A Secure Mobile Banking Using Kerberos Protocol

Dr. Mohammad N. Abdullah* & May T. Abdul-Hadi**

Received on:30/7/2007
Accepted on:6/10/2008

Abstract

Because the network is an open environment, a lot of care must be taken when transferring sensitive information especially when related with financial data. This depends on the principals to be trusted which is a problematic and since the first step in network security is the authentication, this paper presents a proposed modal for secure mobile banking (m-banking) applications using Kerberos (the network authentication protocol).

The aim of this paper is to establish a secure communication between the clients and mobile-bank application server in which they can use their mobile phone to securely access their bank accounts, make and receive payments, and check their balances.

The integration of smart card into classic Kerberos enhances the security for client authentication by storing the cryptographic keys and perform dual factor authentication. Other enhancement to Kerberos is the PKINIT in which the shared key is mapped with public- private key. To build a robust client authentication the client uses his/her mobile phone to author Kerberos's messages, process the replay and establish secure communication with the application server.

Keywords: Authentication, m-bank, mobile, Kerberos, PKINIT, smart card, client\server, shared key, public- private key, cryptographic, applet.

نظام مصرفي نقال آمن بأستخدام كربيروس بروتوكول

الخلاصة

لأن الشبكةَ الدولية للانترنت بيئةٌ مفتوحةٌ، الكثير مِنْ العنايةِ يجب أنْ تُؤْخَذَ بنظر الأعتبار عندما يتم نقلِ المعلوماتَ الحسّاسةِ خصوصاً عندما تتَعلّق بالبياناتِ الماليةِ وهذا يَعتمدُ على الأفراد الّذين من الصعب ان يَكُونوا جديرين بالثقةَ و بما ان الخطوةِ الأولى في أمنِ الشبكة هو التحقُّقُ من هوية المستخدم فأن هذا البحث يقدّمُ نموذج مقترح لتطبيقاتِ المصرفِ النقّالةِ الآمنةِ التي تَستعملُ كربيروس (نظام تحقّقِ الشبكةّ).

إنّ هدفَ البحث هو انشاء إتصالاً آمناً بين الزبائنِ والمصرف الالكتروني بطريقة يُمْكِنُ أنْ يَستعملوا هاتفهم الجوَّالَ للدُخُول الى حساباتهم المصرفية بشكل آمن والقيام بتحويل الأرصدة أو استلام الدُفْعاتَ المالية أوتدقيق الأرصدة.

إنّ اضافة البطاقةِ الذكيةِ إلى كربيروس الكلاسيكي يُحسّنُ الأمنَ للتحقُّقِ من الزبونِ بخَزْنِ المفاتيحِ المشفّرةِ (كلا من مفتاح الزبون و مفتاح المصرف الإلكتروني) وهذا ما يسمى بالتحقُّقَ ثنائي العاملِ، وكذلك يمَنْع هجومِ القوةِ العنيفِ و غَشّ الشبكة ومشكلة الخزن للمفاتيح. وكذلك فأن التحسين الآخر للكربيرس هو ال PKINIT الذي يتم فيه إستعمال المفتاحِ الخاصِّ لتشفير الرسالة والمفتاح العام لفك التشفير الرئيسيِ بدلا من استخدام المفاتيح المشتركة بين الزبون والمصرف الإلكتروني. لبناء نظام تحقق متين سوف يتم استخدام الجهاز النقال الخاص بالزبون لتأليف رسائل كربيرس والتعامل مع الرد ثم تحقيق الاتصال الامن مع السيرفر الذي يستصيف التطبيق المصرفي.

https://doi.org/10.30684/etj.27.6.10
2412-0758/University of Technology-Iraq, Baghdad, Iraq
This is an open access article under the CC BY 4.0 license http://creativecommons.org/licenses/by/4.0

1. Introduction

The security of the banking operations is a tricky subject because of the cheaters who find any opportunity to deceive and take the others money. The customers' identification is the most important subject in the banking operations, which passes through several development stages. In the past and in the computerless society it depends on the bankers' knowledge of the customer identity, which was accepted because of the small community in those days.

After the invention of the computers and the development of the technologies with the growth of the network appeared the automatic banking in the cashless society. This is called the electronic banking or e-banking. The rapid expansion in the smart card application especially in the e-banking strengthens the security, but still some vulnerabilities must be taken in consideration.

The wireless communication with the mobile device and smart card helps to adopt the e-banking using the customer's mobile phone and hence it is called now mobile banking or m-banking for short. The latter performs more security and takes a wide range of enhancement to take care of the customer demand of protection and the ease of use [1].

2. Kerberos Protocol

In Greek mythology Kerberos is a three-headed dog guarding the entrance to the underworld.

Kerberos (in the concept of this paper) is a network authentication protocol developed by the Massachusetts Institute of Technology (MIT) as part of Athena project in mid 1980's.

Kerberos is designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos still relies on the user providing some form of credentials to verify their identity. The exchange of credentials is encrypted throughout the entire authentication process, enabling a secure authentication mechanism. From the user point of view, it does not differ much from a normal sign-on process. The major difference is that after an identity is proven, a temporary *ticket* is issued to the client. This ticket allows the user to access other systems and applications that exist within the circle of trust, or more correctly, the *Kerberos realm*.

Microsoft introduced Kerberos as the new default authentication protocol for enterprise environments in Windows 2000. Every Windows 2000, Windows XP, and Windows Server 2003 OS platform includes a client Kerberos authentication provider.

Kerberos implementation is based on the Internet Engineering Task Force (IETF) standard defined in RFC 4120 which use symmetric cryptography. Kerberos uses a unique ticketing system that provides faster authentication, also provides the following security services: mutual authentication, delegated access control, privacy and data integrity. Kerberos can provide single sign-on (SSO) and support for smart card logon feature. The Kerberos has advantages over other authentication protocols first the password never travels throw the network even in encrypted form (but used for encryption/decryption on the client workstation memory), second Kerberos does not depend on the firewalls because it does not consider that attack could came from the intruders [1, 2, 3].

Kerberos contains of three basic entities the client and the server whom want to authenticate and the Key Distribution Center (KDC), the KDC is a single process comprised of two services: the Authentication Service (AS) and the Ticket-Granting Service (TGS). The AS issues Ticket Granting Tickets (TGTs) to authenticated principals (that is, users, machines, services) for admission to the TGS. The TGS issues tickets for admission to other services in the domain or to a TGS in another trusted domain. Any domain controller can accept authentication requests and ticket-granting requests

PDF created with pdfFactory Pro trial version www.pdffactory.com

addressed to the domain's KDC. Kerberos is based on the idea of *tickets*. A ticket is just a data structure that wraps a cryptographic key, along with some other bits of information. A KDC distributes Kerberos tickets to authenticated users. A KDC issues two types of tickets:

- A master ticket, also known as the ticket granting ticket (TGT)
- A service ticket

A KDC first issues a TGT to a client. The client can then request several service tickets against his or her TGT. To explain how TGTs and service tickets work, let's consider the following key exchange scenario:

1. A client sends a message to the KDC requesting a TGT. The request includes the username of the client, but does not include his password (the password is used to generate user's key).

2. The KDC issues a TGT reply message to the client that contains a session key in encrypted form. To encrypt the session key, the KDC uses a key derived from the client's password which is already stored in the KDC's database. The KDC also issues a TGT ticket for the TGS.

3. The client decrypts the encrypted part of the reply message and extracts the session key from it. The client then authors a request for a service ticket. A service ticket is valid only for communication between two parties (the client and the server whom the client wants to communicate). The server should be already registered with the KDC.

4. The KDC authors a service ticket for the server. This ticket contains the client's authentication data and a new cryptographic key, called a *sub-session key*. The KDC encrypts the service ticket with the secret key of the server (the secret key is a shared secret between the KDC and the server). This means that only the server can decrypt the service ticket. The client copy of the sub-session key is in the encrypted part of the message and encrypted using the session key that the client extracted from the previous message.

5. The client decrypts the message received from the KDC and fetches the sub-session key inside the message as well as the service ticket. It sends the service ticket to the server.

6. The server receives the service ticket and decrypts it to fetch the authentication data of the requesting client as well as the sub-session key. The server then acknowledges the client's request, and a new secure session is established between the client and the server. Both client and server now possess the same sub-session key, which they can use for secure communication with each other. Figure (1) illustrates the Kerberos protocol and message exchange. The client can repeat steps 3 through 6 above for another server application. This means that our Kerberos service can be used to share authentication data and that the same client (which represents a single user) can authenticate with different applications. This effectively enables Single Sign On (SSO) [1, 4, 5]

2.1 Integration of Smart Card into Kerberos

A smart card is a plastic card embedded with a computer chip as an Integrated Circuit Card (ICC) that stores and logs data transaction. This data can either be a value, information or both. It is stored and processed within the card's chip. Smart cards have some advantages; they can easily store large passwords, perform advanced security functions like storage of cryptographic keys and have ability to perform cryptographic algorithms. Smart cards are also provide tamper-resistant storage for protecting sensitive information like private keys, account numbers, passwords, and other forms of personal information. They can isolate security-critical computations that involve authentication, key exchange and digital signatures from other parts of the system. The weaknesses in the Kerberos 5 Protocol have long been known to have vulnerabilities:

-offline password-guessing attacks
-network spoofing
-key storage (master key and session key)

PDF created with pdfFactory Pro trial version www.pdffactory.com

This security problems resulting from the software-only based implementation of Kerberos that can be improved by adding hardware (e.g. smart card). The integration of smart card into Kerberos will enhance the security and perform dual factor authentication which is: Something you know (password or PIN) and something you have or process (smart card) [6, 7].

2.2 Kerberos-PKINIT

Windows 2000 and Windows Server 2003 include extensions to Kerberos Version 5 to support public-key based authentication. These extensions are known as PKINIT which stands for use of Public Key cryptography for INITial authentication and are defined in an IETF Internet draft available as RFC 4556. PKINIT enables the smart card logon process to a Windows 2000 or later domain. PKINIT allows a client’s master key to be replaced with its public key credentials in the Kerberos Authentication Request and the reply messages. Table (1) illustrates the mapping of the Standard Kerberos “Master Key” to the PKINIT “Public-Private Key”. PKINIT introduces a trust model in which the KDC is not the first entity to identify the users (as is the case for classical Kerberos). Before KDC authentication, users are identified by the certification authority in order to obtain a certificate. In this new model the users and the KDC obviously both need to trust the same Certification Authority (CA). The Kerberos smart card logon process works and the steps when the client authenticates using his smart card are:

1. The client starts the logon process by introducing his smart card and by authenticating to the card using his PIN code. The smart card contains client’s public key credentials, his private key and certificate.
2. A TGT request is sent to the KDC (AS); this request contains client’s principal name and a timestamp that are digitally signed using client’s private key along with a copy of client's certificate.
3. To validate the request and the digital signature on it, the KDC will first validate the client’s certificate. The KDC will then query the Active Directory for a mapping between the certificate and a Windows account. If it finds a mapping, it will issue a TGT to the corresponding account.
4. The KDC sends back the TGT to the client. The client’s copy of the session is encrypted with his public key.
5. To retrieve his copy of the session key, he uses his private key.

When a smart card is used in place of a password, a private/public key pair stored on the user’s smart card is substituted for the shared secret key derived from the user’s password. In the public key extension to the Kerberos protocol, the initial AS Exchange is modified so that the KDC encrypts the user’s logon session key with the public half of the user’s key pair. The client decrypts the logon session key with the private half of the pair [2, 8].

2.3 The Wireless Communication

Wireless communications is a huge field, encompassing everything from radio and television broadcasting through pagers, mobile phones, and satellite communications. The field of mobile phones is expanding very fast at the same time that standards and protocols are being adopted, used, updated, and sometimes discarded. Mobile Commerce (M-Commerce) is the ability to conduct commerce, using a mobile device e.g. a mobile phone (or cell phone), a Personal Digital Assistants (PDA), a smartphone while on the move, and other emerging mobile equipment, like dashtop mobile devices. Banks and other financial institutions are exploring the use of mobile commerce to allow their customers not only to access account information, but also make transactions, e.g. purchasing stocks, remitting money, via mobile phones. This service is often referred to as Mobile Banking or M-Banking [9].

3. The Proposed Model Design

The proposed model in this paper is designed to perform a robust client

PDF created with pdfFactory Pro trial version www.pdffactory.com

authentication for m-bank application; this can be done by using the following stages: First, using Kerberos and its unique ticketing mechanism for mutual authentication. Second, the integration of smart card into Kerberos classic will enhance the client authentication. Third using Public Key Cryptographic (Kerberos-PKINIT) and digitally signed the message. Fourth the replacement of the client's workstation with his\her mobile phone will invoke the J2ME technology. Figure (2) illustrates different scenarios of Kerberos clients in mobile bank application (MyBank); the Data Base in the KDC holds the credentials of each client registered in the domain (MyDomain.com). Alice and Bob are mobile clients and the must secure as proposed in the model design, the bank can also receive request from cross-realm, slave KDC ,classic Kerberos and integrated smart card into Kerberos clients as shown in figure (2).

The design of the proposed model has the following concepts:

1. The Kerberos is based on PKINIT which implements public key cryptography. The client sends a signed (with user's private key) in the request for a TGT along with the user's certificate.
2. The client is authenticated to the server using dual factor authentication which are PIN (something you know) and smart card (something you have).

The user's secret key is of two parts, the *userPIN* key and the *bankPIN* key that are installed in the installation process of the Java card applet and the Java card technology ensures that the bankPIN's key are never exposed to any client application accessing the Java card application. The client uses his\her userPIN's key to logon to the application and if the number of unsuccessful logon pass 10 then the card locked and the applet must follows new installation.

3. The client uses his mobile phone to request TGT, process the reply (with the KDC) and establish secure communication (with the application server).

To demonstrate the proposed model of this paper the following steps are required:

- Designing a Java card applet that works as key manager to hold the userPIN's key and the bankPIN's key during the installation and for later use for encryption and decryption. The Java card applet is installed in the Java Card Runtime Environment (JCRE) of the Smart Card.
- Designing a J2ME MIDlet for client graphic user interface (GUI) and deploying this MIDlet in Wireless ToolKit 2.2 using Over The Air (OTA) provisioning.
- Designing a Servlet in the server side for establishing end to end Java application and deploying using Jakarta TomCat Apache server [1].

4. Conclusions

1. Kerberos- PKINIT can be adapted in a mobile computing platform to provide authentication mechanism for mobile users in mobile banking application.
2. The m-banking application uses a SIM card on user's device as a trusted computing platform and also a secure storage for security credentials like cryptographic keys. While the client's mobile phone can host the Java side client application including authoring a ticket request, processing the response and establishing a secure communication with the application server.
3. The user's key has traveled between the J2ME MIDlet and the Java Card applet in plaintext form, because the Java Card resides inside the account holder's J2ME cell phone, no need to secure the communication between the MIDlet and the Java Card applet.
4. If the client's key is compromised the whole authentication process compromise. If the session key is compromised all services registered to the AppServ compromise (per session key duration time). If the sub-session key compromised the requested service will compromise (per sub-session key duration time)

PDF created with pdfFactory Pro trial version www.pdffactory.com

5. References

[1] May Tariq, A Secure Mobile Banking using Kerberos Protocol, Master's Thesis, University Of Technology, Control and Systems Engineering Department, July 2008.

[2] Jan De Clercq, *Windows Server 2003 security infrastructures*, Elsevier Digital Press, 2004.

[3] C. Neuman, T. Yu, S. Hartman, K. Raeburn, The Kerberos Network Authentication Service (V5), Request for Comments RFC 4120 (Obsoletes RFC 1510), Network Working Group, IETF Standards Track, July 2005.

[4] Sun Microsystems, Inc, Smart Card Logon, Microsoft Windows 2000 server operating system, White Paper, 1999

[5] Faheem Khan, Simplify enterprise Java authentication with single sign-on, IBM, developerWorks, September 2005

[6] Tanmay K.M, Smart Card Crash Course, (online) Technical articles, 2003, last modified 5/22/2007, http://members.tripod. com/tanmaykm/work/articles/smartcard.html

[7] Jesus Molina, Hardware implementation of an authentication protocol using Kerberos, Master's Thesis, Universitat Politècnica de Catalunya, 2000

[8] Larry Zhu and Brian Tung, Public Key Cryptography for Initial Authentication in Kerberos (PKINIT), Network Working Group, June 2006, RFC 4556, IETF Standards Track

[9] Sun Developer Network (SDN), Sun Mobile Device Technology - Introduction to Mobility Java Technology, Sun Microsystems Inc.

Table 1: Mapping the Master Key to the Public-Private Key

Standard Kerberos usage of Master Key	**PKINIT**
Client-side encryption of the pre-authentication data	Private key
KDC-side decryption of the pre-authentication data	Public key
KDC-side encryption of session key	Public key
Client-side decryption of session key	Private key

PDF created with pdfFactory Pro trial version www.pdffactory.com

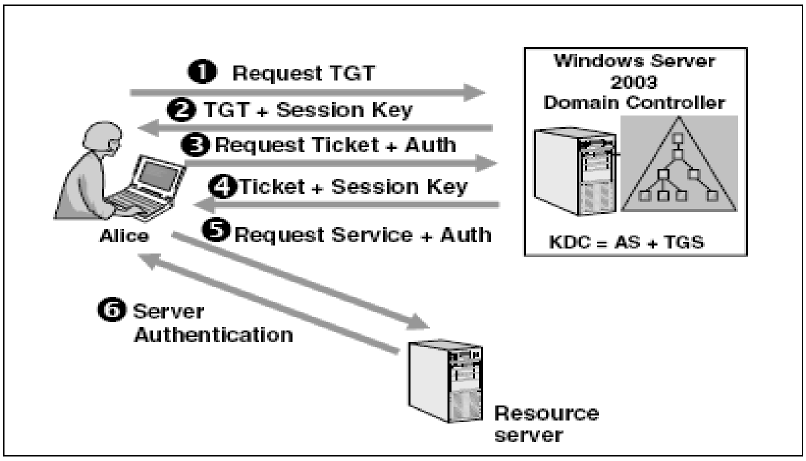

Fig. 1: Kerberos Protocol

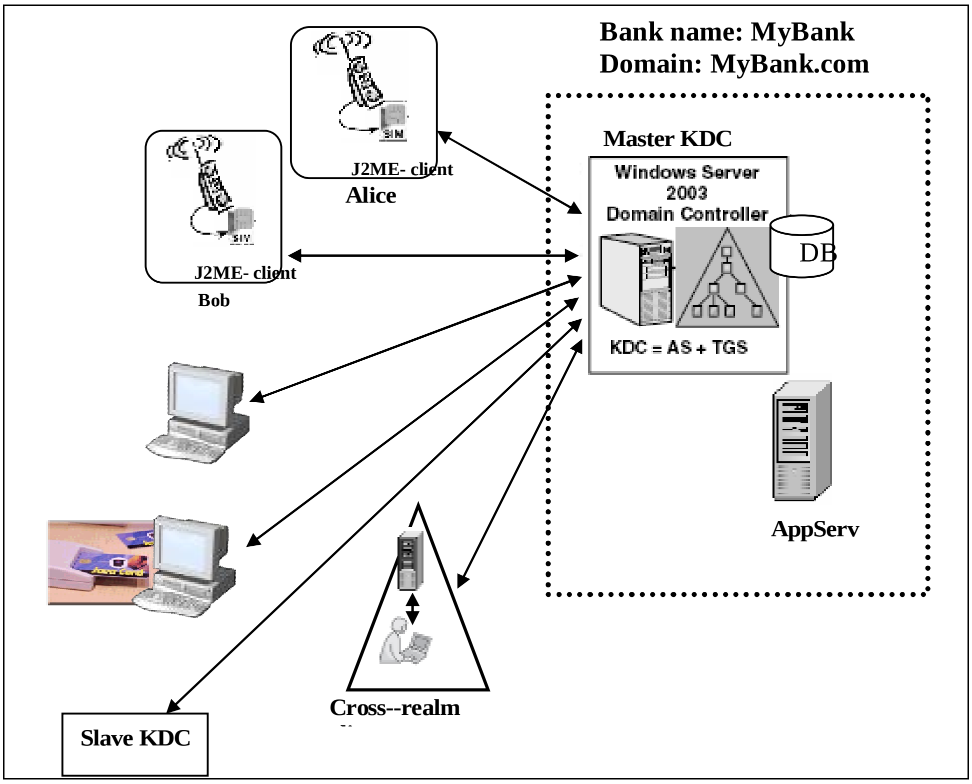

Fig. 2: Kerberos clients for Mobile –Banking Application

PDF created with pdfFactory Pro trial version www.pdffactory.com