A Secure Mobile Payment Model

Tao-Ku Chang
Department of Computer Science and Information Engineering,
National Dong Hwa University, Hualien, Taiwan
tkchang@mail.ndhu.edu.tw

Abstract—Mobile devices are most ubiquitous, consumers can now use their mobile devices to pay for a wide range of services and digital or physical goods. However, consumers' security concerns are a major barrier to broad adoption and use of mobile payments. In this paper, we design a secure payment model in which access control is based on service-oriented architecture. A consumer uses his/her mobile device to get authorization from the bank's server and generate a QR code as the payment certificate.

Keywords-Mobile Payment; Web Services; Security

I. INTRODUCTION

Mobile devices are prevalent and are as powerful and connected as personal computers or laptops. Gartner estimates that by 2013, mobile phones will replace personal computers as the most common web access device [1]. By that same year, Forrester predicts that 48 percent of all U.S. mobile phone subscribers will be smart phone subscribers, a striking jump from just 7 percent in 2008 [2]. A growing number of consumers can use their mobile phones as keys, cameras, and TVs. If mobile phones could also be a payment tool, it will be convenient. Although large-scale mobile payment systems are still in development, a number of mobile financial and mobile commerce applications (e.g., Starbucks app, iTunes, and Google Wallet) are helping to build user experience and encourage the adoption of mobile payments among consumers.

Industry analysts and service providers have identified a number of important drivers for the adoption of mobile payments, such as familiarity and comfort with using mobile technology, strong security, and greater convenience. Security and privacy risks are the major barriers to adoption. Consumers worry about their personal data being hacked or intercepted. They think mobile transactions are less secure than credit and debit card transactions. In fact, mobile payments can be just as or even more secure than traditional payment methods. When consumers are offered a secure online payment environment, which works via advanced mobile web systems, consumers do not need to cough up physical currency each time they want to make a mobile purchase or pay a bill online.

Consumers usually pay for their commodities with pre-paid card or credit card in supermarkets. Many pre-paid cards issued by the specific stores can not be identified when they are lost. Any one who picks up lost pre-paid cards can use it without being caught. Since the clerks seldom check the signature, people are usually not aware of their lost credit card being swiped by a few amount of money if they do not check their accounts regularly. To address this kind of problem is the goal of this paper. The remainder of this paper is organized as follows: Section Ⅱ presents the system architecture, Section Ⅲ presents our implementation, Section Ⅳ gives an overview of related works and technologies, and Section Ⅴ draws conclusions about the work described in the paper.

This work was supported in part by the National Science Council of Taiwan, R.O.C. under grants No. 102-2221-E-259-015.

II. SYSTEM ARCHITECTURE

In this paper, we propose an operational model to make sure the payment is secure with mobile devices. A fine-grained access authorization control is added to the proposed system. The system architecture is described in section *A*.

A. Payment mechanism and architecture

Figure 1 shows the system architecture for the proposed payment model. When a consumer shops and wants to check out, the payment steps are following:

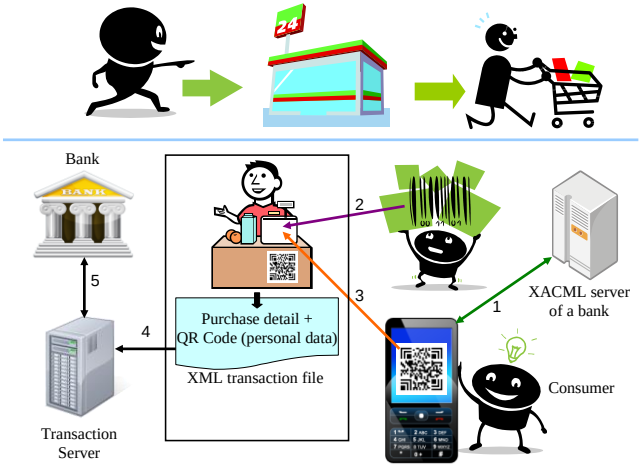


Figure 1: The payment mechanism and system architecture

(1) A consumer executes the App software provided by a bank. This App software allows the consumer to input their account login and password, and then it connects to the XACML server of a bank to check whether the consumer has authority to use

the service. Once authorized, the XACML server executes the QR code certificate service and response a short essential data. App software uses these personal data to generate a QR Code.

(2) The clerk uses a scanner to scan the bar-code on goods bought by the consumer. The total price and transaction data is kept in an XML data format.

(3) The consumer gives the QR Code on the smart phone to the clerk for scanning. The QR code is decoded by the system to retrieve personal data of the consumer.

(4) Personal data and purchase detail are stored into an XML transaction file and then transferred to the server.

(5) The transaction data in the server will be sent to the bank to do settlement process in a period of time.

In this payment mechanism, a consumer who finishes steps the 1 to 3, completes the payment process. Step 4 to 5 is inter-process between stores and banks. Each component of this architecture is described in Section *B–E*.

B. Access control model

The access control is handled by a web service and the security policy is defined and stored in an XACML server. A user can grant such as authority to another person to build a temporary policy, like an additional card of a credit card. The access control manager refers to the temporary and permanent security policy to decide whether a request is accepted or denied. This access control model is also appropriate for cloud computing. According to the data-flow model of XACML, we design an access control model depicted in Figure 2. The model operates by the following steps.

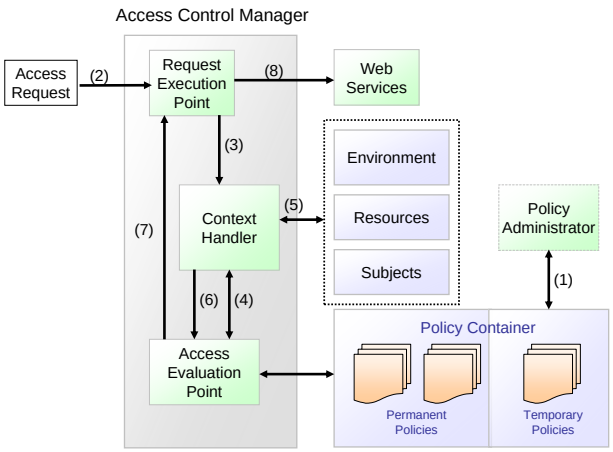


Figure 2: Access control model

(1) Policies in the policy container represent the complete policy for a specified target. Policy administrator writes these policies and makes them available to the access evaluation.

(2) The client sends an access request to the request execution point (REP) for a web service.

(3) The REP sends the request for access to the context handler in its native request format.

(4) The context handler constructs an XACML request context, and sends it to the access evaluation point (AEP). The AEP requests any additional subject, resource, environment and other attributes from the context handler.

(5) The context handler requests the attributes from subjects, resources, and the environment.

(6) The context handler sends the requested attributes and the resource to the AEP.

(7) The AEP evaluates the policy and returns the response to the REP.

(8) If access is permitted, the REP permits access to the resource and sends the client's request to the Web Service.

C. Authorization App

The authorization app is an interface that let consumers input an account login and password and then sends a request containing account login and password to an XACML server. This server verifies that users have authorization to use the QR Code certificate service. After authorization is given, the server responses accepted information with the consumer's personal data. The authorization-app program calls a QR Code encoder to generate a QR Code that is a payment certificate. The payment certificate has a time limit. If the consumer does not use it within the time limit, the QR Code will lose its efficacy. Moreover, it can be used one time only. For security considerations, these data could be encrypted before being encoded. The secure communication protocol, such as Hypertext Transfer Protocol Secure (HTTPS), could be used when data is transferred from the server to mobile devices.

D. QR Code encoder-decoder

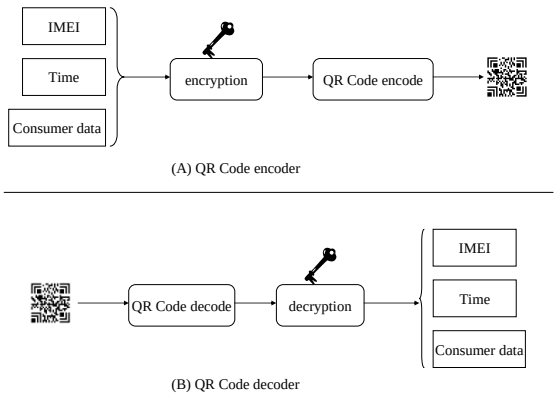


Figure 3: QR Code encoder and decoder

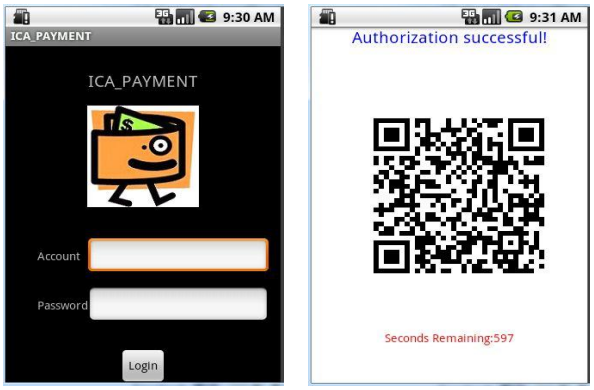
The QR Code encoder-decoder is implemented for mobile devices and cash registers. The encoder is called by the authorization-app program to generate a QR Code. The decoder is invoked when a QR Code is read from a mobile device by a bar code scanner. Figure 3 shows the operational model of a QR Code encoder and decoder.

Data encoded into QR Codes include international mobile equipment identity numbers (IMEI), time, and information of the consumer. Furthermore, QR Codes generated on a server site or client site is also a consideration.

E. XML securing tool and document security language

Sometimes purchase details such as personal id and account, are sensitive and important, and must be protected from even the database administrator. We need an XML securing tool to secure data because the transaction and personal information are formatted in XML. We apply XML encryption and signature technology to reach security consideration, and define a document security language (DSL) that describes how to encrypt and give a signature to an XML document. A DSL document contains key definitions, algorithm definitions, and signature definitions.

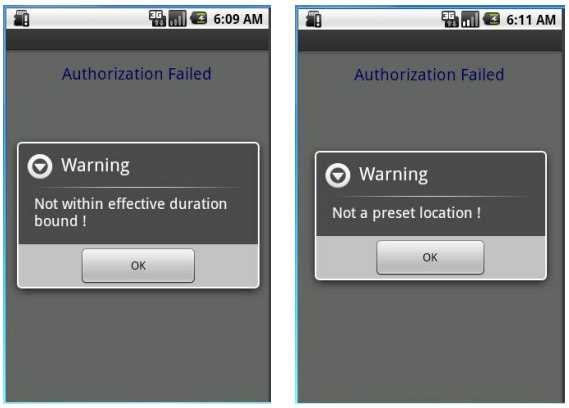
III. IMPLEMENTATION



(A) (B)

Figure 4: : The authorization-app program and the certificate

In this paper, we implement each component described in Section 4. The XACML server is developed by a Java platform and allows consumers to register and preset limits of each transaction. Also, consumers can temporarily and remotely stop their access right to generate QR Codes service once they want to drop this service. The authorization-app interface program is based on the Android platform. It allows consumers to input their account and password to get the authorization of generating a QR code as the payment certificate. In this implementation, the duration of the QR Code can be set to a minute countdown by consumers (see Figure 4(A) and Figure 4(B))



(A) (B)

Figure 5: : Examples of authorization failure

In this security mechanism, a consumer can preset the effective duration boundary of using the service, or the location of a supermarket where the consumer usually uses the service. The QR code certificate service is disabled when not in a valid duration boundary or a designated location. For example, authorization fails today, August 15, 2013 when the effective duration bound is from January 1 to July 31, 2013 (see Figure 5 (A)). It fails when using the service in Hualien City while the designated location is set to be in Taipei City (see Figure 5 (B)). Consumers can also preset an amount limit in one transaction, then the monitor of the cash desk gives a warning message when this transaction exceeds the amount (see Figure 6).

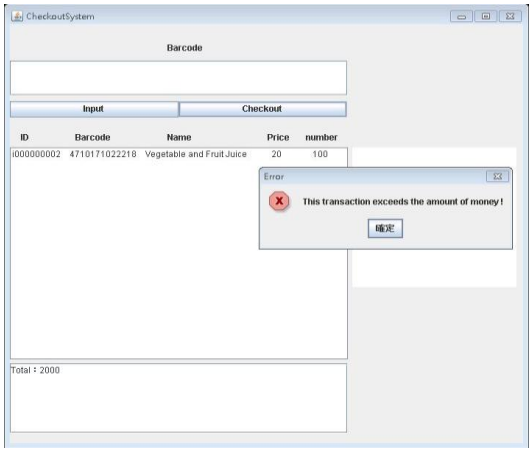


Figure 6: : An example of one transaction exceeds the amount of money

IV. RELATED WORK

In a survey of the Consumer Research Section of the Federal Reserve Board, over half of consumers believed that mobile contactless payments would become a major form of payment in the next five years, and over one-third of survey subjects indicated that they would use this method of payment if it were made available to them [3]. Mobile payments overall are expected to move toward the mainstream to reach $90 billion by 2017 for the US, according to the Forrester report [4].

Privacy and security are becoming ever more important to consumers given the rise of mobile payments and commerce, and continue to be a major obstacle to

widespread adoption. Specific security issues identified vary by survey. Some consumer reservations stem from fear of payment account information being intercepted, threat of unauthorized parties accessing personally identifiable information, and receipt of unsolicited promotional material [3, 5]. According to research from Synergistics, over half of mobile phone owners surveyed indicated identity theft as a top concern related to making mobile payments [6]. Over 50 percent of the consumers surveyed in a First Data mobile payments study believed that making a payment via mobile phone is less secure than making a payment in person or with a credit or debit card [7]. Regardless of the specific reason for the security concern, security issues must be addressed to achieve mass adoption of mobile payments. Many applications on smart phones are developed in Web services architecture [8]. Web service applications run over the open, and unreliable Internet, and Web service providers must ensure security issues like confidentiality, authentication, authorization and the like. There are number of solutions to solve the above problems such as XML encryption [9], XML signature [10], Security Assertion Markup Language (SAML) [11], Extensible Access Control Markup Language (XACML) [12], XML Key Management Specification (XKMS) [13], etc.

The main technologies of mobile commerce are near field communication (NFC), mobile wallet, Quick Response Code (QR Code) [14], etc. NFC is a set of short-range wireless technologies, typically requiring a distance of 10 centimeters or less to initiate a connection. Mobile wallet is a software application that is loaded into a mobile phone, and enables storage of multiple payments credentials and value-added services to be securely accessed in order to initiate mobile payments. A QR Code is a kind of two-dimensional symbology developed by Denso Wave (a division of Denso Corporation at the time) and released in 1994. It contains information in both vertical and horizontal directions and holds a considerably greater volume of information than a bar code. Over the past year, more consumers have been using their mobile devices to not only comparison shop in-store, but also shop via their handset. Many merchants and hypermarkets provide the mobile applications or QR Code for consumers to extend the market to mobile shopping. There are a lot of cases that use mobile device as a payment tool and apply a QR Code as a bridge between product information and consumers. Starbucks Card Mobile is a three-part system that includes 2D bar codes, scanners and mobile phone applications for iPhone, BlackBerry and Android. This system allows Starbucks consumers to pay with their phones at roughly 9,000 locations in the U.S. It has also helped the coffee company stand alone as the only large-scale mobile payments provider [15]. Amazon provided an augmented reality app called Flow [16] that lets consumers discover information about items by scanning QR Codes of products. In 2011, Google launched its Google Wallet app [17]. Consumers can use any card from Visa, MasterCard, American Express, or Discover in conjunction with the app. To pay in-store, select the card, and then just tap a smartphone to any contactless point of sale terminal. Google Wallet keeps you safe and secure. The app has its own PIN, and if you lose your phone, you can remotely disable your mobile wallet. 7-ELEVEN in Taiwan provided QR Code shopping. Consumers use smart phones to scan QR Codes in product advertisements, then connect to the 7-NET website to check out and take the products and pay at a near 7-ELEVEN store. This is the first example for mobile commerce in the convenience store industry in Taiwan.

V. CONCLUSION

It is possible that mobile payments can be more secure than traditional payment methods. The mobile device must be set up correctly with risk mitigation tools having the ability to remotely wipe, delete, lock, and disable a lost or stolen mobile phone, with anti-virus and malware software, and with multiple layers of security to lock both the phone and access to the secure mobile wallet – and the consumer must use the mobile payment capabilities correctly. Furthermore, consumers must understand that they also have responsibilities to protect their payment account credentials and mobile devices. Consumers also need to be educated on what not to do, such as download untested, questionable, uncertified applications or share their mobile phones.

REFERENCES

[1] "Gartner's Top Predictions for IT Organizations and Users, 2010 and Beyond," Gartner, Inc, January 13, 2010.

[2] "Forrester Research Mobile Adoption and Sales Forecast, 2010 To 2015 (US)," Forrester Research, Inc, January 2011.

[3] Matthew B. Gross, Jeanne M. Hogarth, and Maximilian D. Schmeiser, "Consumers and Mobile Financial Services," Board of Governors of the Federal Reserve System, March 2012.

[4] "Mobile Payments Forecast, 2013 To 2017," Forrester Research, Inc, January 16, 2013.

[5] "Consumers and Convergence V: The Converged Lifestyle," KPMG LLP, December 2011.

[6] "Mobile Payments: Consumer Viewpoint," Synergistics Research Corporation, August 2011.

[7] "Consumers Going Mobile: The Transformation of Payments," First Data, November 2011.

[8] "Web Services Architecture, W3C Working Group Note 11 February 2004." http://www.w3.org/TR/ws-arch/.

[9] Takeshi Imamura, Blair Dillaway, and Ed Simon, "XML Encryption Syntax and Processing. W3C Recommendation 10 December 2002," http://www.w3.org/TR/xmlenc-core/.

[10] Mark Bartel, John Boyer, Barb Fox, Brian LaMacchia, and Ed Simon, "XML-Signature Syntax and Processing W3C Recommendation," 12 February 2002. http://www.w3.org/TR/xmldsig-core/

[11] Eve Maler, Prateek Mishra, and Rob Philpott, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1," OASIS Standard, 2 September 2003.

[12] "eXtensible Access Control Markup Language (XACML) Version 3.0," OASIS Standard, 22 January 2013.

[13] Warwick Ford, Phillip Hallam-Baker, Barbara Fox, Blair Dillaway, Brian LaMacchia, Jeremy Epstein, Joe Lapp, "XML Key Management Specification (XKMS), W3C Note 30 March 2001," http://www.w3.org/TR/xkms/

[14] "QR Code Essential," Deso ADC. http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo%3D&tabid=1426&mid=4802

[15] Starbucks Card Mobile App. http://www.starbucks.com/coffeehouse/mobile-apps (accessed April 5, 2013)

[16] Amazon Flow App. https://play.google.com/store/apps/details?id=com.a9.flow&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5hOS5mbG93Il0 (accessed April 5, 2013)

[17] Google Wallet. http://www.google.com/wallet/