Security Issues in Mobile Payment Systems

Shivani Agarwal[1]*, Mitesh Khapra[1], Bernard Menezes[1] and Nirav Uchat[1]

ABSTRACT

The national exchequer, the banking industry and regular citizens all incur a high overhead in using physical cash. Electronic cash and cell phone-based payment in particular is a viable alternative to physical cash since it incurs much lower overheads and offers more convenience. Because security is of paramount importance in financial transactions, it is imperative that attack vectors in this application be identified and analyzed. In this paper, we investigate vulnerabilities in several dimensions – in choice of hardware/software platform, in technology and in cell phone operating system. We examine how existing and future mobile worms can severely compromise the security of transacting payments through a cell phone.

Keywords: M-payment, Mobile worms and viruses, Spyware

1. Introduction

There are several reasons why governments and financial institutions should advocate the use of electronic payments in financial transactions. E-payment systems offer huge cost savings to the government because use of electronic cash is much cheaper than printing paper currency. By obviating the need to transport, handle, store and dispense physical cash, electronic cash offers enormous savings to banks and merchants. Finally, electronic cash offers unprecedented convenience to the customer who does not need to carry currency notes and coins. The widely used ATM (Automated Teller Machine) was one of the early successful experiments aimed at saving costs to the bank and at the same time providing 24 X 7 cash service to the customer. However, ATMs deal with paper currency. Furthermore, the cost of replenishing cash on ATMs and maintaining them is also high. Electronic cash and electronic payment schemes are an attractive alternative from the perspective of cost and convenience. It is expected that in the space of electronic payment systems, mobile payment schemes – those in which at least one part of the transaction is carried out using a mobile device - will soon dominate the world of electronic payments. This is, at least in part, due to the easy availability of mobile phones.

An m-payment system typically involves five main actors. These include a Financial Service Provider (FSP), a Payment Service Provider (PSP), a Mobile Network Operator (MNO), a payer and payee. An FSP is usually a bank and is responsible for performing the backend processing required for settling a transaction between two parties. A PSP facilitates the communication between the FSP and the payer/payee by providing the payment software and user interfaces. The MNO provides the infrastructure necessary for wireless WAN service. In addition, there are regulators who are involved in monitoring compliance with the rules and laws related to m-payments. These are generally government bodies or law enforcement

[1] Department of Computer Science and Engineering, IIT Bombay, India
* *Corresponding Author:* (Email: shivania@it.iitb.ac.in, Telephone: 91-9819377530)

agencies. Security is of paramount importance in an e-payment system. As a first step in designing a cell phone-based e-payment system, it is important to analyze the various security issues that may arise from the choice of platform and of technologies. Therefore, in this paper we look at various vulnerabilities in using a cell phone as a vehicle for e-payment. These range from vulnerabilities in GSM, Bluetooth, SMS, and J2ME to mobile worms and viruses. The remaining part of this paper is organized as follows: In section 2 we provide an overview of the current m-payment systems and technologies. In section 3 we present a taxonomy of the various vulnerabilities that can impact the security of an m-payment system. In section 4 we discuss the impact of recent mobile malware and spyware on the security of m-payment systems. We present our concluding remarks in section 5.

2. M-Payment System Overview

In this section we outline the steps involved in m-payment systems and then explore technologies used to implement such systems.

2.1 Interactions between the Entities involved

M-payment systems are of two types - Remote Payments Systems and Proximity Payment Systems. In the former, the payer and the payee are at remote locations, e.g. a customer places an order from his home to a retail store. In the latter, payer and payee are in the same vicinity, e.g. a customer (payer) buys a cup of coffee from a vending machine (payee). As shown in Figure 1, the following steps are typically involved in carrying out a transaction using a Remote m-payment system:

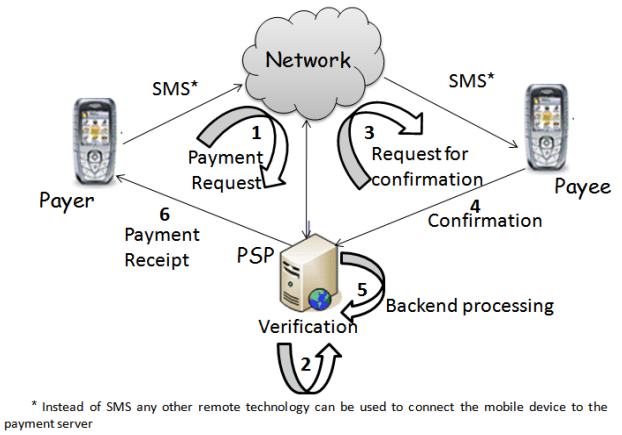

Figure 1: Basic Architecture of a Remote M-Payment System

- The customer uses his mobile device to send a payment request to a PSP over a wireless network. This request includes the details of the payee and amount to be paid.
- A. PSP verifies the credentials of the customer and the payee (basically it checks whether the customer and payee have registered for such an m-payment service).
- Optionally, the PSP might ask the customer for some more details (like a password) for authentication.
- Once the credentials of the customer have been established, the PSP requests the payee for confirmation by forwarding the payment details.
- The payee then sends a confirmation message to the PSP.
- After successful confirmation, the PSP performs backend processing to update the accounts of the payer and the payee.
- It sends a payment receipt to the payer. It might also optionally send a "Transaction completed" message to the payee.

The transaction processing in proximity m-payment systems (Figure 2) is similar to the process followed in remote m-payment systems. The main difference lies in steps 1 and step 3. In remote m-payments, the customer first sends the payment request to the PSP over a wireless network by using a remote wireless technology. The PSP then forwards this request to the payee. However, in proximity m-payments, the customer directly sends the payment request to the payee typically using a short-range wireless technology. The payee then forwards this payment request to the PSP over a wireless network. Figure 2 summarizes the steps in proximity m-payments.

Apart from this classification m-payments can also be categorized based on the payment value involved (Micro, and Macro payments) and the charging method used (Post-paid, Pre-paid, and Pay-now).

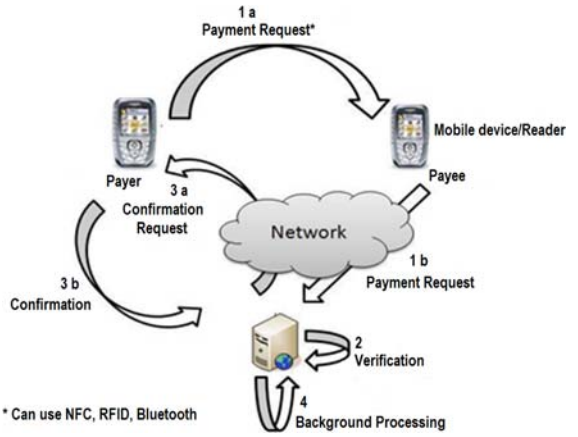

Figure 2: Basic Architecture of a Proximity M-Payment System

2.2 Technologies and Standards for m-payment systems

In order to perform a security analysis of an e-payment scheme it is necessary to understand the underlying standards, technologies, protocols and platforms used. The two popular standards used for mobile communication are Global System for Mobile communications (GSM) and Code Division Multiple Access (CDMA). GSM based phones use a SIM (Subscriber Identification Module) card which is a detachable smart card containing the user's subscription key used to identify a user. In CDMA based phones, the phone itself stores the subscription key. A common technology for remote payment systems is SMS (Short Messaging Service) which is a low cost alternative to making calls. SMS is an attractive technology because of its ease of use and low cost. SMS based payment systems are of two types, namely, the ones which do not require a change in the device infrastructure (SIM card) and the ones which do. In the former case, the user can initiate or authorize a transaction by sending an SMS message using a standard SIM card. PayPal (14) and SMS-Credit (Fong & Lai, 2005) are examples of such SMS based systems.

In the second case, a special purpose SIM card is used which is configured and programmed using the SIM Application Toolkit (SAT). SAT is a GSM standard and is used for programming the SIM so that it is able to initiate various actions. In third generation GSM phones, the device contains a USIM (Universal SIM) in place of a SIM. USAT (Universal SAT) is used for programming the USIM. The mobile operator is responsible for providing the SIM/USIM card and the application logic as in (Hassinen & Hypponen**,** 2005). Their scheme leverages the Public Key Infrastructure (PKI) provided by the Finnish government. Technologies such as Bluetooth, Infrared (IR), Near Field Communication (NFC) and Radio Frequency ID (RFID) are used for proximity payment systems. IR is a directional technology meaning that the two communicating devices need to be in each other's "line of sight". Bluetooth and RFID-based systems do

not have this disadvantage. RFID-based smart cards have attracted the interest of several m-payment associations because of their small connection establishment time (approximately 2 seconds), non-line-of-sight requirement and ease of use. Octopus (15) is an example of a popular RFID-based payment system.

In NFC based payment system, the mobile device carries two smart chips, the normal SIM card and a separate NFC payment chip. The user can initiate a transaction by holding the handset in front of a NFC reader and entering a security PIN (using the handset) to authorize the transaction. Like RFIDs, NFC has a very short connection establishment time (approximately 1 second) which makes it easier to use as compared to Bluetooth or IR. The mobile payment area is rapidly changing because the infrastructure itself is rapidly evolving. Almost all existing approaches focus on 2G or 2.5G infrastructures in order to achieve the critical mass. The début of UMTS (Universal Mobile Telecommunication System), wireless LAN, WiMAX (Worldwide Interoperability for Microwave Access) and newer technologies (3G and beyond) will provide new capabilities that will allow more sophisticated approaches to be developed. In addition to communication technology, one must also select a platform for implementing the e-payment scheme. J2ME (Java 2 Micro Edition) is the most widely used platform for developing cell phone applications because of its easy availability and ready portability. It provides a rich set of APIs/features to develop client side MIDlets for sending SMS messages, communication via Bluetooth, establishment of connection to a payment gateway using HTTP, support for encryption/authentication, etc. It thus aids in the development of client side applications for almost any of the m-payment schemes describe above.

Apart from this, J2ME has an optional package, namely, SATSA (Security and Trust Services API) (16) which is used to interact with the special purpose SIM cards mentioned above. To ensure the security of the system the user has to authorize the use of these special purpose APIs by entering a PIN. Note that this PIN is different from the PIN used to lock a mobile device (which most users are either unaware of or choose to avoid using it). This PIN is a SATSA specific PIN which has to be entered by the user to authorize any cryptographic operation (A user cannot choose to ignore this PIN). New mobile phone models provide advanced security capabilities such as digital signatures, encryption, and biometric authentication. Future mobile payment systems will enhance security by taking into account what 3G infrastructures offer and what the latest mobile phone generation supports.

3. Investigation of vulnerabilities

M-payment system rides on some underlying infrastructure (say GSM) or employs a technology (say Bluetooth or RFID). The security vulnerabilities in such underlying technologies are often ignored while analyzing the security aspects of an m-payment system. An accurate security analysis is possible only if we take a holistic view of the vulnerabilities at each dimension instead of considering only a specific dimension (say protocol or platform) of the m-payment system.

3.1 Layers of support for m-payment systems

With the above discussion in mind we first provide a bigger picture of the several layers that constitute an m-payment system (as shown in Figure 3).

At the highest level we have the m-payment protocol which rides on top of a software development platform and a wireless infrastructure. These underlying components in turn have a layered architecture comprising APIs, Operating Systems, hardware etc. The KVM (K Virtual Machine) is a compact, portable Java virtual machine intended for small, resource-constrained devices such as cellular phones. To ensure the security of the m-payment system as a whole it is necessary that every layer in the m-payment system is robust to attacks like man-in-the-middle attack, replay attacks or impersonation attacks. Security features like authentication, authorization, confidentiality and non-repudiation are an absolute necessity for any m-payment application. The designers of the m-payment application may choose to derive some of these features from the underlying layers. However, while doing so, one must not assume that the underlying

technology is completely secure because it might have some design flaws (for example GSM does not have a provision for mutual authentication) which make way for attacks like man-in-the-middle-attack and replay attacks.

`

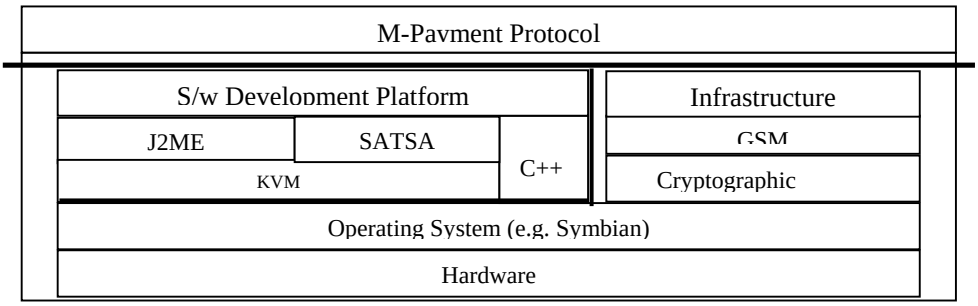

Figure 3: Different layers of an m-payment system

Like other applications on the cell phone, the m-payment application runs on an Operating System (e.g. Symbian) in which case it is important to analyze the vulnerabilities in the Operating system itself (for example, examine the robustness of the OS to worms and viruses). Lastly, we need to analyze the hardware on which the m-payment system runs. SIM card is an example of a hardware component which, if tampered or cloned, might affect the security of an m-payment system.

3.2 Taxonomy of vulnerabilities

Based on the above discussion, we now create a taxonomy of some of vulnerabilities at different layers and their effects. We briefly describe these vulnerabilities and examine how existing or proposed m-payment systems could be affected by them.

The prominent development platforms are J2ME and Symbian C++. Both are known to have vulnerable APIs (which give unauthorized access to restricted functionalities). Further, some of the cryptographic APIs provided by these platforms may use weak cryptographic keys or predictable random numbers which makes them vulnerable to cryptanalysis and dictionary attacks. On the other hand, vulnerabilities in the mobile infrastructure are generally the result of weak cryptographic algorithms or design flaws (as mentioned earlier, GSM does not have a provision for mutual authentication) which make way for attacks such as impersonation or man-in-the-middle attacks. Fixing these vulnerabilities is difficult as it would require a change in the protocol design or in the cryptographic algorithms in the protocol. The most severe vulnerabilities are those related to the hardware itself. For example, as explained earlier, SIM cards play a very important role in an m-payment system as they uniquely identify a subscriber. If an attacker is able to clone a SIM card then he can carry out malicious transactions on behalf of the user. It has already been demonstrated (Rao, Rohatgi, Scherzer & Tinguely, 2002) that it is possible to extract the 128-bit COMP128 keys used to uniquely identify a user using a side channel attack (which involves monitoring side channels such as power consumption and electromagnetic emanations). It is the responsibility of the SIM card vendors to take steps to make the system robust against such side channel attacks. A recent development which has implications to the security of m-payments systems is the evolution of mobile malware. These malware mainly exploit some vulnerabilities in the OS or misuse the various features (say APIs) provided by the OS. To the best of our knowledge, this paper is the first attempt at analyzing the impact of mobile malware on the security of m-payment systems.

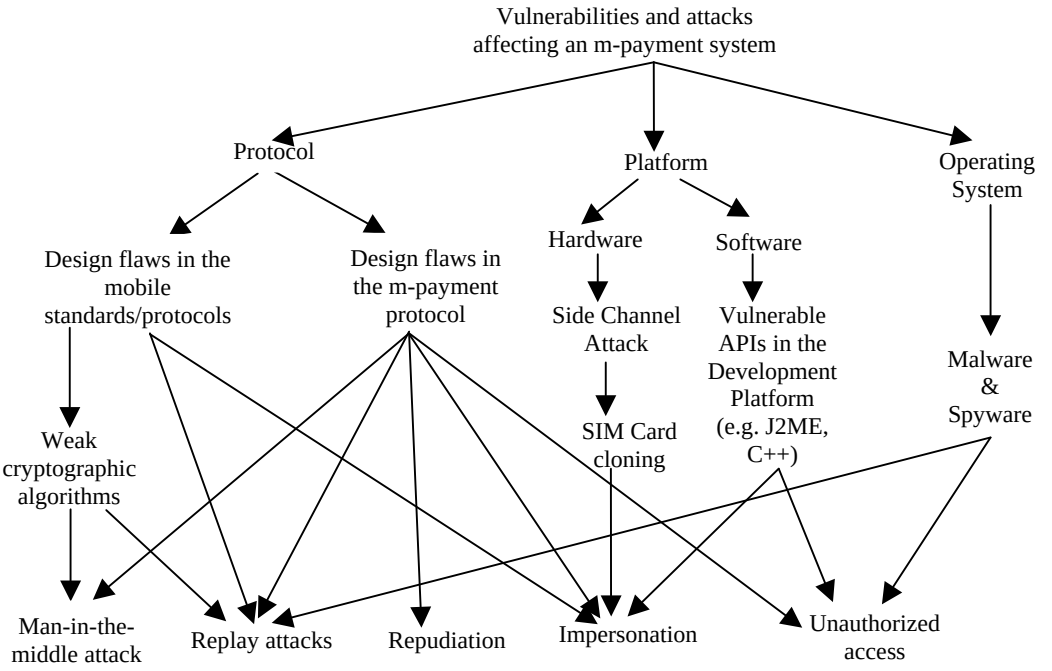

Figure 4: Taxonomy of vulnerabilities affecting an m-payment system

3.3 Vulnerabilities in GSM

One option in m-payment system design is to leverage the existing mobile network infrastructure since it already has authentication and encryption mechanisms in place. Before doing so, a security analysis of the currently used mobile standards is necessary. 2G GSM uses A5/1 and A5/2 stream ciphers for encrypting data so that over-the-air communication privacy can be ensured. The design of both these algorithms was kept secret. However, it became public knowledge through reverse engineering and it was demonstrated (Barkan, Biham & Keller, 2006) that A5/1 and A5/2 do not provide an adequate level of security. However, 3G mobile communications uses a block cipher, A5/3 which is much stronger as compared to A5/1 and A5/2. Some attacks on 2G GSM protocols were published in (Barkan, Biham & Keller, 2006). These attacks include a cipher text only attack on A5/2 which requires few milliseconds of encrypted on the air conversation for finding the correct key in less than a second on a computer. On networks using A5/1 and A5/3, these attacks are more complex. For these networks a man-in-the-middle attack can be launched whereby an attacker impersonates a base station to the user. These attacks are possible because the keys are same for A5/1 and A5/2. Although, A5/3 can be used with key lengths of 64–128 bits, the current GSM standard uses 64-bit A5/3 which makes this algorithm vulnerable to all such attacks.

Most of the existing m-payment schemes rely on the GSM network's security in one way or another. It is difficult to ensure secure m-payments if the underlying network is not secure, no matter what security mechanisms the proposed m-payment scheme uses. For example, in SMS-Credit (Fong & Lai, 2005), the user's PIN is sent in the SMS message. An attacker can hack the GSM encryption key using cipher-text only attacks; decrypt the message and then get the PIN. The attacker can also alter the message by call hijacking.

3.4 Vulnerabilities due to the choice of platform (Case study: J2ME)

No matter how safe and secure an m-payment scheme is, it could still be insecure due to the security vulnerabilities in the platform chosen for implementing the scheme. As mentioned in the previous section, J2ME is one of the preferred platforms. Several researchers have developed prototypes for their m-payment systems using J2ME. Debbai, Saleh, Talhi & Zhioua (2005) documents a vulnerability in some Java-enabled phones that can be exploited to write a malicious MIDlet that sends SMS messages without requiring the user's authorization. This could affect the security of some SMS based schemes which require the user to send a SMS message (to the payment gateway) to initiate a transaction. If a malicious MIDlet is installed on the user's phone which sends SMS messages then it *would be possible to initiate a transaction without the approval of the user*.

Another vulnerability in Sun's Reference Implementation allowed one MIDlet to access the data stored by another MIDlet using some lower level APIs. *This makes all sensitive data (like encryption keys) stored by an m-payment MIDlet unsafe from another malicious MIDlet.* Hence it is important to ensure that the selected platform does not allow unauthorized access to the data stored by one application to another application. Random numbers are used in several cryptographic algorithms and are also used to compute the encryption key when using HTTPS (SSL) for secure communication. It was found that Sun's Reference Implementation uses simple logic for generating random numbers (based on the system timestamp) *which can be guessed by an alert hacker* using sniffing tools. This could affect the security of m-payment systems and hence it is important to ensure that the chosen platform is capable of generating truly unpredictable random numbers. Many of these vulnerabilities cease to exist in the later versions of J2ME, we still discussed them briefly to make the reader aware of the security precautions that should be taken while selecting a platform for implementing an m-payment scheme.

3. 5 Vulnerabilities due to the choice of technology (Case study: Bluetooth)

The vulnerabilities in the technologies employed such as Bluetooth and RFID can severely compromise the security of a proximity payment scheme. Several attacks have been proposed on the Bluetooth protocol in the past. Most of these attacks are either theoretical in nature (D. Kugler, 2003) or are possible due to a bug in the implementation (and not because of a bug in the protocol itself) (Carettoni, Merloni & Zanero, 2007) (17) (18). For example, Bluetooth air sniffers make it possible to sniff the raw data being exchanged between two devices. Access to this data could open several possibilities for an attacker such as cracking the PIN. A man-in-the-middle attack can be launched (17) by cracking the link key. Such attacks could have a serious impact on the security of m-payment schemes. In practice, it would be a bit farfetched to expect a hacker to purchase expensive Bluetooth air sniffers but then again a hacker might find it worth making an investment considering the monetary gains that he could make by breaking into an m-payment scheme.

4. Security issues due to mobile malware and spyware

Over the past few years, computer scientists have been witnessing the evolution of a vicious species a.k.a. Mobile malware!! The evolution of mobile viruses started with proof-of-concept worms like Cabir (25). The intention of the authors of Cabir was not to cause any financial damage to the victims but to make the world aware that it was possible to write viruses that could infect mobile devices. However, this idea was further explored by some virus writers who succeeded in writing viruses which caused the phone to malfunction (e.g. Skulls (26)) or caused financial damages to the user by sending SMS messages (e.g. Viver (21)) from his/her phone. Currently, every week about ten mobile phone Trojans are added to antivirus databases. Going by the current trend it is clear that this threat will only increase in the future. In light of these facts, it has become very crucial for the m-payment research community to analyze the impact of such malware on the security of m-payment systems. In this section we discuss the security implications of such malware on some m-payment schemes.

4.1. Recent mobile malware and spyware

Recent studies (19) show that the world of mobile malware is dominated by Trojans and not by worms or viruses. The main reason for this is that Trojans do not need any propagation vector and simply rely on the user's curiosity to download and install them. They attract the user's interest by masquerading as utility programs or popular games. The user's happily install such programs without realizing that they could actually be installing a spyware which is capable of recording their incoming and outgoing SMS messages as well as call logs for dialed and received calls. The spyware then sends this data to an account on a server owned by the spyware writer. One example of such a spyware is Flexispy (20) which has been in the wild for some time. In an SMS based m-payment system such malware could seriously impact the privacy of the user because a malicious attacker could make minor modifications to such spyware and track all the transactions carried out by a user.

Another spyware by the name of PbStealer (29) is capable of stealing all the entries in the user's phonebook. This spyware masquerades as a utility program capable of compressing the phonebook to save memory space. However, instead of compressing the phonebook, it copies all the entries to a text file and sends this file to any Bluetooth device in range. A malicious attacker could modify this spyware to steal sensitive data from the user's phone (as described in section 4.4). Very recently (May 2007) another Trojan (by the name of Viver (21)) was discovered which is capable of sending SMS messages to premium phone numbers without the user's approval and thereby cause financial damages to the user. Such Trojans could have severe impact on the security of m-payments systems like (Fong & Lai, 2005) which use SMS messages to initiate and authorize a transaction. The above facts suggest that it is possible to write malware which tampers/steals/sends SMS messages (or any other data) from the victim's device. Moreover, they also suggest that the idea of making financial gains by using such malicious programs has already been explored by malware writers. Considering the immense potential of making financial gains by attacking an m-payment system it would only be wise to expect that such malware writers could use their knowledge to attack m-payment systems. With this in mind, we next analyze the potential threat posed by malware/spyware to some m-payment systems.

4.2. Using spyware to crack the user's PIN

The idea of using a PIN to access an m-payment application has been proposed in several schemes (Gao, Edunuru, Cai & Shim, 2005),(Antovski & Gusev, 2003)(14). Apart from this, as described in section 2, some schemes like (Hassinen & Hypponen**,** 2005) which use SATSA to send encrypted SMS messages require the user to enter a PIN to authorize access to these special purpose APIs. Any person who knows this PIN can send signed SMS messages using the user's cell phone (E.g. If an office colleague, say Mr. XYZ, somehow cracks A's PIN then he might borrow A's cell phone under the pretext of making a phone call and secretly send a signed message from A's phone using the PIN. The message could be of the form "Pay Mr. XYZ Rs.1, 00,000."). Thus, if a hacker is able to find a way to crack the victim's PIN then he might be able to make financial gains at the victim's cost. This attack suggests a possible way of hacking the PIN using a notorious species of spyware known as Key Loggers.

Key loggers are an extremely notorious species of computer spyware used extensively to capture the keys pressed by a user. The captured data can then be sent to the author of the spyware using a TCP/IP connection. An intelligent hacker would then be able to accurately guess the passwords to the user's e-mail or bank accounts by doing a careful analysis of the captured data. As of now, such Key Logger programs have not been written for mobile phones. However, there are some utility programs (22)(23) for Symbian OS based phones which are capable of capturing all key events in the background without the knowledge of the user. A hacker could make small modifications to such a program and turn it into a Key Logging spyware and distribute it under the disguise of a utility program. He could then combine this with other Trojans and send this captured data as a SMS message to his own mobile device (as is done by Viver) or to a remote server (as is done by Flexispy). The hacker could then accurately obtain the user's PIN by

carefully analyzing the captured data. Now, if he gains access to the user's mobile device then he could send signed messages on behalf of the user. Such an attack could compromise the security of several PIN based m-payment schemes.

4.3 Using spyware to capture data for cryptanalysis

If a hacker succeeds in installing KeyLogger software on a mobile device (as described in section 4.1] then he can practically capture all the data keyed-in by the user. With the help of this data the attacker would be able to launch an attack using cryptanalysis as described below:

- Consider a scheme which sends an encrypted message from User A to User B. The message would be encrypted using B's public key.
- Using a Key Logger the attacker could capture the raw text entered by User A (i.e. the unencrypted data entered by User A).
- Using a spyware (like Flexispy) the attacker could capture the encrypted SMS message sent by User A to User B.
- If the attacker gets sufficient pairs of raw text and cipher text then he can guess B's private key using a "Known plain text" attack.

Although such an attack would be very slow (because it involves trying several keys before the correct key can be cracked) a desperate hacker would not mind spending time on this considering the financial gains at stake.

4.4 Miscellaneous attacks using mobile spyware

We conclude this section by introducing miscellaneous attacks which would be possible using known spyware/malware and utility programs. One such attack would be to use malware which steals data from the user's phone (like PbStealer). Such malware could be used to steal sensitive data like the user's PIN from the user's cell phone. These attacks need to be taken seriously considering the fact that there are some J2ME based schemes (Antovski & Gusev, 2003) which store sensitive data (like the user's private key) on the phone (on persistent storage). Apart from this the users might themselves store their PIN and other sensitive data in text files on the phone which could be compromised using such malware. There also exist some utility programs which record the voice data of all the calls made by the user. A hacker could make small modifications to such a utility program and convert it into a voice recording spyware. This could be combined with another Trojan (like PbStealer) to send the recorded data using Bluetooth. Such a malware could affect the security of some schemes (Gao, Edunuru, Cai & Shim, 2005) which use voice recognition for authentication as an attacker could replay the message recorded using the spyware.

Some schemes like (Fong & Lai, 2005) require the user to enter a SMS to initiate and authorize a transaction. The security of such a system can be breached with the help of some mobile malware/spyware as describe below:

- With the help of a spyware (like Flexispy) record all outgoing SMS messages of the customer. One of these messages will contain the authorization code required to authorize the transaction. (Of course, here we are assuming that the user has authorized some transaction earlier).
- With the help of a Trojan (like Viver) send an SMS from the customer's cell to the payment gateway without the customer's knowledge. This message will contain the authorization code discovered in step 1.

The above discussion suggests that the security of SMS and PIN based systems can be compromised using malware and spyware like Key Loggers. With the continuous evolution of mobile malware and spyware the possibility of such attacks would only increase in the future. Hence it is important to keep such malware in view while considering the security aspects of m-payment systems. It would be good if different

organizations and governments which implement and deploy these m-payment schemes take it upon themselves to make the users aware of these threats. There are already several companies like F-secure (27) and Kaspersky (28) labs who have realized the importance of combating mobile malware. These companies have released several antivirus programs which provide complete protection from the different types of mobile malware that we discussed above. In addition, they also provide several tools for disinfecting specific viruses (E.g. A separate disinfection tool is available for Cabir, Skulls, Flexispy, etc). It is advisable to install such software to protect one's cell phones from any malware related threats.

5. Concluding Remarks

Of all the mobile applications m-payments is the one in which security is of paramount importance because of the financial value at stake. It is not an exaggeration to state that one has to be paranoid while analyzing the security aspects of an m-payment system. A system is only as secure as the weakest link in the security chain and hence it is important to analyze every link in the chain. The links in the chain include hardware including the SIM card, the operating system, the software development platform and the APIs it provides. It also includes wireless protocols and standards like GSM and Bluetooth. In this paper, we provided a holistic view of the many security dimensions of an m-payment system. We looked at the vulnerabilities in the underlying technology and platform as well as the security threats posed by recent and future mobile worms and viruses. We hope that our study helps m-payment systems designers to better analyze the security of their m-payment systems.

References

1. L. Antovski and M.Gusev [2003]. *M-payments.* 25th International conference Information technology Interfaces ITI 2003.
2. E.Barkan, E. Biham, and N. Keller [2006]. *Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication.* Technion-Computer Science Department-Technical Report CS-2006-07.
3. L. Carettoni, C. Merloni and S. Zanero [2007]. Studying Bluetooth Malware Propagation: The BlueBag Project. *IEEE Security & Privacy.* 2007, Vol. 5, 2.
4. Debbai, M. Saleh, M. Talhi and C. Zhioua [2005]. *Security analysis of mobile Java.* In Proceedings of Sixteenth International Workshop on Database and Expert Systems Applications. pp. 231- 235.
5. S. Fong and E. Lai [2005]. *Mobile Mini-payment Scheme Using SMS-Credit.*. International Conference on Computational Science and Its Applications-ICCSA. pp. 1106-1114.
6. J. Gao, K. Edunuru, J. Cai, and S. Shim [2005]. *P2P-Paid: A Peer to Peer Wireless Payment System.* Proceedings of the 2005 Second IEEE International Workshop on Mobile Commerce and Services.
7. M. Hassinen, and K. Hypponen [2005]. *Strong mobile authentication.* 2nd International Symposium on Wireless Communication Systems. . pp. 96-100.
8. D. Kugler [2003]. Man in the Middle Attacks on Bluetooth. *In Financial Cryptography '03, Long Beach. Lecture Notes in Computer Science, Springer-Verlag.*
9. M. Ohkubo, K. Suzuki and S. Kinoshita [2003]. *Cryptographic approach to privacy-friendly tags.* In RFID Privacy Workshop.
10. J. Rao, P. Rohatgi, H. Scherzer and S. Tinguely [2002]. *Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards*. Proceedings of the 2002 IEEE Symposium on Security and Privacy.
11. M. Rieback and A. Tanenbaum [2006]. *Is your cat infected with a computer virus?* 4th Annual IEEE International conference on Pervasive Computing and Communications.
12. Y. Shaked and A.Wool [2005]. *Cracking the Bluetooth PIN.*: In Proc. 3rd USENIX/ACM Conf. Mobile Systems, Applications, and Services (MobiSys).
13. S. Weis, S. Sarma, R. Rivest and D. Engels [2003]. *Security and privacy aspects of low-cost radio frequency identification systems.* In First International Conference on Security in Pervasive Computing.
14. PayPal. (Online) www.paypal.com.
15. Octopus Cards. (Online) www.octopuscards.com.
16. Java Community Process: JSR-000177 Security and Trust Services API for J2ME. (Online) http://jcp.org/aboutJava/communityprocess/final/jsr)77/.
17. Trifinite. (Online) www.trifinite.org..
18. The Bunker. (Online) http://www.thebunker.net/resources/bluetooth.

19. Mobile Malware Evolution: An Overview, Part 1. (Online) http://www.viruslist.com/en/analysis?pubid=200119916.
20. F-Secure Malware Information Pages: Flexispy.A. (Online) http://www.f-secure.com/v-descs/flexispy_a.shtml.
21. F-Secure Malware Information Pages: Trojan:SymbOS/Viver.A. (Online) http://www.f-secure.com/v-descs/trojan_symbos_viver_a.shtml.
22. Global Key Capture. (Online) http://symbianexample.com/global_key_capture_capture_key_presses_globally#attachments.
23. Symbian Tips, Tricks & Code. (Online) http://developer.sonyericsson.com/site/global/techsupport/tipstrickscode/symbian/p_jogdial_symbian.jsp.
24. InfraRed. (Online) http://irda.org/displaycommon.cfm?an=1&subarticlenbr=28.
25. F-Secure Malware Information Pages: Cabir. (Online) http://www.f-secure.com/v-descs/cabir.shtml.
26. F-Secure Virus Descriptions : Skulls.A. (Online) http://www.f-secure.com/v-descs/skulls.shtml.
27. F-Secure website. (Online) http://www.f-secure.com/.
28. Kaspersky Lab. (Online) www.kaspersky.com.
29. F-Secure Virus Descriptions : Pbstealer.A. (Online) http://www.f-secure.com/v-descs/pbstealer_a.shtml.
30. UMTS. (Online) www.umts-forum.org.

About the Authors

Bernard Menezes is currently Professor at Department of Computer Science, IIT Bombay. He was a visiting faculty member at the University of New Mexico, Albuquerque and an assistant Professor at the University of Maryland, College Park. His research interests include Network Security, Forecasting, smart E-business and RFID applications. He has published extensively in national and international journals and conferences.

Shivani Agarwal is a MTech student at Department of Computer Science, IIT Bombay under the supervision of Prof. Bernard Menezes. Her specific areas of research interest are Network Security, Algorithms, and Mobile applications.

Mitesh Khapra is a MTech student at Department of Computer Science, IIT Bombay. His specific areas of research interest are Natural language processing and Network Security. Nirav Uchat is a MTech student at Department of Computer Science, IIT Bombay. His specific areas of research interest are Wireless Networking and Network Security.