A Secure Banking System Based on Client/Server Architecture

Ma Theingi Tun

Computer University (Mandalay)

Abstract

Online banking has made it easy to carry out the personal or business financial transaction without going to bank and at any suitable time. This facility enables to transfer money to other accounts and checking current balance alongside the status of any financial transaction made in the account. In order to maintain privacy and to avoid any misuse of transactions, it is necessary to follow a secured architecture model which ensures the privacy and integrity of the transactions. This system is developed by using hash function named Message Digest Function (MD5) and RSA algorithm. The purpose of this project is to develop an on-line banking system that provides customers with the facility to check their accounts and do transactions on-line. The system will provide all the banks facilities to its customers when their authentications (user-id and password) match. This system is implemented by using Java programming language.

1.Introduction

Online banking is defined as the automated delivery of new and traditional banking products and services directly to customers through electronic and interactive communication channels. Online banking includes the systems that enable financial institution customers, individuals or corporates to access accounts, transact business, or obtain information on financial products and services through a public or private network. Online banking is changing the banking industry and is having the major effects on banking relationships. Banking is now no longer confined to the branches where one has to approach the branch in person to withdraw cash or deposit a cheque or request a statement of accounts. In this system, any inquiry or transaction is processed online without any reference to the branch at any time.

Banking on-line automates many of these processes, saving time and money. For all banks, online banking is a powerful tool to gain new customers while it helps to eliminate costly paper handling and manual teller interactions in an increasingly competitive banking environment. In this system, security is a crucial requirement. The security is required to protect customers' privacy and to protect against fraud.

Any internet banking system must solve the issues of authentication, confidentiality, integrity and nonrepudiation which means it must ensure that only qualified people can access an internet banking account. Most of the attacks on online banking used today are based on deceiving the user to steal login data and valid transaction number (TAN).

2. Theory of Cryptography System

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the internet) so that it cannot be read by anyone except the intended recipient. Cryptography has become widely used tool in communication, computer networks, and computer security generally. Cryptosystems are the methods and functions used to encrypt and decrypt data. A variety of methods have been used, in the past, to hide information by the use of encryption. Encryption has been an adopted method for a long time thoughtout history. There are two types of cryptosystem. These are Secret Key (Symmetric Cryptosystem) and Public Key Cryptosystem (Asymmetric Cryptosystem).

Symmetric cryptosystem is sometimes called conventional cryptosystem. In symmetric cryptosystem, the encryption key and the decryption key are the same. These cryptosystems are also called secret key cryptosystem. The sender and receiver agree on a key before they communicate securely.

Public key cryptosystems are also called asymmetric cryptosystems. Public key is commonly used to identify a cryptographic method that uses an asymmetric-key pair: a public key and a private key. Public key encryption uses that key pair for encryption and decryption. The public key is made and is distributed widely and freely. The private –key is never distributed and must be kept secret. Given a key pair, data encrypted with the public key can only be decrypted with its private key; conversely, data encrypted with the private key can only be decrypted with its public key. This characteristic is used to implement encryption and digital signature.

An attempted cryptanalysis is called an attack. There are several types of attacks that a cryptanalyst may use to break a code, depending on how much information they have. Cryptanalytic

attacks are generally classified into six categories that distinguish the kind of information the cryptanalyst has available to mount an attack. The categories of attack are listed here roughly in increasing order of the quality of information available to the cryptanalyst or equivalently, in decreasing order of the level of difficulty to the cryptanalyst. The objective of the cryptanalyst in all case is to be able to decrypt new pieces of ciphertext without additional information. The ideal for a cryptanalyst is to extract the secret key. Ciphertext-Only Attack is one in which the cryptanalyst obtains a sample of ciphertext, without the plaintext associated with it. This data is relatively easy to obtain in many scenarios, but a successful ciphertext-only attack is generally difficult, and requires a very large ciphertext sample.

3. RSA and MD5 Algorithms

In cryptography, RSA (which stands for Rivest, Shamir and Adleman who first publicly described it) is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations. The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.

Alice transmits her public key (n,e) to Bob and keeps the private key secret. Bob then wishes to send message M to Alice.

He first turns M into an integer $0 < m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to:

$$c=m^e \bmod n$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

The security of the RSA cryptosystem is based on two mathematical problems: the problem of factoring large numbers and the RSA problem. Full decryption of an RSA ciphertext is thought to be infeasible on the assumption that both of these problems are hard, i.e., no efficient algorithm exists for solving them. Providing security against partial decryption may require the addition of a secure padding scheme.

The RSA problem is defined as the task of taking the roots modulo a composite n: recovering a value m such

that $c = m^e \bmod n$, where (n,e) is an RSA public key and c is an RSA ciphertext. Currently, the most promising approach to solve the RSA problem is to factor the modulus n. With the ability to recover prime factors, an attacker can compute the secret exponent d from a public key (n,e), then decrypt c using the standard procedure. To accomplish this, an attacker factors n into p and q, and computes $(p - 1)(q - 1)$ which allows the determination of d from e. No polynomial-time method for factoring large integers on a classical computer has yet been found, but it has not been proven that none exists. See integer factorization for a discussion of this problem. Rivest, Shamir and Adleman have shown that finding d from n and e is equally hard as factoring n into p and q. However, this proof does not imply that inverting RSA is equally hard as factoring.

As of 2010, the largest (known) number factored by a general-purpose factoring algorithm was 768 bits long using a state-of-the-art distributed implementation. RSA keys are typically 1024–2048 bits long. Some experts believe that 1024-bit keys may become breakable in the near term (though this is disputed); few see any way that 4096-bit keys could be broken in the foreseeable future. Therefore, it is generally presumed that RSA is secure if n is sufficiently large. If n is 300 bits or shorter, it can be factored in a few hours on a personal computer, using software already freely available. Keys of 512 bits have been shown to be practically breakable in 1999 when RSA-155 was factored by using several hundred computers and are now factored in a few weeks using common hardware. A theoretical hardware device named TWIRL and described by Shamir and Tromer in 2003 called into question the security of 1024 bit keys. It is currently recommended that n be at least 2048 bits long. In 1994, Peter Shor showed that a quantum computer could factor in polynomial time, breaking RSA.

4.System Design and Implementation

Prototype Web application "Banking" System is under construction for two roles of users: (Customer and Administrator). Modelling of prototype Web application "Banking" System in notation UML we shall begin with development of Use Case diagrams.

Except for these two roles, we shall enter still "abstract" role of simple "User". Any "User", in case of his registration in Web application "Banking" System, becomes «The Registered User» and banking application starts to play a role of "Customer", or "Administrator".

This system is described in two parts, one part is for signature generation at sending site and another for signature verification at receiving site. In this system, RSA and MD5 algorithms are used for signing and verifying a digital signature.

To generate digital signature, sender creates a message digest of the information to be sent. He/She represents this digest as an integer m between 0 and n-1, and uses sender's private key (n,d) to compute the signature: $s= m^d \bmod n$

Then, sender sends this signature s to the receiver.

To verify digital signature, receiver uses sender's public key (n,e) to compute integer :$v = s^e \bmod n$. He/She extracts the message digest from this integer, and independently computes the message digest of the information that has been signed. If both message digests are identical, the signature is valid.

MD5 is used for the hash value of the message. MD5 processes variable length message into fixed length output 128 bit and calculates a cryptographic hash or secure checksum of plaintext. The hash is appended to the message before encryption. On decryption, the hash is recomputed from the resulting plaintext. The plaintext hash decrypted to a different value from before, the hash computation leads to a different value. If the value does not match the expected message is invalid.

Figure 3 shows the signature generation process for customer to create account, withdraw and transfer money. First, the customer must need to fill up the requirement satisfaction and then he/she receives account no and security code. Then, the customer must type for security code and sign. The system generated message digest by using MD5 algorithm and signature by using RSA with sender's private key.

5.Conclusions

The system has applied the combination of asymmetric (RSA) and Message Digest (MD5) algorithm in order to get the advantages of message digest and asymmetric algorithms. Nowadays, Internet is very widely useful all over the world. By using the internet, the message is transferred from one place to another in everywhere. The customer can use this system anywhere that has been connected online. The customer who does not have own PC and without online connection can also be able to use this system. By using this system, the safety condition in the exchanging of cash is successful. For securely and privately transmitting the data over the Internet, most protocol use both

public key and secret key cryptography. To implement public key cryptography, RSA algorithm is used with the key size of 1024-bit. The client/server architecture include the RSA and MD5 in the internet banking environment to enrich the privacy and integrity of the sensitive data transmitted between the clients and the application server.

6.References

[1] Annop, MS (2007), "**Public Key Cryptography-Applications Algorithms and Mathematical Explanations**", India:Tata Elxsi. http//www.dkrypt.com/home/pkcs.

[2] Behrouz A.forouzan, "**Cryptography and Network Security**", MCGRAW-HILL, International Edition 2008.

[3] B.kaliski, [CMSRKEM] JRandall, Use of the "**RSA-KEM Key Transport Algorithm"** in CMS,draft-ietf-smiayme-rsa-kem-05.txt,September 2007.

[4] Channu kambalyal, "**3-Tier Architecture**", Client/Server news group URL:news.comp.client-server.

[5] D.Hemmendinger, A.Ralston, E.D.Reilly, eds. "**Client/Server Term Definition**", International Thomson Computer Publishing 1998.

[6] Dobbertion, Hans (1996), "**The Status of MD5 After a Recent Attack**" (fttp://ftp.rsasecurity.com/pub/cryptobytes/crypo2n2.pdf).

[7] GrayC.kessler, " **An Overview of Cryptography**", May 1998.

[8] R.Rivest, S.Shamir and L.Adleman, "**A Method for obtaining Digital Signatures and Public Key Cryptosystems**" http://www.articsoft.com/wp-pik-intro-html.

[9] Setiz J,Stickel E, "**Journal of Intetnet Banking Commerce",** Aprial 2009, vol.14,no.1

(http://www.arraydev.com/ commerce/ jibc/).

[10] William Stallings, "**Cryptography and Network Security**", Third Edition.

[11] http:// www.faqs.org/rfcs/rfcs1321.html.

[12] http:// www.en.wikipedia.org/wiki/SHA hash functions.

[13] http:// www.rsa.com/rsalabs/node.asp.[TWIRL and RSA Key Size]