The missing link: Human Interactive Security Protocols in mobile payment

Chen Bangdao, A.W.Roscoe, Ronald Kainda, L.H. Nguyen

Oxford University Computing Laboratory
{Bangdao.Chen, Bill.Roscoe, Ronald.Kainda, Long.Nguyen}@comlab.ox.ac.uk

Abstract. A new family of protocols, based on communication over human-based side channels, permit secure pairing or group formation in ways that no party has to prove its name. These protocols are particularly suitable for authentication on mobile phones where PKI or trusted third party solutions are not practical to cover all scenarios. Rather, individuals are able to hook up devices in their possession to others that they can identify by context. By using one of these Human-Interactive Security Protocols (HISPs), we present a new design of mobile payment system to improve on the security of existing solutions, while providing a reliable and ubiquitous foundation for mobile security in general.

1 Introduction

This paper builds upon the work in [1] and describes a radically new model of e-payment in which the human payer first gains a secure connection with the payee via a HISP (Human-Interactive Security Protocol). This assures the payer that his or her device is connected to the intended party, enabling the remainder of the transaction to be performed electronically.

In 2008 Mobile Money Summit, a conference dedicated on mobile payment, a report [2] revealed the number of deployed Mobile Payment Services (MPSs) reached 120 in 2008, more than double the number in 2007. And they predicted an increasing trend towards MPS on the world market and the growing importance of using it as a market opportunity and as a poverty alleviation tool. One of the most popular implementations is Safaricom M-PESA System in Kenya, a MPS based on STK (SIM Application Toolkit) which mostly uses SMS[1] (Short Message Service) as the service bearer, and nearly two million users had registered with it within a year of its nationwide roll-out [4, 5]. Similar mobile payment projects have been initiated in other developing countries as well, for example, in Philippines, where four million customers had signed up for Smart Money offered by mobile operator SMART [4] by 2008. MPS also shows a growing popularity in developed countries. For example, in Japan, NTT DoCoMo launched "osaifu-keitai"(mobile wallet) in 2004 and within one year of its roll-out, 20 million DoCoMo users were equipped with this function and 1.5 million activated

[1] M-PESA in Tanzania is based on USSD (Unstructured Supplementary Service Data), a communication service similar to SMS.

the credit card functionality [6]. It is based on the use of contactless IC cards[2], which is similar to the Oyster Card used in the London underground. In 2008, 608000 retailers accepted DoCoMo's mobile money [3]. The world's biggest mobile phone manufacture Nokia launched a global project on MPS in 2009 called Nokia Money and started its first nationwide roll-out in India in Feb. 2010 [7]. Bank of America, JPMorgan Chase and other banks have published their own MPS applications as well.

Despite these developments, the MPS market is currently at a pre-standardisation phase; various industries and consortia are competing to form a dominant standard [8]. In recent years, several high profile consortia have been formed to facilitate access to finance on mobile platforms. Examples of such consortia are Mobile Money and Mobile World. An announcement in 2009 by the UK Payments Council that cheques are to be phased out by 2018 has heightened the need for a secure replacement payment system and can be an incentive to the development of MPS and possible standardisation at national level.

MPS mainly falls into three categories [2]: person-to-person money transfer, person-to-business payment, and Mobile Banking (access bank accounts, withdraw, deposit, or transfer money between accounts). It is clearly desirable for Mobile Banking to be extended to allow convenient person-to-person and *ad hoc* person-to-business payment. We observe that there are three main problems in most current MPSs:

A. Dependence on Public Key Infrastructure (PKI) and Wireless Application Protocol (WAP)[3]. PKI's reliance on unique naming is impractical in mobile payments where interactions are highly *ad hoc*. On the other hand, WAP 1.x (a common version on most mobile phones) suffers a serious security problem [10] while its improved version is only available on latest mobile phones.
B. Dependence on NFC [9]. NFC requires hardware infrastructure support from mobile phone companies and only a small fraction of existing mobile phones have been enabled with this technology. This approach is, therefore, not a convenient and ubiquitous foundation for MPS. Its security based on locality is questionable since malicious parties can sometimes get close to honest devices/parties. NFC, by its very nature, cannot be used at a distance as in on-line payment.
C. Dependence on the security of Mobile Networks (GSM, 3G). Data sent via Mobile Networks can be revealed always because they are not encrypted. For example, using GSM Cellular Interceptor[11, 12], which is a sophisticated and powerful device that can intercept GSM traffic nearby, the criminal can collect users' identities and users' account numbers, passwords or PINs easily.

Our proposal aims to improve security, convenience, and privacy of existing payment methods. It is designed to reduce the dependence on PKI, WAP, and NFC; and it does not rely on the security of Mobile Networks. In particular, it facilitates peer-to-peer payment where, unlike credit card payment, the

[2] The underlying technology of Docomo's contactless IC card is based on Near Field Communication (NFC).

[3] The security of WAP is mainly based on the use of a PKI.

payee needs no special certificate or privilege. Our proposal works similarly for person-to-business and peer-to-peer payments. Therefore we will only distinguish between these two when talking about the special characteristics of one or the other.

This paper is organized as follows: Section 2 gives a background on the underlying theory of our design. Section 3 discusses current MPS and related research highlighting the problems we have mentioned earlier in this section; Sections 4 and 5 discuss factors necessary for designing MPS. We then present our prototype implementation in Section 6, and conclude and discuss future work in Section 7.

2 Background

Bootstrapping security among devices where PKI or shared secrets are unavailable is, at best, challenging. Over the past few years, however, a new family of authentication protocols that are based on human trust and interaction have been introduced. These protocols are often referred to as Human Interactive Security Protocols (HISPs). They use two kinds of channels: a high bandwidth channel (denoted $\longrightarrow_N$) subject to the Dolev-Yao attack model [26] and a low bandwidth (empirical) channel (denoted $\longrightarrow_E$). Due to its limited bandwidth, the empirical channel transmits a Short Authentication String (SAS) that is used to authenticate data exchanged over the insecure high bandwidth channel. Examples include [27, 30, 31]. An extending survey on this family of protocols can be found in [29].

HISPs assure human users that there is no attack that allows an intruder to get a system into an insecure state (where the connections established are other than what the humans believe) with probability greater than 2^{-b} where b is the size (in bits) of the SAS transmitted over the empirical channel [29]. They provide a convenient way to bootstrap security in a way that can be used in a wide variety of contexts such as where devices involved are co-located and where they are not, and where authentication is provided to all devices or asymmetrically to one. Similarly, they can be used in convenient consumer devices or as part of a security process in a more elaborate type of system. In our design, we apply HISP to mobile payment design and hence term our proposal as Human Assured Mobile Payment (HAMP).

Among proposed HISPs, our design is based on the Symmetrised Hash Commitment Before Knowledge (SHCBK) protocol [27, 28]. Our choice is based on the need to optimise human interaction with respect to the level of security achieved, and to reduce computation cost [29] which enables us to carry out efficient implementations of mobile payment on low-power processors.

To make SHCBK work in our proposal, we make a number of modifications. In the modified version of the protocol, we define two parties C (the customer or the payer) and M (the merchant or the payee). To run this protocol between C and M, each party produces a random key (hk_C or hk_M) whose size is equal to typical cryptographic hash function, e.g. 160 bits; C generates a session key k

(symmetric key) while M needs to provide a fresh public key pk_M (which does not need to be certified because we can authenticate it via the protocol). For a payment transaction, each party needs to provide necessary information such as name, the date, the nature of payment, the amount of payment, and other information. We define such information as ID (ID_C and ID_M) and other information they want to authenticate as $INFO_C$, $INFO_M$. The modified SHCBK protocol [1] is set out below:

1. $C \longrightarrow_N M: ID_C, INFO_C, hash(hk_C, ID_C), hash(k)$
2. $M \longrightarrow_N C: ID_M, INFO_M, pk_M, hash(hk_M, ID_M)$
3. $C \longrightarrow_N M: \{k\}pk_M, hk_C$
4. $M \longrightarrow_N C: hk_M$

5a $M \longrightarrow_E C: digest(hk_C \oplus hk_M, (ID_C, ID_M, pk_M, k, INFO_C, INFO_M))$
5b C compares the *digest value*[4] with its own version.

By using *hash* (a cryptographic hash function) in Messages 1 and 2, C and M are committed to the values of hk_C and hk_M even though neither party knows both of them. Messages 3 and 4 require C and M to disclose hk_C and hk_M over the normal channel ($\longrightarrow_N$) which allow devices to check the integrity of Messages 1 and 2. In addition, C and M compute a *digest value* which is transmitted from M to C via a non-spoofable empirical channel ($\longrightarrow_E$). C is convinced that a secure connection between herself and M has been established only if the value of *digest*[5] received via $\longrightarrow_E$ matches the one generated by itself. More information and the security analysis of the above protocol can be found in [27, 28].

3 Market Survey and Related Research

3.1 Market Survey

On 1st March, 2010, we made an examination of the Apple iPhone, Blackberry, Nokia, and Android stores, revealed that there were a total of 19 banking applications (provided by 18 banks), 3 third-party banking applications, 1 e-wallet application (by Paypal), and one peer-to-peer application called Starbucks Card Mobile by Starbucks.

Current MPSs are based on SMS, USSD, WAP, or IVR (Interactive Voice Response) [9]. Other than those based on WAP, MPSs use GSM/3G Cellular Network as a trusted communication channel to send data in clear text. For example, Nokia Money, M-PESA, and Smart Money [4] are all based on GSM/3G networks. As pointed out earlier, GSM/3G Cellular Networks are vulnerable to message intercepting.

[4] The *digest value* represents the SAS that is manually compared by humans.

[5] A digest [27] is a cryptographic function related to a *universal hash function*. It has two arguments, namely a key and data. It should be designed so that *inter alia* the likelihood (as the key k varies) that $digest(k, A) = digest(k, B)$ is minimised for all $A \neq B$.

Two-factor authentication mechanisms are also commonly used, employing what the user has (include IMEI (International Mobile Equipment Identity) number, IMSI (International Mobile Subscriber Identity) number, and phone number) and what the user knows, often a short password or PIN. This approach can be defeated by GSM/3G Cellular Network intercepting, or phone or SIM card cloning [14]. For example, E-Stealth Mobile Phone Spy [13] is downloadable software which requires no installation on target mobile phone and has over 2 million users—alerting us about the importance of protecting mobile phones against those attacks.

3.2 Related Research

There are a small number of papers discussing concrete empirical designs of mobile payment systems, and SA2pMP [15], mFerio [16], MFAMP[6] [17], MP-Auth [18], GOVPKI[7] [19], MobiCash [20], Mobile-to-Mobile Payment System (MMPS) [21], and P2P-Paid [22] are discussed here (see Table 1: A comparative view on mobile payment designs).

Name	Language	Connection	Main Security Mechanism
SA2pMP	J2ME	HTTP/HTTPS	ECDSA & PKI
mFerio	Not specified	NFC	Fingerprint & NFC
MFAMP	J2ME	SMS	OTP
MP-Auth	J2ME	wire line/Bluetooth	PKI & Password
GOVPKI	J2ME	Bluetooth	PKI
MobiCash	Not specified	Not Specified	PKI
MMPS	Not specified	SMS	PKI
P2P-Paid	J2ME	Bluetooth	Payment authority

Table 1: A comparative view on mobile payment designs.

We notice that the security of SA2pMP, MP-Auth, PKI, MobiCash, and MMPS is based on PKI. PKI is problematic for mobile payment in the sense that knowing that you are paying a large retailer does not tell you that you are paying your own bill.

MP-Auth [18] presents the idea of using a mobile phone as a trusted device to separate sensitive data, like credit card details, from a PC. However, the multi-layer security among the phone, the user's PC and the bank server (or a merchant server) is complicated which decreases the system's efficiency. If we use HISP, then we could consider the route from the phone to the bank server

[6] [17] describes a multi factor authentication mobile payment solution, and we term it as MFAMP.

[7] [19] describes a solution based on a governmental PKI infrastructure, and we term it as GOVPKI.

as a single insecure connection by adding a second channel which stands directly between the user and the bank. In this way, we could simplify the multi-layer system effectively and securely. We will discuss more about this in the rest of this paper.

3.3 The missing link

The results in Section 3.1 and 3.2 correspond to the three main problems we have pointed out in Section 1. We also reveal another fact: except for those based on NFC, current solutions do not rely on any direct electronic communication between a payer and a payee, therefore there is weak or no authentication between payer and payee. Most MPS rely on authentication provided by third parties, for example, banks, Mobile Network Operators (MNO), and other MPS providers. They all, therefore, require online support to complete the authentication (see Figure 1). This fact is a result of the dependence on the use of PKI, of which the authentication is completed by a third party. And other payments that do not use PKI need the service provider for authentication.

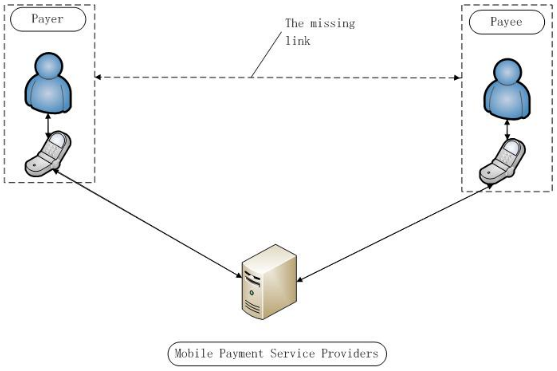

Fig. 1: The missing link

The authentication by a third party only verifies names, and, given that the mobile payment often happens in an *ad hoc* style, users have difficulty to link the authenticated names to the images of desired entities in their minds. For example, Bob wants to pay a person called John, but a man-in-the-middle attacker uses another account with the same name John to “cheat” the system; the system authenticates the name “John” and tells Bob the authentication is successful, and in the mean time, the attack is successful.

The requirement of online connection does not satisfy all the needs for making payment. For example, if we are to replace cheques, we need to consider occasion when no online connections are available, and to make payment without authentication service from the third parties. To enable users to make payment from anywhere any time, methods of bootstrapping security directly between payer and payee are necessary.

In Figure 1, the payee can also be viewed as the merchant's Point Of Sale (POS) machine or server and there could be more than one service provider between payer and payee. The arrows shows a general route of communication which could be more sophisticated in practice.

We demonstrate here that our method can establish a link directly between the payer and the payee to provide strong authentication which can be used as an additional layer of security of the existing MPS or as the only security method.

4 Requirement

4.1 Acceptance of mobile payment

We examine the main acceptance model of mobile payment. Kreyer *et al.* [23] proposed a three-factor model for evaluating mobile payment and we call it model A:

- Cost: direct transaction cost, fixed cost of usage and cost for technical infrastructure (for example, the requirement of new mobile device)
- Security: Confidentiality, Integrity, Authorization, Authentication and Non-repudiation
- Convenience: the ease of use or the benefit from using mobile payment

A more recent study made by Dahlberg *et al.* [24] suggests a five-factor model for creating successful mobile payment solution: M-payment market and providers, consumer power (or consumer demands), merchant power, traditional payment services, and new electronic payment services. To limit our discussion, we choose to discuss consumer power in this section. In [24] they have extracted 20 factors about consumer adoption from 14 papers, among which we only focus on the most important 5 factors (we call it model B): cost, compatibility, ease of use, trust and usefulness. The compatibility here means the consistency between the mobile payment service and the users' experiences, needs and habits.

Based on the above acceptance models, we divide our requirement discussion into two parts: Usability and security. SMS based MPS (SMPS) and Cash Payment (CP) are compared with HAMP. Results are shown in Table 2.

4.2 Usability

Easy to use The payment process should be efficient and effective. SMPS have high complexity and are difficult for users to manage, and sending and receiving SMS can be slow. CP is simple but error prone and expensive. HAMP provides a convenient and accurate way of inputting, generates and sends data automatically, and runs faster and cheaper than using SMS. However, it requires installation of software on mobile phones.

Easy to learn People without particular knowledge about the underlying technology should be able to learn how to use the payment system quickly. SMPS has a long list of different formats of messages and different service codes, which is impractical to make prompt payment and difficult for users to learn. HAMP takes advantage of an installed application that provides a fully automatic instruction system to guide users through all the steps during the payment. In addition, the optimized human effort design reduces the learning difficulty of necessary steps of human interaction. CP is naturally familiar to people and it satisfies this criteria.

4.3 Security

We observe that our design has brought the control of security into users' hands which allows them to determine the level of authenticity of the payment and increase the flexibility in the entire security that lies behind the payment, which distinguishes it from most MPS solutions. Therefore the security criteria we need to study here are divided into two categories: *objective security* and *subjective security*. A similar definition is given by Linck *et al.* [25], in which they indicate *objective security* is a concrete technical characteristic often refer to the five security objectives: confidentiality, authentication, integrity, authorization and non-repudiation; *subjective security* addresses users' questions about how to feel secure when using mobile payment.

A survey of [25] revealed confidentiality, encryption, stating "security" (a tautological declaration of the term "security"), transparency and traceability, authentication and authorization are the top five categories in terms of users' subjective view of security. This shows that users' need of security in mobile payment is strong. As the statements overlap with those in *objective security*, we choose to discuss visible security in *subjective security*.

We call our security design as *organic security* because it allows users to bootstrap security based on human trust rather than relying on PKI or other hidden security technology which is difficult for them to understand and manage.

Objective security

Confidentiality. The payment information must be protected from an unauthorized party. SMPS and CP apply no encryption for the data sent and received during the payment. HAMP uses a suite of cryptography functions to protect the confidentiality of the data.

Authentication. This ensures two parties can trust each other. As we have discussed in Section 3, SMPS uses two-factor authentication which is vulnerable to attacks, and it only partially satisfies this criteria. CP depends on human trust only and may not be enough in some scenarios, for example, where the payer does not know the payee. HAMP uses multi-factor authentication mechanism which can effectively authenticate the two parties.

Authorization. Only authorised users can make a requested payment. With the phone and the password/PIN, users of SMPS are authorised to make a payment, but we have shown that this is not enough; and no authorization is applied to CP (for example, the lost money can be used to make payment without authorization). HAMP's multi-factor authentication strategy and its account management – which can include password/PIN etc. – can effectively justify the authorization compared with SMPS and CP.

Integrity. Procedures must be in place to guarantee the system has not been corrupted by an attacker. We have not found any evidence that the SMS can be modified on the air, therefore SMPS satisfies this criteria; and we assume the users are aware of the amount of money and the transaction details, therefore CP satisfy this criteria as well. With effective encryption and verification mechanisms, HAMP satisfies this criteria. Indeed, no trust is necessary in the main communication media used by a HAMP.

Non-repudiation. The user must not deny the performed transaction and must provide proof in case that this situation occurs. Due to the weak authentication and authorization of SMPS and CP, they both fail to meet this criteria.

Subjective security

Visible security. In security design, one approach is to make security "transparent" by hiding security elements from users and freeing them from the need to understand underlying security principles. Dourish *et al.* [32] point out that this approach has a drawback in that when a computer system has a problem or an unexpected error happens, it does not help users to cope with their security environment. In addition to the visible security elements we found in the existing security designs (for example, the password or PIN), we introduce digest comparison as a comprehensive way of endowing users with a strong capability of formulating and controlling of security. We observe that during the digest comparison, several factors are affecting users' visibility of security. First is the creation of an empirical channel that users trust and are familiar with. The process of creation can be completely arbitrary and depends only on users' perspective of how trust and security should be conducted. This, however, does not undermine the actual security to be established. Second, users' action of inputting a SAS is manually conducted in order to force them to determine security critically and correctly rather than simply pressing an "OK" button that would be vulnerable to complacency. A similar approach has been used in a train signalling system[8]. SMPS requires the user to input the password and CP employs technologies against counterfeiting notes, therefore they both satisfy this criteria.

5 Proposed architecture

5.1 Overview

Figure 2 shows our two-layer design. The dotted lines represent communications possibly necessary during payment depending on payment method used. Layer

[8] See: `http://www.park-signalling.co.uk/verb.htm`

Criteria	Satisfied (SMPS)	Satisfied (HAMP)	Satisfied (CP)
Easy to use	n	y^-	y
Easy to learn	n	y	y
Confidentiality	n	y	y
Authentication	y^-	y	y^-
Authorization	y^-	y	n
Integrity	y	y	y
Non-repudiation	y^-	y	n
Visible security	y	y^+	y

Table 2: SMPS, HAMP, CP: satisfiability of requirements.

A represents an infrastructure which supports a variety of payment such as (i) e-cash, (ii) credit card authentication and (iii) e-banking. Layer A depicts the security created by the HISP. Layer B provides additional functionality (such as e-banking) and possibly security. We assume that all communications to and within Layer B are secure. The existence of secure and authenticated communication means that the human payer does not have to type in much information: perhaps just the authentication string, PIN and a confirmation key. Depending on application, you might consider either Layer A or Layer B sufficient for Micro-payments, or with both to secure larger payments.

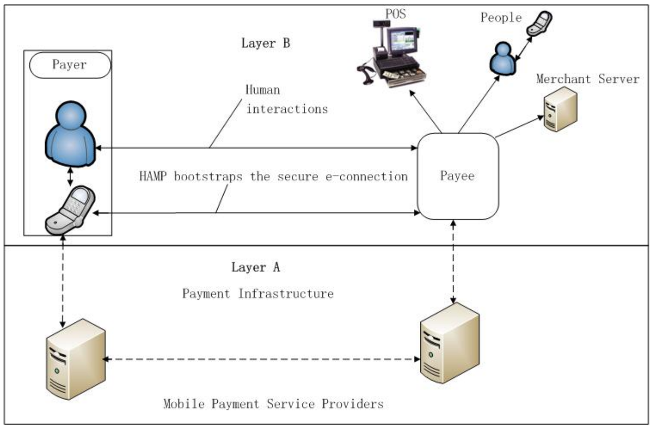

Fig. 2: Two-layer design.

The following sections introduce the security mechanisms we use in Layer B, which provides several options according to different requirements and scenarios.

5.2 Lightweight Cryptography Scheme

Prior to our research on Mobile Payment, we created a prototype of a HISP-based secure credit card payment scheme using a secure handheld device to read details from the credit card[9]. One design criterion was to implement it on a low cost processor, perhaps one with only 2Kb of memory and 64Kb of code space. We have adopted the same suite of cryptography functions on the mobile platform as on the low cost processor by taking into consideration that many mobile phones being used today are not necessarily equipped with advanced chip sets with high computing power and large space of memory and code storage.

We used 1024-bit RSA ($65537(2^{16} + 1)$ as the public key exponent), SHA-1 and AES-128 as the cryptography suite from FIPS 186, sp800-78-2, sp800-57. It would not add significant cost to strengthen the hash or symmetric key algorithm. Note that, as in the protocol set out earlier, we only use one-time uncertified asymmetric keys.

5.3 Multi-factor authentication strategy

Multi-factor authentication can be used in two different facets of HAMP Payment, both in establishing that the payer is authorised to pay, and in establishing that payment is being made to the correct party.

The primary means of proving that payment is to the correct party is using the HISP. However the fact that the payer uploads the details of the payee (including perhaps a logo) and only then confirms the payment provides a powerful secondary mechanism since these details will be included in any payment instruction issued to the banking system.

The existence of this secondary mechanism justifies the use of a short authentication string of 4–6 digits. Even if an attacker guesses correctly he is unlikely to get someone to agree to a fraudulent payment.

During a payment we use the HISP and secondary security features to ensure that payment goes to the correct party. In most applications it is also necessary to prove that the payer is authorised to access the given account. Note that the binding of phone to service provider (e.g. bank) is long term and can be supported by more traditional cryptography. The means of authenticating the payer might include:

What I have: the mobile phone, which includes:

- IMEI number: it is unique to each mobile phone allowing the user to be identified by his/her device.
- IMSI number: it is a unique number associated with all GSM and Universal Telecommunication System (UMTS) network mobile phone users. It is stored in the SIM card in the mobile phone.

[9] See the video "Low-power implementation of secure payment" on `http://www.comlab.ox.ac.uk/hcbk`.

– User's name: it is used to facilitate the payment, for example, the payer and payee may need to know how to call each other; and if we need to generate e-cheque, this must be included in the e-cheque for the verification by the banks.

What I know: PIN.

What I am: biometrics that could be given or allowed by the mobile phone, this is optional because we do not require any particular support from the mobile phone itself. In our implementation, we choose to use the user's photo as the biometrics in the protocol; it could be uploaded by the user or taken by his or her mobile phone.

We believe there are two steps in the design that are critical: the Hook-up of the connection and the manual comparison of the SAS. Finding the optimum ways to improve them is essential to a successful commercial product based on our design. However, without giving the concrete scenario of the use case, it is very difficult to determine which kind of methods to use in the two steps. Therefore, we try to choose methods that with low cost and have a ubiquitous foundation among mobile phones, but it does not define how this should be done or give any constraint on how the implementation should follow.

5.4 Connection

Rapidly bootstrapping is essential to both usability and the acceptance of security. The main problem in bootstrapping the pre-secure connection is informing one party of some ID such as a phone number or a Bluetooth address code of the other. Some communication medias are suitable only at short range, e.g. Bluetooth and NFC. Others can be used at any range, e.g. telephony (making phone call or sending SMS) and the Internet (by using GPRS/3G). We define two main types of mobile payment based on locality: proximity payment and remote payment, which are implemented and discussed in Section 6.

We notice that companies and organizations are trying to find the most usable ways to quickly set up a connection between the devices. NFC uses the distance as the main identifier to connect two devices, while the majority of others require hardware support on mobile phones, there is one that is "free": Near Sound Data Transfer (NSDT) is to let the mobile phone to generate a short acoustic signals which is picked up by a small dedicated device[10]. Another innovative method is to use the time and location as the main identifiers to bootstrap the pair[11]. It is now only limited to phones with motion sensors. By bumping two phones together, each phone sends a "bumped" signal to the main server which then measures the time of the received signal together with the location details gathered from the GPS on the mobile phones. Given two phones with close locality to each other and an almost synchronised sending of the "bumped" signal, the main server then makes the decision of which two phones

[10] See www.tagattitude.fr.

[11] See www.bu.mp.

to be paired. However, the server could be confusing when many pairs of people in a small area (for example, a conference room) try to bump each other's phone.

5.5 Empirical channel schemes and mechanisms of comparing SASs

The empirical channel can take many forms. To simplify our design, we take into consideration of most commonly used strong empirical channels: "Seeing is believing". By seeing the SAS in person, for example, using a small screen to display the SAS on the merchant's till, displaying it via HTTPS web pages, or the payee shows his mobile phone screen to the payer or even writes it down on a board or piece of paper and then shows it to the payer in person. "Hearing is believing". By hearing the SAS in person, the user can be assured that this value indeed comes from the right person, and no one could fake the origin of this value. For example, the payee reads out the digest value to the payer, the customer receives a phone call from the merchant telling him or her the SAS. "Strong secure connections". For example, NFC and a point to point short wired connection.

The comparison of the SAS can be conducted in many ways. A research done by Kainda, Flechais and Roscoe [33, 34] has pointed out there are two convenient methods of conducting digest comparison. The first one is "Compare and Confirm", which is convenient but subject to security failures; word-matching and number-typing are designed and tested under this category. The second one is "Copy and Enter", which is secure but requires more human effort; repeated numeric comparison is designed and tested under this category [34]. Their research shows possible ways to minimise human effort while making no compromise on security. For simplicity of implementation and demonstration, we use number inputting in our demos where the payer manually reads and enters the SAS on his or her mobile phone.

6 Implementation

To demonstrate the features of our design, we have implemented two applications A and B. The first one demonstrates person-to-person mobile payment[12] (also demonstrates as the proximity payment), and the second one demonstrates person-to-business mobile payment[13] (also demonstrates as the distant payment). In both cases we have assumed that the payment is managed through online banking or an e-wallet accessed through the mobile phone.

Application B shows an additional feature that our design can simplify the online payment system and make use of the user's mobile phone as the authenticator to conduct safer online payment or online banking, which has been discussed earlier in Section 3.2. The scenario of B is as follows:

[12] See the video "Peer to peer payment between two mobile phones" on `http://www.comlab.ox.ac.uk/hcbk`.

[13] See the video "On-line payment" on `http://www.comlab.ox.ac.uk/hcbk`.

1. The customer C has come to the point of paying on an internet session and is confident that the HTTPS[14] session is connected to the merchant M.
2. C presses a button on the website for mobile payment and starts (*) the payment application on his mobile phone. The button gives C's phone payment number to M securely via HTTPS.
3. M calls C's mobile phone and runs the initial messages of the protocol with it.
4. M calculates the digest and displays it on existing HTTPS window.
5. Assuming C wishes to carry on; he types this number into phone which then decides if numbers agree. Agreement gives secure connection.
6. M sends details of the payment it wants over the secure (authenticated and encrypted) connection including amount, name, possible logo and bank information.
7. The payment is displayed on mobile phone (in our implementation, in the form of a cheque) and C is asked to confirm payment (*).
8. Payment is processed by e-banking, which generates a "receipt" to send to M.

It will be necessary in practice to have the customer prove his identity as part of this process. One or both of the points marked (*) are appropriate. And Application A works similarly.

"Chip and PIN", the EMV protocol, has been broken by a Cambridge research team [35] recently, and they have also pointed out that the Chip Authentication Protocol (CAP) [36], a protocol used in card reader based online banking, is flawed. Our implementation of B can be easily extended to an online banking example, which not only eliminates the man-in-the-middle attack, one attack that has not been properly addressed by the current online banking solutions, but also gives a way to a safer implementation of CAP by connecting the card reader to a PC, which allows a display of full transaction details and options of extra security bits, as indicated in [36].

The main implementation on mobile phones is based on J2ME, and the cryptography functions are implemented by using the Bouncy Castle Cryptography Library. We notice that the public key encryption requires over 10 seconds on some low end mobile phones. To avoid long waiting in the scenario of person-to-business payment where we assume the merchants have better equipments than the users, the implementors can delay the encryption of the symmetric key until the users complete the digest comparison, which can make use of the time of comparison to compute the result.

7 Conclusion and further work

Many new technologies have been invented for the purpose of making secure and convenient mobile payments; and more and more companies and organizations

[14] HTTPS can be replaced by making a phone call, sending a text message, or any other ways that the user can trust.

are working together to introduce and promote different MPSs in the market. Each of these factors shapes and changes the direction of mobile payment design, and a solution to solve all of the problems seems to be even more complicated. We have presented a completely new design which takes a different prospective on the current MPSs and related research results; by adding another layer of security, we could reduce the dependencies on infrastructures, and therefore open the way for the future of more cost-effective and ubiquitous solutions of making mobile payment.

Surveys and tests are needed in the future to determine the users' actual acceptance of our new design, for example, the users' confidence of the bootstrapped security and the manageability of the applications.

References

1. A.W. Roscoe, Chen Bangdao and L.H. Nguyen. *Reverse Authentication in Financial Transactions.* To appear in 2nd International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use. 2010.
2. J. Dolan. *Accelerating the Development of Mobile Money Ecosystems.* Proceeding of MOBILE MONEY SUMMIT 2009.
3. B. Jenkins. *Developing Mobile Money Ecosystems.* Proceeding of MOBILE MONEY SUMMIT 2008.
4. G. Ivatury and I. Mas. *The Early Experience with Branchless Banking.* CGAP. 2008
5. O. Morawczynski. and G. Miscione. *Examining trust in mobile banking transactions: The case of M-PESA in Kenya.* Pretoria: Human Choice and Computers 2008, pp.287-298
6. N. Parmelee. *Docomos' holding strong.* See: `http://www.fool.com/investing/international/2007/02/01/docomos-holding-strong.aspx`
7. See: `http://conversations.nokia.com/2010/02/15/nokia-money-pilot-begins-in-india-video/`
8. A.S. Lim. *Pre-Standardisation of Mobile Payments: negotiations within consortia.* Proceedings of the International Conference on Mobile Business. 2005
9. J. Ondrus and Y. Pigneur. *An Assessment of NFC for Future Mobile Payment Systems.* ICMB 2007
10. U. Varshney *Mobile Payments.* IEEE Computer 35(2002)(12),pp.120-121.
11. See: `http://www.global-security-solutions.com/PGFDigitalCellularIntercepter.htm`
12. See: `http://www.interceptors.com/intercept-solutions/Passive-GSM-Interceptor.html`
13. See: `http://www.e-stealth.com/ULTIMATE-BLUETOOTH-MOBILE-PHONE-SPY-2010_p_588-8.html`
14. P. Bardon, S. Field, N. Davey, G. McAskie, R. Frank. *The detection of Fraud in Mobile Phone Networks.* Neural Netw. World 6, 4. 1996.
15. Y. Zhu and J. E. Rice. *A Lightweight Architecture for Secure Two-Party Mobile Payment.* International Conference on Computational Science and Engineering, 2009.
16. R.K. Balan, et al. *mFerio: The Design and Evaluation of a Peer-to-Peer Mobile Payment System.* Proceedings of the 7th international conference on Mobile systems, applications, and services, 2009.
17. F. Aloul, S. Zahidi, W. El-Hajj. *Multi Factor Authentication Using Mobile Phones.* International Journal of Mathematics and Computer Science, 4(2009), no. 2, pp. 65-80

18. M. Mannan and P.C. van Oorschot. *Using a Personal Device to Strengthen Password Authentication from an Untrusted Computer.* Financial Cryptography and Data Security, pp. 88-103, 2008, Springer.
19. M. Hassinen, K. Hypponen, and K. Haataja. *An Open, PKI-Based Mobile Payment System.* Emerging Trends in Information and Communication Security, pp. 86-100, 2006, Springer
20. S. Bakhtiari, et al. *MobiCash: A New Anonymous Mobile Payment System Implemented by Elliptic Curve Cryptography.* WRI World Congress on Computer Science and Information Engineering, 2009.
21. A. Saxena, M.L. Das, A. Gupta. *MMPS: a versatile mobile-to-mobile payment system.* Proceedings of the International Conference on Mobile Business, 2005.
22. J. Gao, K. Edunuru, J. Cai, S. Shim. *P2p-paid: a peer-to-peer wireless payment system.* Proceedings of the 2005 Second IEEE International Workshop on Mobile Commerce and Services.
23. N. Kreyer, K. Pousttchi, K. Turowski. *Standardized Payment Procedures as Key Enabling Factor for Mobile Commerce.* Third International Conference, ECWeb 2002. Aix-en-Provence 2002, pp. 400-409.
24. T. Dahlberg et al. *Past, present and future of mobile payments research: A Literature Review.* Electronic Commerce Research and Applications, 2007.
25. K. Linck, K. Pousttchi, D.G. Wiedemann. *Security Issues in Mobile Payment from the Customer Viewpoint.* Proceedings of the 14th European Conference on Information Systems (ECIS 2006)
26. D. Dolev and A. Yao. *On the security of public key protocols.* In Information Theory, IEEE Transactions on, volume 29(2), pp. 198-208, 1983.
27. L.H. Nguyen and A.W. Roscoe. *Efficient group authentication protocol based on human interaction.* In Proceedings of FCS-ARSPA 2006, 9-31.
28. L.H. Nguyen and A.W. Roscoe. *Authenticating ad hoc networks by comparison of short digests.* Information and Computation 206 (2008), 250-271.
29. L.H. Nguyen, A.W. Roscoe. *Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey.* Computer Security, 2010.
30. S. Vaudenay. *Secure Communications over Insecure Channels based on Short Authenticated Strings.* Advances in Cryptology - Crypto 2005, LNCS vol. 3621, pp. 309-326.
31. S. Laur and K. Nyberg. *Efficient Mutual Data Authentication Using Manually Authenticated Strings.* Volume 4301 on LNSC, 90-107, 2006.
32. P. Dourish, R.E. Grinter, J. Delgado de la Flor, M. Joseph. *Security in the wild: user strategies for managing security as an everyday, practical problem.* Personal and Ubiquitous Computing, 8 (6) (2004) 391-401.
33. R. Kainda, I. Flechais and A.W. Roscoe. *Usability and Security of Out-Of-Band Channels in Secure Device Pairing Protocols.* In the Proceedings of SOUPS 2009.
34. R. Kainda, I. Flechais and A.W. Roscoe *Secure and usable out-of-band channels for ad hoc mobile device interactions.* 2010.
35. S.J. Murdoch, S. Drimer, R. Anderson, M. Bond. *Chip and PIN is Broken.* 2010 IEEE Symposium on Security and Privacy.
36. S. Drimer, S.J. Murdoch and R. Anderson. Optimised to Fail: Card Readers for Online Banking. Financial Cryptography and Data Security 2009.