Secure Online Banking on Untrusted Computers

Yanlin Peng, Wenji Chen, J. Morris Chang, Yong Guan
Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50011
{kitap,wenjic,morris,guan}@iastate.edu

ABSTRACT

Frauds and attacks for online banking have been increasing quickly. In this study, we design a smart card-based solution called secure online banking companion (SOBC) for safe online banking even on untrusted personal computers. Portability and cost are also highly considered in the design. We have implemented a prototype of the solution on a Java Card simulator, which shows the solution can be implemented easily using current smart card technologies.

Categories and Subject Descriptors

J.7 [**Computers In Other Systems**]: [consumer products]

General Terms

Security

Keywords

E-commerce, Online Banking, Security, Smart Card

1. INTRODUCTION

Nowadays, online banking becomes the preferred choice for many customers due to convenience. However, various frauds and attacks for online banking are increasing quickly in recent years. Financial institute's security measures is evolving relatively slow and is not strong enough in many cases. This study aims to design a secure and practical solution for online banking services with the following features: mutual authentication, privacy, transaction integrity, portability and low cost.

Two types of attacks are especially popular in the context of online banking service: phishing and malware attacks [7]. Phishing attack is a social engineering attack. Attackers misguide victims, probably via phishing emails or DNS attacking, to a phishing web site which usually have similar looks as the targeted legitimate websites. Phisher may record the consumer's credential (*credential stealing*), or intercept the session between the consumer and the web site at the real time (*session hijacking*). Malware attack compromises users' computers by covertly installing and running malicious software, e.g. key loggers and Trojans. Malware can also steal consumers' credentials or hijack consumer's ongoing sessions easily. Once a computer is compromised, the attacker may take full control of the computer. Hence, malware attacks are more difficult to resist.

There have been many studies for phishing and malware attacks. However, none of them is able to meet our research goal and provide a practice security solution in the context of online banking. Anti-phishing solutions [1, 4, 3, 10, 8], although effective for some cases, all depend on the trustworthy of the consumer's personal computer, hence are subject to malware attacks. To combat malware attacks, users are usually suggested to patch the operating system timely, install personal fire wall, and keep the anti-virus software up-to-date to avoid being compromised. However, many computers with up-to-date anti-virus software still become victims of malware, due to zero-day attack [6].

Smart card, which is designed for security, has the potential to build a low-cost and highly portable solution for online applications. Although there have some smart card-based solutions, such EMV (Europay-Mastercard-Visa) [2] for offline payment, Dynamic Passcode Authentication [9] and Chip Authentication Programme (CAP) [5] for online payment, none of them can meet all of our research goals.

In this poster, we design a smart card-based device for secure online banking. The device combines a smart card and a USB-size smart card reader. The micro-processor smart card provides secure storage for private keys and a trusted computing platform. The smart card reader contains trusted output (a small display) and trusted input (control buttons). The device is in the size of a USB drive and is highly portable. The device is also connected to a computer via a USB interface, which relieves consumers from transferring data manually and adapts to complex online banking applications. After registration, a consumer can carry the device with her and use it for secure online banking even on an untrusted computer. The sensitive data will be displayed on the device only. All operations are under the consumer's control and are conducted in a secure way.

Besides, operations are also simple and straightforward. The consumer just plugs the device into a computer and open a browser to visit the bank's web site as usual. When prompted, the consumer reviews the information on the device's display and pushes the confirm button if she agrees with the operation. The cost is quite affordable. Similar devices, but not with the same functions, on the market are priced at several teens of bucks.

2. THREAT MODEL

We assume the following attacking methods:

- Credential stealing. When a consumer is redirected to

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0244-9/10/10.

a phishing web site, the consumer's login credentials may be harvest by the phshing site. When a consumer is using a compromised computer, the consumer's login credential may be recorded by key-loggers.

- Session hijacking. A phishing site or a compromised computer may hijack sessions between the consumer and the legitimate server at real time as *a man in the middle*. The transaction data sent on this session may be modified maliciously. The attacker may also generate and inject fraud transactions into this session.

- Sensitive information recording. On a compromised personal computer, consumer's sensitive data other than login credentials, such as virtual credit card number, may be recorded and used later by attackers.

3. THE DESIGN OF SECURE ONLINE BANKING COMPANION (SOBC) DEVICE

In this section, we design a smart card-based device called SOBC (Secure Online Banking Companion) device to assist online banking on untrusted computers.

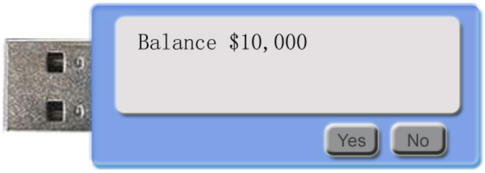

Figure 1: The proposed secure online banking companion (SOBC) device

At first, the smart card must be a micro-processor chip card with the capability to provide cryptographic function that we need. We combine the smart card and a USB-size smart card reader into one device looking like Figure 1. Smart card is internally connected to the card reader and exchange messages via ISO 7816 specified communication protocol. The card reader acts as a bridge between the personal computer and the smart card. Since the smart card and card reader are compacted to one small device, it is easy to carry and the risk to use a rogue card reader is eliminated.

The SOBC device has a USB port, a standard interface which is available on almost all computers. Connection with a personal computer makes it possible to exchange complex and long messages, which are critical for modern online banking applications. Not only long message, like account number, makes typing tedious and error-prone, but also certificate-based authentication cannot be handled by human beings.

The device has a small display and at least two control buttons, which provide trusted output and inputs. Because the personal computer is untrusted, all inputs and outputs from the computer could be distorted. However, the consumer must make decision based on *true* information and needs to give correct instructions to the SOBC device. Considering the requirements of portability and cost, we decide to put a small display and two confirm/cancel buttons (i.e. Yes/No buttons in Figure 1) on the device as trusted output and input. For input, we do not use a full-size key board or a pin pad, because they will dramatically increase the size and cost of the device.

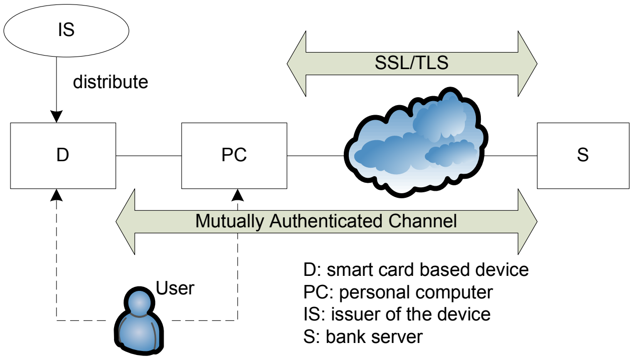

Figure 2: Architecture

4. THE DESIGN OF ARCHITECTURE

The architecture of our solutions is as Figure 2. Five entities are involved: a personal computer, a smart card-based external device, a user, a bank server, and a device issuer.

The personal computer, although untrusted, still provides interfaces for online banking. Because personal computers have become an economic and comfortable method for online banking for many people. People prefer to use the big screen and full-size keyboard connected to personal computers. Personal computer also has more power than smart card to render rich web applications. However, sensitive information should be protected from untrusted personal computers.

The smart card-based device is an external device that contains a smart card and is able to perform trusted computing as we discuss in previous Section 3. The device is designed be portable, low cost and provide a high level of assurance for secure online banking.

The consumer is the owner of the smart card-based device and knows what transactions she wants to perform. The limitation of a consumer is that she does not have an efficient communication interface with the personal computer and cannot perform complex computation. The smart card-based device assists the consumer for her limitations. The device exchanges data with the personal computer, display true information on its secure display and perform complex computation like encryption and decryption. However, the device itself does not know consumer's intentions. Hence, operations of the device must be placed under full control of the consumer.

The issuer produces smart card-based devices and distributes them to consumers. Before the device is distributed to a customer, it should be initialized correctly. The issuer generates a pair of RSA key. This operation can be done on the smart card operating system, such that the private key is not known by any party except the card itself. An X.509 certificate for the device is also generated, signed by the issuer and store on the card. The smart card will be programmed to have the capability to establish a mutual SSL connection and encrypt or decrypt messages.

The bank server provides online banking services to consumers. Our design keeps some parts of the current online banking service. A normal HTTPS session is established between the browser of the personal computer and the bank server. The consumer enters her username and password as usual. However, our design requires the application to check the existence of the SOBC device and use the certificate stored on the device as a second authentication factor. I.e., without physical possession of correct

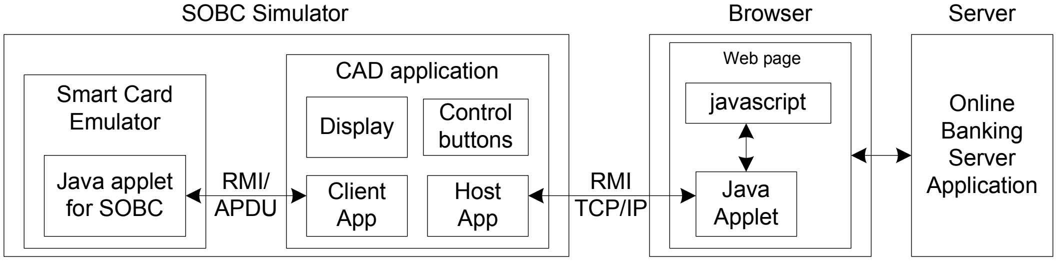

Figure 3: Prototype Implementation Diagram

SOBC device, the consumer's account cannot be accessed. This eliminates credential-stealing attacks in a large extent, since most credential-stealing attacks are online only. After the consumer has been authenticated correctly, a mutual HTTPS session is established between the device and the bank server to protect all communication between the SOBC device and the bank server. The mutual HTTPS session would eliminates session-hijacking attacks. A web page from the bank server typically contains two parts of information: non-sensitive information which is decrypted and displayed by the browser; sensitive information which is displayed as a placeholder on the browser and will be decrypted and displayed on the SOBC device after a double-clicking on the place holder. When the consumer wants to perform a critical transaction, e.g. transfer money or pay utilities, she enters the transaction contents using the keyboard. The transaction data will be reviewed by the consumer on the SOBC device. Once the consumer pushes the confirm button to agree, the transaction data will be signed with the private key on the SOBC device. The bank server will check the validity of the signature before processing the transaction. Also, for each critical operation, the consumer has to push the confirm button to agree.

5. PROTOTYPE IMPLEMENTATION

We implement a prototype of SOBC using the Java Card simulator which is distributed with the most recently 3.0.2 Java Card Development Kit. Java Card is a technology that enables smart card development with familiar Java technology. In our implementation, there're four parts like Figure 3 shows: server application, Java applet in browser, car acceptance device (CAD) application and Java Card applet.

The server side application simulates a simple online banking application with user authentication, balance reading and money transferring. We implement the server using Java servlet technology, which receive requests from the client, respond to the requests, and deal with digital signing/verification as well as symmetric/asymmetric algorithms.

The CAD application is a simulation of the cad acceptance device, which is also written in Java. The CAD has a small display screen and two controls buttons (Yes or No) for user interaction.

The on-card application is a Java Card applet based on Java Card technology. It is capable of generating random numbers, using RSA for asymmetric encryption/decryption, using AES to encrypt/decrypt messages, and signing/verifying digital signatures. Now we are using Java Card simulator which only provides RSA/AES/Signature algorithms with short keys. A real Java Card development kit would have a more practical implementation on cryptographic algorithms and support longer secure keys. The on-card application exchanges message with CAD application using Java RMI (Remote Method Invocation) based on APDU message formats which are defined by ISO 7816.

We also implement a Java applet inside the browser to help Javascript exchange request/response with the SOBC device. Javascript is usually the client-side programming language for web applications. However, Javascript is not capable of communicating with the SOBC device. We implemented such connectivity using the Java applet technology. The applet communicates with the CAD application using Java RMI (Remote Method Invocation) based on the TCP/IP protocol. The Java applet will be downloaded from the server when the consumer visits to the server's web page. It will run inside the browser and work as a bridge between the CAD application and the Javascript.

6. CONCLUSION

We design a smart card-based device to assist secure online banking. The device including following critical components: a trusted, yet resource limited, computing platform with cryptographic library (smart card) and trusted input (on-device control buttons) and output (on-device display).

7. REFERENCES

[1] N. Chou, L. Robert, T. Yuka, and C. M. John. Client-Side defense against Web-Based identity theft. In *NDSS*, 2004.

[2] EMVCo. EMV 4.2 specifications, 2008.

[3] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In *WWW'07*, pages 649–656, Banff, Alberta, Canada, 2007.

[4] A. Y. Fu, L. Wenyin, and X. Deng. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD). *Dependable and Secure Computing, IEEE Transactions on*, 3(4):301–311, 2006.

[5] Mastercard. Mastercard authentication solutions: Two-factor solutions designed to enhance security for online banking and e-commerce. mastercard.com.

[6] Panda Security. Millions exposed to identity theft. itnewsafrica.com.

[7] M. Savage. Phishing, malware to strain banks in 2009, Jan. 2009. techtarget.com.

[8] B. Schneier. Two-factor authentication: too little, too late. *Commun. ACM*, 48(4):136, 2005.

[9] Visa. Dynamic passcode authentications. visaeurope.com.

[10] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *WWW'07*, pages 639–648, Banff, Alberta, Canada, 2007.