is possible for a fatal error to occur but at a very low probability. Here is how a fatal error in the copy and enter can occur.

If, for example, we have 123 456 789 displayed on one device and the user is required to copy this string into the other device. Then, it happens that an intruder is present and has modified the digest value on the second device giving an output of 123 446 789 as the value of the digest. Taking the worst case scenario, the user copies the value 123 456 789 as 123 446 789 (accidentally) onto the second device, the device will compare this value with what it has locally and will confirm that there is a match. In this way, an intruder succeeds in fooling the device through a mistake committed by the user.

However, if the initial probability of success of an attack on the protocol is 2^{-b}, wrongly copying a digest value from one device to another does, by no means, increase the chance of a successful attack. It still remains 2^{-b}. Implementation details need to take care of the need to have the user copy the string accurately. Experimental results, shown later, revealed the strengths and weaknesses of this method as perceived by participants who took part in the experiment.

3.4.5 Compare and select

This mode of data comparison is similar to compare and confirm. One device displays its digest value (in a user friendly format) and the other device(s) display their digest values together with other randomly generated values of the same format. The user has to select from the other device a value that is the same as the one shown on the device with one value.

This method is different from compare and confirm in a sense that the user in compare and select does not confirm or reject that the two digest values are matching but rather just selects a value that is displayed on the other device and then the device confirms if the two values are matching on not.

The compare and select mode of comparison requires less human effort than the copy and enter and slightly more than the compare and confirm. In terms of the security offered, it offers the same security level as the copy and enter method.

For an attack to be success on this method, the intruder will have to rely on the human users' weaknesses just shown in the two previous methods and also needs to gain some idea on the value of the final digest value so that he can make his final digest value as close to the one displayed by *I* as possible. However, the (S)HCBK protocols have been designed such that it is impossible for an intruder to gain even a slightest idea of what the final digest value will be. As a result, an intruder will only rely on the human users' weakness. Here is how an attack can be successful on this method:

If a device has a different value other than that shown on *I*'s device (possibly a value from an intruder), that device displays that value together with other randomly generated values and request that the user selects the value shown on the other device. If the user makes a random selection without even checking the value, the success of an attack has a chance of N^{-1} where N is the number of values displayed on the device.

However, the selection of the digest value is by no way random as the user has to match one of the values with that shown on *I*'s device. If no value matches, the user rejects or the device will reject if the user chooses a wrong value from the possible choices. As a result, the probability of a fatal error is very minimal and will be pegged at 2^{-b}, which is the same as the overall security of the protocol and hence it does not reduce the security of the protocol.

To minimise human user errors in this method, the display of the digest value together with the random values has to be random so that the human users have no other way of determining the digest value apart from checking the values carefully. If the display of the random values together with the digest values is not random, human users will find other ways of determining the digest value, say it is always second on the displayed list. The random values need to be generated on the basis of the digest value to help have values that seem random even to the human user rather than having values that are very similar or in a worst case scenario a random value that is the same as the digest value.

3.4.6 Direct connection

This mode uses direct connection channels that may not be suitable for transmission of large amounts of data but suitable for the exchange of the digest value. Proposed methods include the use of laser light, infrared, Near Field Channel (NFC), and wired channel.

This mode does not even require the digest to be in any other format (e.g. images, strings etc). It removes the extra burden required to convert the digest value into another format and the human users do not have to do anything except watching the screen of their devices to see if the device confirms that the connection is successfully established or not.

Because these methods are point to point, they can only work well with two devices that are physically adjacent to each other. Attacks are also possible on infrared and laser channels. The wired channel is an impractical method because if it was not, we would then rather communicate over wires than on insecure wireless channels.

3.5 Scenario-mode of comparison matrix

The modes of comparison discussed above only apply to a subset of the scenarios being considered. Table 1 below shows a mapping of the scenarios to applicable modes of comparison.

Scenario	Mode of comparison
Two devices (single user)	Digital cameras, Compare and confirm Direct channels (wireless and wired) Compare and select, Audio channel Compare and enter
Multiple devices (single user)	Digital cameras, Compare and confirm Direct channels (wireless and wired) Compare and select, Audio channel Compare and enter
Multiple devices	Digital cameras, Compare and confirm Direct channels (wireless and wired) Compare and select, Audio channel Compare and enter
Two devices (not adjacent)	Compare and confirm, Compare and select Audio channel, Compare and enter
Multiple devices (not adjacent)	Compare and confirm, Compare and select Audio channel, Compare and enter
Medium security	Digital cameras Direct channels (wireless and wired) Compare and select, Audio channel Compare and enter
Maximum security	Digital cameras Direct channels (wired only) Compare and select, Audio channel Compare and enter

Table 1. Scenarios verses modes of comparison

3.6 Material compared

Each of the above modes of comparison only handles certain types of material. This means that for any of the above modes of comparison methods to work, the digest value needs to be converted to an appropriate format whenever necessary. Table 2 below shows a mapping of the methods to materials that they can handle

Mode of comparison	Material compared
Compare and confirm Compare and select	Digits, hexadecimals, alphanumeric, random art, images, sentences
copy and enter	Digits, hexadecimals, alphanumeric, sentences
Digital camera	Barcodes
Audio channel	Sound
Direct connection	Digest value transmitted in raw form

Table 2. Modes of comparison verses material compared

The visual channel (compare/confirm and compare/select) has the widest range of materials that can be compared while direct channels do not need the digest value to be converted to anything. Some conversions require more processing than others. For example, converting a digest value to an audio sound may need more computation power from the device compared to converting a digest value to a decimal value.

CHAPTER 4

Maximising the security of the protocol

In many instances, a protocol is said to be secure if it is proved secure mathematically. Proving the security of a protocol mathematically is not uncommon among security experts and researchers since it is critical to ensure that the protocol is mathematically secure. However, history has shown that protocols have failed not because they were mathematically insecure but because of other factors that where not considered to play a major role in the implementation and or use of the protocol. We consider those factors in this chapter. We begin by discussing technological factors, followed by human factors and finally, but not the least, reliability of various methods of the empirical channel. We then discuss how each of these relates to one another and how we can find the best comprise in order to maximise security.

4.1 Technological factors

Technological factors address the difficult of compromising the protocol given the available technology. This can easily be addressed by using a digest function that produces a large number of bits that makes an attack infeasible. For example, the chance of a successful attack on the (S)HCBK protocols is 2^{-b}, where b is the number of bits of the digest value. The larger the value of b the stronger or more secure (technologically) the protocol is. This means that to make a protocol technologically secure, we need to make the value of b as large as possible.

Another area that has to be addressed technologically is how the protocol responds to a (perceived) failed attack? If an attack is detected, the protocol should rerun in a manner that minimises the chance of a successful attack taking into account that the intruder may have gained some information in the previous failed attack. The protocol then should run in a retry or more secure mode compared to the previous run. In a retry mode, we want the protocol to be more resilient to attacks. This can be achieved by increasing the number of bits of the digest value.

For each aborted or failed session, the protocol can increase the number of bits of the digest values or use other means by which it becomes harder for an intruder to go unnoticed.

4.2 Human factors

Bruce Schneier (2000) wrote that "Security is only as good as the weakest link, and people are the weakest link in the chain." Security does not only involve mathematical proofs but also other components such as human factors. As a result, the security of a protocol depends on the mathematical proofs just as much as it depends on the human users.

While mathematical proofs will remain reliable and certainly usually correct, human users should not be trusted to handle large amounts of data with the level of accuracy that is required to ensure the desired level of security. As the results of the experiment will show in the next chapter, people are prone to making errors and because of this we want to ensure that, in the protocol, we minimise the risk of such fatal errors by human users.

We also recognise that it is possible for human users to make non-fatal errors. This will prompt the protocol to abort in the absence of an attack. For example, if a user wrongly copies a digest value the protocol will abort when there is no attack at all. This may result in loss of service. Forcing users to re-enter or reselect the value of the digest all the time will result in too much work on the part of human users which may result in low acceptance of any protocol employing such methods.

We suggest that whenever the value entered, selected, or confirmed by the user does not match the one the device has, the user is prompted to reconfirm his/her action by re-entering, reconfirming or reselecting. This should only be done in cases where the values do not match the first time a user enters, selects or confirms. We base this suggestion on the assumption that the users involved are honest. The implementation of the protocol must ensure that no other messages, whatsoever, are shown to human users apart from a mere prompt asking them to reconfirm their previous action. Once the user reconfirms and, if still there is a mismatch, the protocol should abort and users should use a better, and probably expensive or less convenient, background network for the required communication.

When authentication fails, either the digest values are not matching or the long hash values are not matching, there are a number of options that can be sort. We pointed out earlier that the protocol may be rerun in retry mode but it also an option to completely abandon the run or to find an alternative secure network to be used. We discuss these options below.

4.2.1 Retry mode

In retry mode, the protocol should run as before. However, the human compared digest value needs to be longer than one used in the previous run. This is in order to reduce the attacker's chance of carrying out a successful attack in the second run. The long hash may also be increased and the devices carry out the normal comparison on the long hash.

This approach raises a number of questions; what if the device's long hash value does not match with the received information while the user compared digest values match? What if the opposite happens? For cases where both fail or are successful, it is straight forward since a failure for both will be taken as an attack otherwise authentication is successful. These questions are not only for the retry mode as long hash comparison happens in the normal run of the protocol as well.

To address these questions, we need to address the question of which method is more trustworthy under what circumstances between the two. Human users are trustworthy if the value they enter, select or their confirmation matches with what the device has while information exchanged over the insecure channel can easily be changed and cause the comparison to fail or match. As a result, whenever device compared values (long hashes) are matching we still need user compared values to match as well in order to have a successful device authentication.

It is clear that information exchanged over the high bandwidth channel may not be authentic and human users sometimes may make mistakes while copying, selecting or just confirming digest values. This leaves us with a question of deciding whether to accept device confirmed values when user confirmed values do not match or not. In a situation like this, we will give the user a benefit of doubt and prompt them to reconfirm their actions. If the values still do not match, then it is an attack on the protocol.

Table 3 below shows the scenarios that arise when the protocol is run with proposed actions to be taken for each scenario when it occurs. There has to be a limit on the number of retries that the protocol can go into since rerunning the protocol in retry mode does not ensure that the intruder is not present. As a result, if the protocol has to be run in retry mode, it should only run once. Rerunning the protocol several times only increases the amount of human effort required to compare the digest values.

Long hash	Digest value	Authentication
Match	Match	Successful
Match	No match	Users reconfirm action
No match	Match	Failed
No match	No match	Failed

Table 3. Matching scenarios

However, this may be an easy way for an intruder to carry out a Denial Of Service (DOS) attack and successfully deny legitimate users a service. In situations where the devices are adjacent to each other, it is highly probable that legitimate users may be able to spot who is causing interference on their authentication process. However, in crowded environments and devices that are not adjacent to each other, it is harder to physically spot an intruder by legitimate users.

4.2.2 Aborting the protocol

The second option is to abort the protocol all together when the digest values do not much the first time. This approach is based on the assumption that if an attacker is present in the initial run of the protocol, chances are that he will still be present even in the second run and cause a mismatch of the digest values.

4.2.3 Using background network

This involves using an alternative or background network through which communication can take place. This may be removable memory media, a direct wired network, or any other way in which information may be exchanged between the devices without using the normal high bandwidth channel. This approach is likely to be expensive and probably less convenient. As a result, this may be application dependent.

This has the advantage that it will bypass the intruder by avoiding the normal high bandwidth channel and hence even if the attack is still present when the communication takes place using this channel, he may not have control on the alternative network.

4.3 Reliability

We discuss reliability in terms of risks involved on the empirical channel. With two devices adjacent to each other, we assume that an intruder has no control over this channel while for devices that are far apart an intruder can easily eavesdrop and, in certain channels, even modify messages.

Over the empirical channel, we are not worried about secrecy but authenticity. As long as a user at the other end is sure that a message *m* is from the user s/he is communicating to, then the channel is secure. Because of the authenticity required we assume that phone conversations are secure since both users at the end the phone line will be in a position to verify that they are talking to someone whose voice they recognise. The same applies to video communications; each user sees the other user(s) and hence they are sure that the values they are receiving are from that particular user and none else.

As users can only participate in only one run of the protocol at a time, a replay attack is not possible even if an intruder recorded a voice of a legitimate user reading his digest value to the other members of the group. This is because for a replay attack to be successful, an intruder will need to bring the other user(s) to a stage where they have to compare the digest value and then play the recorded message. This also means that the digest value has to be the same as when the message was recorded for this attack to be successful.

4.4 Technological factors versus human factors

A trade off has to be found that will find the best mix of technological and human factors that maximises security while the protocol remains highly usable. In the (S)HCBK protocols, we focus on values that are between 16 and 32 bits. However, we would like to maximise as much as we can on both technological security and human factors security. This means that if the protocol remains highly usable when we use a digest value of 160 bits, then we will use a digest of that many bits.

Usability plays a major role in deciding how many bits we have to use for our digest value. For methods that do not require human users to manually compare the values, the limit on the number of bits used depends on the mode of comparison. For example, due to a limitation of mobile phone screens, an 83-bit sized barcode is the maximum we can use for our digest value. It is also possible, however, to use digest values of more than 83-bits by using multiple barcodes that can be captured in sequence. We discuss how we can maximise technological security and at the same time minimising human effort when we discuss the results of an experiment in the next chapter.

CHAPTER 5

Experimental results

5.1 Details of the experiment

In order to get the views of prospective users of the protocol, an experiment was carried out on the various modes of digest value comparison to determine their usability. In the experiment, we had one independent variable and two dependent variables. The independent variable is the representation of the digest value. This may be numeric, alphanumeric, words, sentences, images etc. We restricted ourselves to only textual representations which include all the above except images for an obvious reason that to have enough images that can cover a wide space of the digest, we need a lot of memory which we do not expect in many of pervasive devices. Other reasons for avoiding images is that we need high resolution displays if images have to be displayed in clear and distinguishable manner to the users while randomly generated images suffer from the problem of near-similarity.

The dependent variables are time and user errors. The time taken to compare two strings and user errors depend on the representation and in some cases on the length of the digest value. We needed to capture the time it can take a user to finish comparing strings for variable representations and methods. User errors are a critical dependent variable in this experiment as this gave us an idea of how secure or insure a particular representation is. Two types of errors were captured; fatal errors and non-fatal errors. Capturing the time taken to accomplish a task and the number of fatal and non-fatal errors provided us the much needed quantitative data.

We carried out the experiment using two devices simulated by software on the screen of a computer. The simulated devices are much similar to a conventional mobile phone than anything else in terms of the screen size and type of keypad. The only method of input to these devices was through the mouse. Figure 2 below shows a screen short of the simulated device.

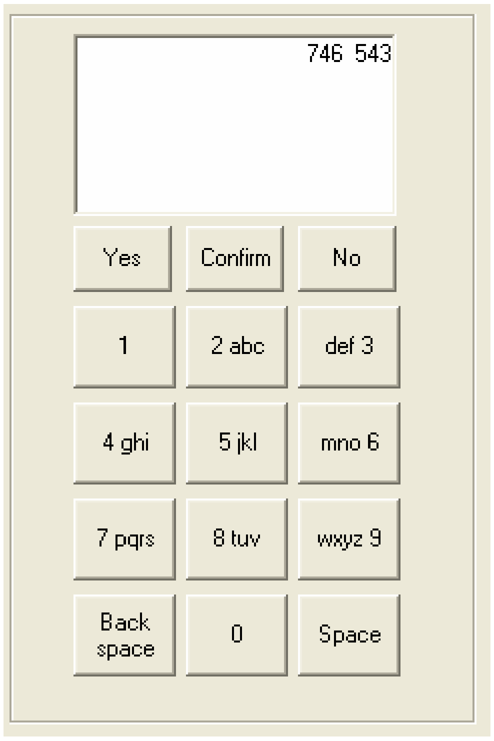

Figure 2. Simulated device

Twenty participants took part in this experiment and they came from a wide range of backgrounds from Diploma students to DPhil students. However, we admit that there was no greater variation in terms of age of these participants as all of them were in the range of 18 to 35 years old.

Each participant was presented with a set of tests which s/he can do at his/her own normal pace. Participants were asked to accomplish the tasks at their own normal working pace and level of concentration. No mention was made that we were capturing the time they were taking to accomplish the tasks. This was a deliberate decision to prevent participants from working at accelerated speeds so as not to be seen as ‘slow’. There was no emphasis also on the accuracy that was to be attained. Participants had to work at their normal rates and accuracy to get a true reflection of what may happen when they know they are not monitored or no one will ever review their actions later.

The first ten participants had twenty tasks to accomplish for each type of material compared for each of the three modes of comparison. For the compare/confirm method, we had on average 4 pairs (out of 20) of strings that were not matching and the rest were matching. The order of these tasks was changed from participant to participant. This was because some people start strong and towards the end they get tired and their inputs may not reflect their normal working rate capacity while others start at a slow note and gradually gain momentum and more concentration. These factors were also important to be incorporated into the experiment. Owing to the material compared and mode of comparison, some tasks were accomplished quite faster compared to others as results will show.

Because of the length of the overall time it was taking for participants to accomplish the tasks, the next ten participants were tested on ten tasks for each type of material for each mode of comparison except copy and enter where they were tested on six tasks for each.

Figure 3 below shows a screenshot of a scenario where a user is matching hexadecimals (48 bits) using the compare and select method.

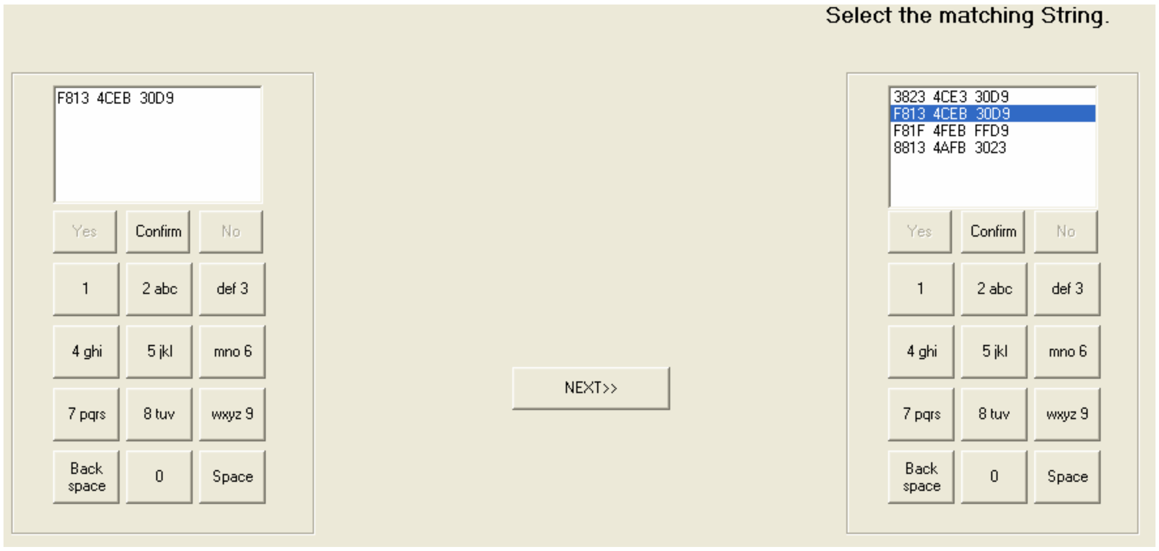

Figure 3. Compare and confirm of hexadecimals

In figure 3 above, the participant had to select one string from the four strings shown on the right hand side which matches the one shown on the left hand side device and then click ‘confirm’. The participant will then click the NEXT>> button to move to the next task. At the end of each set, say comparing 48-bit hex numbers, a participant is presented with the screen shown in 4 below to give their feedback. This provided us with qualitative data. A participant had to rate whether the method was very usable, usable, or unusable and whether it was very difficult, difficult, easy or very easy.

It is important to note that the numbers (both decimals and hexadecimals) are split into groups of 3 and 4 for 6 digit-long and 12 digit-long numbers respectively. This helps a user to deal with smaller bits of information rather than dealing with a full number at once. This will be more difficult especially for the 12-digit-long numbers. A human brain is said to process 7 items at once [21]. Splitting the numbers into smaller groups is, hence, recommended.

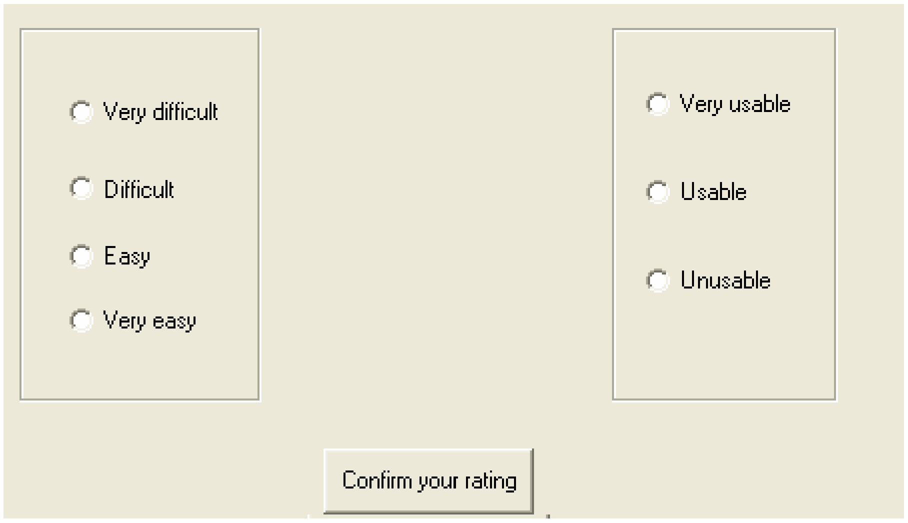

Figure 4. Method rating

The two ratings sound like they are one and the same but they are not. It is also easy to think that one implies the other. However, a method that is said to be very difficult does not necessarily imply that it is unusable. For example, some participants felt some methods are very difficult and at the same time felt that they are still usable.

Some modes of comparison did not cover all types of material that can be used for that mode. For compare and confirm, a user was presented with decimals (6 and 12 digits long), hexadecimals (6 and 12 characters long), sentences, words (three and six words), and country names (three and six countries). Compare/confirm is the only mode of comparison that made use of all the types of materials considered in this experiment for reasons being that it is easier to use all the types of materials compared with other modes of comparison.

For compare and select, a user was presented with decimals and hexadecimals both 12 characters only. We only utilised decimal and hex numbers for this mode of comparison because we can easily make deductions for this method based on the 12 characters long numbers without using the 6 character numbers. This also reduced the number of sets of tasks a participant had to accomplish and subsequently reducing the overall time a participant took to finish the whole exercise. Because the screens of the devices were quite small, we did not use sentences, words, or country names for this method as a single sentence, set of words or country names normally covered more than a single line on the device screen.

For copy and enter, a user was presented with decimals and hexadecimals (6 and 12 characters long for both) only. We did not use words, sentences, or country names for this mode of comparison for an obvious reason that it is impractical to expect users to type complete words or sentences on such devices with a limited keypad such as the one used in this experiment.

5.2 Summary Results

Table 4 below shows a summary of the results of the experiment in terms of errors and the average time a user took to accomplish each a single task for each method. As defined earlier, fatal errors are errors that cause one or more devices to agree that the digest values are matching when they are not while non-fatal errors are those that cause one or more devices to abort the authentication process when the digest values are matching.

The errors shown are in percentages and are calculated as follows; the percentage rate of fatal errors is the total number of fatal errors divided by the total number of non-matching strings for that particular method. For example, in the compare and confirm method, if we have a total of 10 strings to compare of which 3 are not matching and a user confirms 1 of the 3 that they are matching then the percentage error rate is 33.33%. For non-fatal errors we take the total number of matching strings and divide by the number strings that a user thinks are not matching converted to percentage.

Before discussing the results in detail, we can point out a few general points prominent in table 4. We note that, generally, the compare/confirm method has the least completion time and the copy/enter method has the longest time. Doubling the number of bits doubles the time it takes a user to complete the comparison process. Because of the limited keypad used in this experiment, it is expected that entering digits will require less time compared to entering the same length of a hexadecimal given that digits are defaults on this keypad.

It is also notable that increasing the length of a string does not affect the user error rate, at least within the limits of our experiment. This is true for all types of string that used variable bit length.

Bits	Method	Average Time	Fatal Errors	Non-fatal errors
20 bit digits	C/Confirm	4	20.4	0
	C/Enter	11.8	N/A	0.5
40 bit digits	C/confirm	9	21	0
	C/Enter	25.8	N/A	0
	C/Select	9.1	N/A	1
24 bit hex	C/Confirm	3.8	8.5	0.6
	C/Enter	22.6	N/A	0.5
48 bit hex	C/Confirm	9	22.9	0.6
	C/Enter	42.4	N/A	1.7
	C/Select	7.3	N/A	1
33 bit words	C/Confirm	4	16.1	0
66 bit words	C/Confirm	6.7	25.4	0
24bit country names	C/Confirm	4	2.1	0
48 bit country names	C/Confirm	8.4	9.5	0.5
41 bit sentences	C/Confirm	5.6	38.8	0

Table 4. Average time and error rates

Table 5 shows the user ratings for each method. The abbreviated columns correspond to, from left to right, Very difficult, Difficult, Easy, Very Easy, Very Usable, Usable, and Unusable correspond to user ratings.

Bits	Method	V/D	D	E	V/E	V/U	U	U/U
20 bit digits	C/Confirm	0	0	6	14	9	11	0
	C/Enter	0	2	11	7	4	16	0
40 bit digits	C/confirm	0	0	11	9	6	14	0
	C/Enter	0	3	10	7	2	17	1
	C/Select	0	1	11	8	7	13	0
24 bit hex	C/Confirm	0	1	9	10	8	12	0
	C/Enter	0	8	11	1	2	18	0
48 bit hex	C/Confirm	0	3	13	4	7	13	0
	C/Enter	1	11	6	2	0	13	7
	C/Select	0	0	9	11	9	11	0
33 bit words	C/Confirm	0	0	12	8	8	12	0
66 bit words	C/Confirm	0	0	12	8	6	14	0
24bit country names	C/Confirm	0	1	11	9	6	14	0
48bit country names	C/Confirm	0	0	15	5	5	15	0
41 bit sentences	C/Confirm	0	0	12	8	8	12	0

Table 5. User ratings

5.3 Compare and confirm

This method was generally rated to be the easiest with least completion time for all the different types of strings used in the experiment. However, the results also show that it has the highest fatal error rate compared to the other two methods. This is mainly attributed to the definiteness of the user’s input. Sometimes users would confirm that the digest values are matching while they actually meant otherwise.

Over the various types of material with varying lengths, fatal errors ranged from 2.1% to 38.8% while non-fatal errors ranged from 0% to 0.6%. In comparison with the experiment carried out by Uzun et al [13] using this method on 4-digit numbers, their fatal error rate was 20%. A fatal error rate of 2.1% is unacceptable in a security application. As a result, this mode of comparison is not acceptable despite the ease of use pointed out by the participants.

5.4 Copy and enter

Participants in the experiment preferred decimals to hexadecimal for an obvious reason that digits were easy to enter from the kind of keypad provided as opposed to letters that are part of hexadecimal numbers. Doubling the length of a digest value to be copied doubled the time it took a participant to complete copying. For example, copying a 20 bit decimal number has an average time of 11.8 seconds while copying a decimal number of 40 bits has an average time of 23 seconds.

This method has an advantage of not succumbing to fatal errors. As pointed out earlier, a fatal error for this method has a chance of 2^{-b} to succeed. This is the same as the chance for an intruder carrying out a successful attack on the protocol itself and hence this method by no means reduces the security of the protocol. From the experiment, the highest non-fatal error rate was 1.7% (for 48-bit hexadecimal numbers).

Going by the inputs of participants, this method is highly usable. Those spoken to expressed that though entering long strings (especially hexadecimal digits) was challenging, they still felt it was usable and if security was guaranteed by using such a method they can use it.

Of the 20 participants only 3 indicated that entering 12-digit long decimal numbers was difficult while 10 indicated it was easy and 7 indicated it was very easy. Nineteen participants indicated the method was usable while only 1 felt it is unusable. Only 2 participants indicated that entering 6-digit decimal numbers was difficult while no one indicated that the method was unusable.

For hexadecimals numbers, a total of 12 participants indicated that entering 12-characters long hex numbers was difficult while 8 felt the same for 6-character long hex numbers. Seven participants felt entering 12-characters long hex numbers was an unusable method while no one felt the same for the 6-character long hex numbers.

From the above results, it is clear that with a limited keypad such as the one used in this experiment, it is easier to enter decimal numbers compared to hexadecimal numbers since each decimal digit has its own key and users do not have to press more than once to get the required digit. This led to participants favouring decimals rather than hexadecimals. The length of decimal numbers does not matter much as the results show that 2 participants felt entering 6 digits was difficult while 3 felt the same for 12 digits. Eleven felt it was easy to enter 6 digits and 10 felt the same for 12 digits while 7 felt it was very easy to enter 6 and 12 digits. Apart from the completion time that doubled, the length of decimal numbers did not have much impact on the participants.

5.5 Compare and select

In general, users had no major complaints about this method either verbally or otherwise. They felt it was easier than copy and enter and much more like compare/confirm. With this method, only decimals and hexadecimals of length 12 were used in the experiment.

Of the 20 participants, only one indicated that this method is difficult for decimal numbers while no one felt the same for hexadecimal numbers. There was also no one who indicated that the method was unusable for either decimals or hexadecimals. There is no major difference in terms of time taken to accomplish the task of comparing using this method and using the compare/confirm method. The average time for accomplishing a task using this method is 7.3 seconds for hexadecimals and 9.1 seconds for decimals.

It has a non-fatal error rate of 1% for both decimals and hexadecimals. Fatal errors have the same chance of occurring on this method as the intruder's chance of carrying out a successful attack, 2^{-b}. Both this method and the copy/enter do not reduce the security of the protocol in anyway. In terms of usability, participants favoured the compare/select method rather than the copy/enter.

CHAPTER 6

Implementing the 1-bit empirical channel

6.1 direct human interaction

In the formal presentation of the HCBK protocol in chapter 1, we see that message 2b is transmitted over the empirical channel. This is a 1-bit message as human users may only need to indicate whether they got message 2a from *I* or not. We present, in this chapter, proposed implementation of the empirical channel on message 2a.

The easiest approach would have been to avoid message 2b so that the protocol may run without requiring human effort until message 4b. This may be achieved using time limits that may be enforced on the initiator device and slave devices. However, message 2b is necessary for security and avoiding it allows attacks on the protocol to be successful. As a result, the approach to be taken here is not too avoid message 2b and thereby eliminating human effort but to minimise human effort and ensuring that human users are enforced to carry it out.

Different scenarios may require different approaches of implementing the channel of message 2b. For example, it seems easier to implement this channel in a pair-wise authentication rather than in a big group. Below we propose a generalised implementation that may cover most of the scenarios.

Upon sending message 2a, *I*’s device prompts if all the other users received the message and waits for response from *I*. Each device that receives message 2a indicates to the user that it has received message 2a and the user communicates this to *I*. After everyone else has confirmed, *I* confirms to his/her device that every device received the message and the device proceeds to sending message 3.

A modified approach of the above may be that instead of devices indicating that they have received message 2a, they only indicate when they have not. In this way, devices do not display any messages on the screen except those that have not received the message. This means that only *I* will have a message on the screen asking if the other members received message 2a.

In small groups, say less than 10 members, it is easier to ask if everyone has received message 2a than in a large group as the responses in a large group may overshadow those who have not received and go unnoticed by *I*. Because of this consideration, it is better for *I* to ask for those who have not received message 2a than asking for those who have. This is because we are interested in knowing who has not received message 2a and not those who have. In scenarios with devices adjacent to each other, this may be done by raising hands or by talking to *I* to indicate that one has not received message 2a while with devices not adjacent, this may only be achieved by users using any of the available channels to communicate back to *I*.

Following the above argument that we are more interested in finding any member of the group who has not received message 2a rather than those who have, the question is does *I* get a prompt asking for those who have received or a prompt asking for those who have not? In order to match *I*’s mind perception at such a time, the device should as well ask something like “Has anyone NOT received the message?” This will allow *I* to answer the prompt with the same answer s/he got from the group. Hammer et al [32] have shown that people use positive constraints more intuitively although they fail to use them perfectly and the use of negative constraints enables a less natural but potentially more accurate categorisation strategy. As a result, it is perfect in both instances to ask question that are negating by the use of NOT as shown in the above question.

This method has a number of issues that need addressing. First, *I*'s device may wait forever to receive a confirmation that all other devices received message 2a. This may be solved by putting an upper bound on the time that *I* has to confirm to the device that the other devices received message 2a. Once that time elapses, the protocol times out and aborts.

The second issue is that it is possible that *I* may confirm that the other devices have received message 2a when that is not the case. This is possible especially in large sized groups where it is difficult to know who is attentive and who is not. This requires some kind of enforcement that may put a demand on the users to check the devices and report to *I* if they get the message that they have not received message 2a.

6.2 Summary

We see from the above section that human effort is a necessity on message 2b as this is required for security. In this regard, the SHCBK (Symmetrised HCBK) is more favourable than the HCBK as no extra human effort is required in addition to comparing digest values.

The major challenge is enforcement. How do we make sure that users check their devices and respond to *I* accordingly? How do we make sure that *I* takes the time to ask the members of the group and gives his/her response to the device accordingly? These are challenging questions that we have not answered in this research but which we wish to answer in our future research. For now, what we have is sufficient for the protocol to work in a robust way based on the assumption that users will not overlook message 2b.

CHAPTER 7

Summary and Conclusion

Having discussed and analysed the various methods that are candidates for the implementation of the empirical channel, it is clear that no method emerges to be the best in terms of the security offered as well as its usability. It has been shown that different methods have different strengths and weakness that are related to the scenario in which they are applied. For example, it is impractical to choose the digital camera mode of comparison on devices that are not adjacent to each other where as if applied to devices that have inbuilt digital cameras and are adjacent to each other, it may be the best method. The recommendations in this chapter are given based on the theoretical as well as experimental analysis presented in the previous chapters.

7.1 Recommendations

The results of the experiment show that both the copy/enter and compare/select methods are both secure and usable. They both offer 2^{-b} chance for an intruder to carry out a successful attack. This means that they by no way reduce the overall security of the protocol. The copy and enter method is generally more difficult than compare and select method. With a limited keypad similar to one on mobile phones where two or more letters share a single button, it is more difficult to enter hexadecimal numbers than decimal numbers.

As a result of the above, we recommend that the compare/select method is the best in terms of usability and the security offered. We recommend the copy/enter method only in devices where it is difficult to display four values on the screen. In these devices, the copy/enter method will be the alternative.

With either of the above method in use, all device scenarios are covered. A single user can easily copy/enter or compare/select the digest value on the devices that are being paired. In a multi-user environment, one human user, say *I*, may read out the digest value to the other members of the group who may then enter or select that value on their devices.

It is also easy to transfer these values for long distance communication. This may be done over the phone where users can recognise each other's voice or using video communication and users can see each other even the displays of the other users' devices and read the values displayed.

In situations where the human user entered or selected a value that does not match the digest value on the device, the user should be given a benefit of doubt and asked to reconfirm their action. Hoping it was a non-fatal error, this time the human user will be more careful and enter or select the correct value. However, if it happens that the value is still not matching with the digest value, the protocol should abort and the run abandoned. This is because if the intruder was present in the previous run of the protocol, chances are that he will still be present in the second run as well. Hence the protocol run can be abandoned after the first perceived failed attack.

If at all the protocol is rerun after the perceived failed attack, it should run with a longer digest value. This is to avoid increasing the chance of an intruder carrying out a successful attack. The protocol should be implemented such that it is only rerun once. If the rerun fails also, then it has to be abandoned at that stage.

In certain applications, abandoning the run of the protocol may call for the need of an alternative mode of communication. However this is application dependant.

7.2 Future work

We intend to pursue further a number of questions that this project has not answered, specifically on the empirical channel of message 2b in the HCBK protocol. We have shown how difficult it is to ensure that users check for message 2a and respond to *I* if they have not got the message. We intend to explore this further by exploring way in which it may be achieved. It may involve modifying the protocol in order to achieve the desired goal.

7.3 Conclusion

In this project, we have assessed usability of a number of methods that were candidates for the implementation of the empirical channel in the (S)HCBK protocols. We have presented the weaknesses and strengths of these methods with a focus on their easy of use and the amount of security offered. We have avoided employing more expensive methods that may require extra devices or the less secure methods that lack human user verification of the integrity of the exchanged information.

We have presented the various scenarios in which the (S)HCBK protocol may be applied and the reliable channels that human users involved in the run of the protocol can use. We have also proposed a an implementation of the empirical channel in message 2b of the HCBK protocol may be implemented.

The project has achieved its goal of analysing human factors in the (S)HCBK protocols. It has successfully identified human factors that may lead to compromising the overall security of the protocol despite the protocol itself being robust. Many security protocols have failed not because they were not technically robust, but because the designers did not take into account human factors. As security is a chain of interconnected parts, humans are part of that chain, and if they are not taken into consideration at the design or implementation stage of the protocol, they may just be that ‘weakest link’. We have pointed out how the protect should respond in failed attack and presented the security of the proposed empirical channels. The project has also identified new avenues of research that will enhance the HCBK protocol specifically on the empirical channel of message 2b.

8. References

[1] McCune, J. M., Perrig, A., and Reiter, M. K. 2005. Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy* (May 08 - 11, 2005). SP.

[2] Adrian Perrig and Dawn Song. Hash visualization: Anew technique to improve real-world security. In Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC ’99), pages 131–138, July 1999.

[3] Dirk Balfanz, D.K. Smetters, Paul Stewart, and H. Chi Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In Symposium on Network and Distributed Systems Security (NDSS ’02), February 2002.

[4] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In Security Protocols, 7th International Workshop, 1999.

[5] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human-verifiable authentication based on audio. In ICDCS ’06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, 2006.

[6] Rene Mayrhofer, Martyn Welch, "A Human-Verifiable Authentication Protocol Using Visible Laser Light," *ares*, pp. 1143-1148, The Second International Conference on Availability, Reliability and Security (ARES'07), 2007.

[7] N.M. Haller. The S/KEY one-time password system. In Proc. of the ISOC Symposium on Networks and Distributed Systems Security, 1994

[8] A. Madhavapeddy, D. Scott, R. Sharp, and E. Upton. Using camera-phones to enhance human-computer interaction. In *Proceedings of Ubiquitous Computing (Adjunct Proceedings: Demos)*, 2004.

[9] J. S. Shapiro, J. Vanderburgh, E. Northup, and D. Chizmadia. Design of the EROS trusted window system. In *Proceedings of the USENIX Security Symposium*, pages 165–178, 2004.

[10] Neil Haller, Craig Metz, Philip J. Nesser II, and Mike Straw. *RFC 2289: A One-Time Password System*. Internet Activities Board, February 1998.

[11] A.W. Roscoe and L.H Nguyen. *Efficient group authentication protocols based on human interaction.* Proceedings of ARSPA 2006

[12] C. Soriente, G. Tsudik and E. Uzun. HAPADEP: Human-Assisted Pure Audio Device Pairing

[13] Uzun E., Karvonen K., and Asokan, N. *Usability Analysis of Secure Pairing Methods*, Nokia Research Centre. Technical paper, 2007. Accessible at http://research.nokia.com/tr/NRC-TR-2007-002.pdf

[14] Cynthia Kuo, Jesse Walker, and Adrian Perrig. *Low-cost Manufacturing, Usability, and Security: An Analysis of Bluetooth Simple Pairing and Wi-Fi Protected Setup*. Usable Security (USEC), February, 2007.

[16] Bluetooth Special Interest Group. Simple Pairing White Paper. August 2006. Available at: http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf

[17] Near Field Communications Forum. Available at: http://www.nfc-forum.org/home

[18] WiFi Alliance Protected Setup. Announcement August 2006. Available at: http://www.wifi.org/news/pressrelease-081606-WiFiProtectedSetup/

[19] Comparative Usability Framework Project. Available at: http://sconce.ics.uci.edu/CUF

[20] Juola, P. and Zimmermann, P. *Whole-word phonetic distances and the PGPfone alphabet* Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on, Vol.1, Iss., 3-6 Oct 1996 Pages:98-101 vol.1

[21] C. Gehrmann, C. J. Mitchell, and K. Nyberg. *Manual Authentication for Wireless Devices*, 2004. RSA Cryptobytes, Vol. 7, No. 1.

[22] George A. Miller, *The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information*, Psychological Review, 63 (1956), 81-97

[23] Neil Haller. *RFC 1760: The S/Key One-Time Password System*. Internet Activities Board, February 1995.

[24] L. M. Feeney, B. Ahlgren, and A. Westerlund. *Demonstration abstract: Spontaneous networking for secure collaborative applications in an infrastructureless Environment:* International conference on pervasive computing (pervasive 2002). 2002.

[25] Nitesh Saxena, Jan-Erik Ekberg, Kari Kostiainen, and N. Asokan. *Secure Device Pairing based on a Visual Channel.* In 2006 IEEE Symposium on Security and Privacy, 2006.

[26] Roscoe, A.W., Human-Centred Computer Security, 2005

[27] Srdjan Capkun and Mario Cagalj *Integrity Regions: Authentication Through Presence in Wireless Networks* ACM Workshop on Wireless Security, WiSe 2006

[28] Cagalj M., Capkun S., Rengaswamy R., Tsigkogiannis I., Srivastava M., and Jean-Pierre Hubaux, *Integrity (I) codes: Message Integrity Protection and Authentication over Insecure Channels*, in Proceeding of the IEEE Symposium on Security and Privacy (S&P), 2006

[29] Schneier, B. 1995 *Applied Cryptography (2nd Ed.): Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, Inc.

[30] http://en.wikipedia.org/wiki/Dolev-Yao_threat_model

[31] M. Jakobsson and S. Wetzel. *Security Weaknesses in Bluetooth*. CT-RSA 2001, LNCS vol. 2020, Springer-Verlag, pp. 176-191, 2001.

[32] Zhu H., Zhou J., Li H., Su, J. and Li, X., *3D barcode preprocessing scheme based on image recognition.* SPIE Vol. 4875, p. 651-655, Second International Conference on Image and Graphics, Wei Sui; Ed., 2002

[33] Hammer, R. and Hochstein, S. *Category Learning from Equivalence Constraints.* XXVII Conference of Cognitive Science Society (CogSci2005).