Jurnal Teknologi, 38(D) Jun. 2003: 103–124

AN AUTHENTICATION METHOD FOR SECURE INTERNET TRANSACTION USING SMART CARD AND SECURE COPROCESSOR

MUSTAFA A. AYAD[1], V. PRAKASH[2], N. K. NOORDIN[3] AND
W. A. WAN ADNAN[4]

Abstract. Consumers for years have put their trust in physical means of proving personal identity. However, with the advent of Smart Cards, creating the same sense of trust in the security world is a challenge. Smart Cards, carrying digital signatures, private keys and certificates, offer an approach to trust in the security world that integrates the familiar card form with the capability to provide strong authentication. The same characteristics that make Smart Cards attractive change the threat environment to secure communications in traditional computer systems. In this paper, we propose new methods of authentication in systems that use Smart Cards and secure coprocessor to ensure secure Internet transactions and provide solutions on how to defend against some of the possible attacks.

Keywords: Smart card, authentication, security, internet, and coprocessor

Abstrak. Sekian lama pengguna telah meletakkan kepercayaan secara menyeluruh dalam menunjukkan pengenalan diri secara fizikal. Walau bagaimanapun, dengan terciptanya Kad Pinta, ia telah memberi sejenis kepercayaan yang sama dalam dunia keselamatan. Kad Pintar mengandungi tandatangan digital, kekunci, lesen dan sijil-sijil persendirian, yang mana telah berjaya meningkatkan kepercayaan di kalangan pengguna dengan kesahihan maklumat serta ciri-ciri keselamatan yang tinggi. Selain itu, Kad Pintar menjadi lebih menarik dengan keupayaan untuk mewujudkan sistem komunikasi yang lebih selamat dalam sistem komputer yang telah sedia ada. Dalam kertas kerja ini, kami memperkenalkan beberapa kaedah baru dalam sistem yang menggunakan Kad Pintar dan pemproses-bersama untuk memastikan keselamatan dalam transaksi internet serta menyediakan penyelesaian untuk menghalang kebarangkalian jika berlaku sebarang masalah dalam sesebuah sistem.

Kata kunci: Kad pintar, pengesahan diri, keselamatan, internet, dan pemproses-bersama

1.0 INTRODUCTION

The wide spread of the Internet and E-commerce has raised concerns about what is the most secure way to conduct transactions in the Internet. As the Internet evolves, we can anticipate conducting more of our typical transactions through the World Wide

[1–4] Department of Computer and Communication Systems Engineering, University Putra Malaysia, 43400, Serdang, Malaysia.
[1]maayd@yahoo.com,[2]prakash@eng.upm.edu.my,[3]nknordin@eng.upm.edu.my
[4]wawa@eng.upm.edu.my

Web [1]. The Smart Card provides the advantage of potential user mobility and sequential access to one machine by multiple users. Smart cards are considered a viable secure option to provide user authentication when conducting transactions from terminals other than our home computer by requiring users to insert a Smart Card into a reader and enter a Personal Identification Number (PIN) to grant access. Although this is a valid reasoning, the physical nature of the Smart card adds new vulnerabilities to our user-network interface that attackers can use to gain unauthorized access to private information.

The security of credit card transactions remains as the number one concern both for the Internet users who have yet to make an online purchase, and for those who have performed online transactions. The US National Consumer's League Internet Fraud Watch (NCL) has reported that American consumers lost over $3.2 million to online scams in 1999, a 38 percent increase over the previous year [2]. Importantly, the vast majority of cases reported to the NCL involved payments by cheque or money order, with credit card transactions accounting for only three percent of cases. An online survey in January, March 2000 found that 41 percent of regular Internet users had shopped online two to four times, while 22 percent had made one online purchase. Significantly, 19 percent had made over 10 online transactions. Due to continued strong growth in the number of Internet users, it seems likely the total number of people shopping online and the proportion of frequent shoppers will continue to increase. Internet-related fraud is certainly a matter for concern, and authorities around the world are active in combating a range of illegal activities being conducted over the Internet. One of the most widely publicized cases of credit card fraud occurred in the US, where a group of web-based companies made small, recurrent charges to hundreds of thousands of credit cards [2, 3].

Improvements in security and encryption technology can make it more difficult for criminals to intercept online transactions. Both Netscape Navigator and Microsoft Internet Explorer use Secure Sockets Layer (SSL) to encrypt data before sending it over the Internet. SSL scrambles personal data and provides an unbroken key or lock that appears in the bottom of the browser window. This technology provides a secure connection that keeps data private during transmission over the Internet. However it does not authenticate the parties at either end of the transaction.

VISA International and MasterCard International, with support from many of the world's top financial institutions, are presently working to develop a more advanced encryption process called Secure Electronic Transaction (SET). SET involves a system of digital certificates provided by card issuers, and encryption. It enables the identity of both merchant and cardholder to be authenticated, and also ensures that neither the merchant or cardholder's bank sees the purchaser's credit card number [5].

The number of accounts with balance, the number of cards reported lost or stolen and the numbers of cards fraudulently used increased from 1985 to 2001. The number of cards fraudulently used in 1985 is 21,026 cards, while it is 116,139 in 2001 [4].

Enhancement of authentication protocols and encryption technology will improve the security and makes it more difficult for hackers to intercept or tamper with online transactions. For this purpose, we propose the use of smart card and secure coprocessor to provide a highly secure environment for the Internet transactions. Two cases of authentication are proposed in this paper, which can provide a secure environment for handshaking and secure transaction processing.

This paper surveys most popular threat models for smart cards, looks at several proposed solutions and describes how the Smart Card may be integrated into a secure communications environment to augment the security level of some existing protocols.

2.0 TAMPERING TECHNIQUES

There are several tampering techniques, described as follows [6]:

- Micro probing techniques, which can be used to access the chip surface directly, thus observation, mani pulation, and interfering with the integrated circuit are possible.
- Software attacks, which use the normal communication interface of the processor and exploit security vulnerabilities found in the protocols, cryptographic algorithms, or their implementation.
- Eavesdropping techniques, which monitor, with high time resolution, the analog characteristics of all supply and interface connections and any other electromagnetic radiation produced by the processor during normal operation.
- Fault generation techniques use abnormal environmental conditions to generate malfunctions in the processor that provide additional access.

Smart Card System

There are many parties potentially involved in any smart card-based system [7]. Usually, there are at least five or six, including the cardholder, the terminal, the data owner, the card issuer, the card manufacturer, and the software manufacturer.

- The cardholder is the party who possess the smart card and decides when and where to use.
- The data owner is the party who has control of the data within the card. In cases such as using a card as a mechanism for carrying digital certificates, the card owner is also the data owner.
- The terminal is the device that offers the smart card its interactions with the worlds

- The card issuer is the party who issued the smart card. This party controls the operating system running on the smart card, and any data that is initially stored on the smart card.

Smart Card Threat Models

An attack is defined as attempt by one or more parties involved in a smart card transaction to cheat either the system or the other party [7].

- **Attacks by the Terminal Against the Cardholder or Data Owner.** These are the easiest attacks to understand. When a cardholder puts his card into a terminal, he trusts the terminal to relay any input and output from the card accurately [7].
- **Attacks by the Cardholder Against the Terminal.** These involve fake or modified cards running rogue software, with the intent of subverting the protocol between the card and the terminal.
- **Attacks by the Cardholder Against the Data Owner.** In many smart card-based commerce systems, data stored on that card must be protected from the cardholder. In some cases, the cardholder is not allowed to know that data. A building access card, for example, could have a secret value inside the card; knowledge of this value could allow the cardholder to make additional access cards.
- **Attacks by the Cardholder Against the Issuer.** There are many financial attacks that appear to be targeting the issuer. In fact, the attacks are targeting the integrity and authenticity of data or programs stored on the card. These attacks are made possible by the issuer's decision to use a smart card system where the cardholder holds data for the issuer or other party.
- **Attacks by Third Parties Using Stolen Cards.** There are two differences between this attack and an attack by the cardholder. One, the thief does not have access to any secret information required to activate the card. And two, the thief has only a limited amount of time to carry out his attack before the cardholder will notice that his card has been stolen.

System Preparation

The system we are going to propose consists of a Central Repository (file server) and workstations that will be used by users, to establish a secure data communications. We must configure our Central Repository and workstations securely to protect them from malicious programs that destroy the important data used, by using only software acquired from reliable well-established vendors and testing all new software on an isolated computer to check their performance and integrity.

Central Repository Preparation

The central repository (server) must be equipped with Secure Coprocessor (SC), which creates a new security partition. Cryptographic techniques reduce some of these vulnerabilities and enhance security. For example, using secure coprocessor to boot a system through its secure kernel and ensure that the correct operating system is running provides privacy and integrity guarantees on memory not otherwise possible [8,9].

Performing the necessary integrity checks with a secure coprocessor can solve the computer integrity problem. Because of privacy and integrity guarantees on secure coprocessor memory and processing, we can have confidence in results from a secure coprocessor that checks the integrity of the central repository state at boot-up. If the secure coprocessor is first to gain control of the system when the system is started, it can decide whether to allow the computer CPU to boot after checking the disk-resident bootstrap program, operating system kernel, and all system utilities for tampering.

Several steps must be followed when Central Repository switched on for the first time during preparation:

1. Central Repository uses its secure coprocessor to boot a system through its secure kernel to ensure that the correct operating system is running to provide privacy and integrity guarantees on memory.
2. The secure coprocessor's secure kernel transfers control to central repository operating system using a bootstrap program.
3. We must load a user-build program that used to load the secret keys and secret data to secure coprocessor.
4. The user-build program (Keys Loader Program KLP) has its own serial number (S/N) or (ID), this program is very important and must be protected against (e.g. Trojan Horse) attacks, so we must know from file directory and File Allocation Table (FAT) the following information:
   - Name of the program.
   - The addresses of starting and ending sector of the program on the disk storage.
   - Size of the program in bytes.
   - Date and time of creation.
   - Attributes (read only).
5. We must know the processor (S/N) and BIOS (S/N) of our central repository.
6. Compute the Cryptographic Checksum (CC) for all above-mentioned values; Figure 1 shows Cryptographic Checksum calculation procedure. This computed Checksum must be stored in the non-volatile memory of the secure coprocessor.

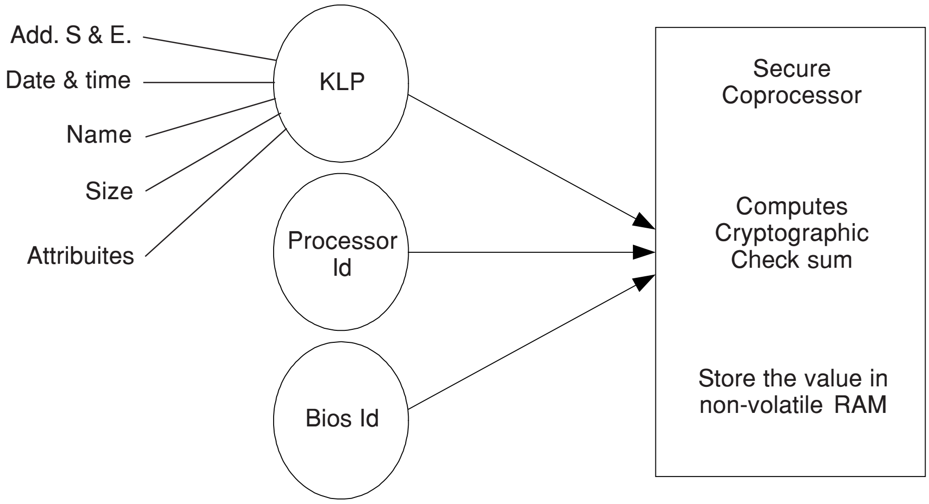


Figure 1 Loading and Computing Cryptographic Checksum

7. When we want to load any secret keys or data to the secure coprocessor, the Keys Loader Program (KLP) is loaded to secure memory of secure coprocessor. Cryptographic checksum are recomputed by secure coprocessor and compared with stored one.
8. If the computed value of Cryptographic Checksum and stored one are different, then some one has modified the (KLP) and hence, the (KLP) is terminated due to the nature of cryptographic checksum.
9. When any malicious programs such as Trojan horse modifies this (KLP), it will immediately be known to the secure coprocessor.

The above-proposed method ensures that server will use its secure coprocessor to boot a system through its secure kernel. This to ensure that the correct operating system is running, and that any loaded values to the secure coprocessor is controlled and checked by the program (KLP) and that computation of cryptographic checksum is validated.

The contents and functions of the central repository equipped by secure coprocessor after the system preparation will be discussed in more details in the next sections.

Smart Card Preparation

For each new user the Central Repository (issuer), issues a smart card that contains:

- User ID, such as (name, identity no., other information).
- Server ID, which introduces the server who issues the card.
- No secret keys are held on the card.
- Algorithms and procedures to generate keys.

Notice that, there is no secret information (key, data) held on the card. The Central Repository (CR) also issues a PIN Number (PIN #) for each user, which must be memorized by the user himself.

Workstations Preparation

Each workstation equi pped with a smart card reader, which reads the smart card information. Since no secret information held in the smart card, any unexpected attacks would not affect the process of authentication. All processing operations for generating keys that are used for user authentication process must be done inside the smart card memory (like cryptographic smart card), which increase the integrity and confidentiality of the system, such as user authentication, token authentication, DES encryption and DES key generation [5].

Central Repository Contents and Functions

Central Repository is a file server that maintains a large database. This database contains the information about users and servers, which must not be stored in plaintext to prevent any attacker from easily destroying, or modifying the information. The solution is to store the database in a hashed form to prevent the data to be available to the attackers. However, from the comparison of hashing algorithms, it is clear that the most suitable algorithm is SHA-1. With the higher processor speeds available now days, SHA-1 will acts very well. The hashed database contains records with identification of user ID, the personal identification number (PIN #), and the server ID that user belongs to, They are all in hashed form and challenge-response information about the user in plaintext form. Then the database encrypted on the hard disk of the file server.

The Central Repository (CR) equipped with a secure coprocessor (SC), which alleviates the impersonation and eavesdropping attacks, so all secret key and cryptographic checksums of algorithms and programs are stored in the nonvolatile memory of SC. Therefore any physical attempt to tamper the contents result in zeroing the content of nonvolatile memory. The SC is used to safely store the following values:

1. The key used to encrypt the hashed database.
2. The Central Repository certificate, which is generated by Certificate Authority (CA) that is encrypted by the (CA) Private key.
3. Public key of Certificate Authority (CA) and private key of Central Repository.
4. Store cryptographic checksum of encryption/decryption and hashing algorithms.
5. Compute check sum for KLP and all secret programs.

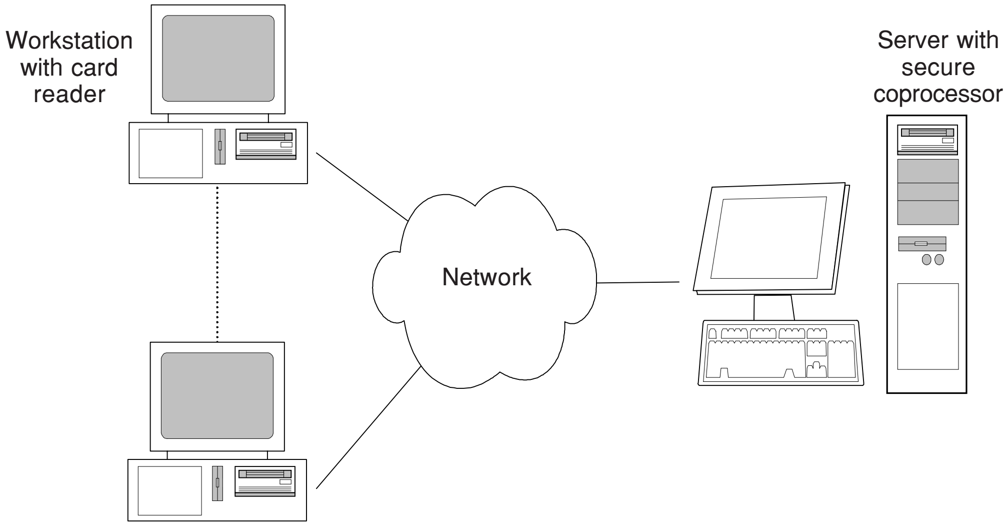


Figure 2 Server and workstations communication

Workstations Contents

They are connected to Central Repository by means of a network and communication links (see Figure 2). Each workstation equi pped by a smart card reader, which reads the smart card information.

The Central Repository Authentication

We consider two cases to authenticate a client, depending on where the user will use his smart card:

Case 1: Authentication of user to the Central Repository that issues his smart card.
Case 2: Authentication of user to the issuer Central Repository through an intermediate Central Repository (smart card that is issued by another CR trusted by the central repository where the client is now connected).

Case 1 (See appendix A for flow chart)
The user A has smart card issued by central repository A. He will use the smart card information through a workstation to authenticate the Central Repository A that issues this smart card (see Figure 3).

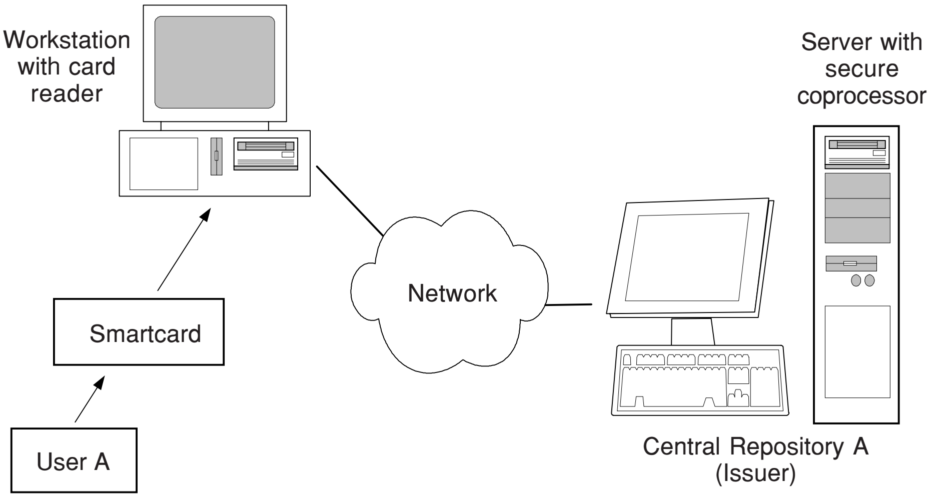

Workstation with card reader
Server with secure coprocessor
Network
Smartcard
User A
Central Repository A (Issuer)

Figure 3 User A Authenticates the Issuer Central Repository

Authentication Procedure

1. The user inserts his smart card to the smart card reader connected to the workstation.
2. The reader reads the information contained in smart card like user ID and server ID, and then sends them in plaintext through communication links to the Central Repository. There are no threats to effect the process because:

   No secret information stored on the smart card except the identification of the user and server.
   Sending this information through the channel in plaintext eavesdropper of the communication channel to obtain this information yet it causes no harm to our authentication process because this information are not secret and available for all as mentioned in previous steps.

   If eavesdropper catches this information and stores them to be used later on, it cannot be utilized because it cannot continue the rest of the steps in the process, since for each new authentication process the server will use a new challenge-response system.

3. The Central Repository checks the server ID on its hashed database (to know whether it is the issuer of this user or not, in this case it is the issuer).

4. The Central Repository hashes the user ID, and starts seeking the hashed database, until it matches the value. Then it accesses the record to get hashed PIN number:

The Central Repository chooses a random challenge from the identification and information of the user registered by the central repository during the issuance of the user 's card. The central repository then decides one question as a challenge to the user who accesses the machine connected to this central repository. This question must be encrypted by the PIN # as a key using DES encryption algorithm and then send through communication link. Questions for the challenge-response stage should be personal information only known to the user and may include among others user's date of birth, middle name, mother's name, registration date, etc.

Only the card's user and the central repository must know the response, so if the workstation catches and stores this challenge and response, it cannot use this challenge and response again because the challenge of the next session will be different.

The response or the answer to the challenge must be replied by the user himself, encrypted by PIN # and then send to the (CR) .The response will not be written into smart card, so that the attackers cannot access any information from the card, if the card is stolen.

If the user answers the challenge question correctly, then the central repository will uses the response as a part of its generated key.

There are many algorithms to generate keys[10], such as DES key generation, which is available in cryptographic smart card, and will be used in our proposed methods

The user's cryptographic smart card generates the equivalent key by combining the values of hashed PIN number, hashed ID and hashed response using DES key generation algorithm that can be run on the cryptographic smart card.

5. After the challenge is answered correctly, the Central Repository generates a key from the hashed information (hashed PIN + hashed ID + hashed Response), using DES key generation algorithm, say (K_{cr}).

6. The Central Repository encrypts a random number say (R), then send the encrypted value to workstation serving the user.

$$K_{cr}\ \{R\} \rightarrow \text{user's workstation}$$

7. At this time user must enters his PIN Number that must be hashed, then combined with the user (ID) stored in the card and the response replayed by the user, all in hashed form, to generate a key (K_{usr}), which must be equals to the key (K_{cr}) generated by Central Repository.

$$\text{Hash } \{\text{PIN \# + user ID + response}\} \rightarrow \text{DES key generates} \rightarrow K_{usr}$$

$$K_{cr} = K_{usr}$$

8. The workstation must decrypt the value $K_{cr}\{R\}$, by using the key (K_{usr}) generated by the user, if it cannot, then the request is rejected.

$$\text{Decrypt } \{ K_{cr} \{R\} \} \rightarrow \text{ using } (K_{usr})$$

9. Suppose the decrypted value at the workstation is, say (S), and then the workstation sends the value (S+1) in plaintext to the Central Repository that is to avoid reflection attacks.

$$\text{Decrypt } \{K_{cr}\{R\}\} \rightarrow \text{ using } (K_{usr}) \rightarrow \text{ giving } (S) \rightarrow \text{ send } (S+1)$$

10. The Central Repository compares the value (R+1) to the value (S+1), if both are equal then the Central Repository authenticates the user's workstation.

$$\text{If } (R+1) = (S+1) \text{ then authentication true}$$

11. The Central Repository generates a session key (K_{sess}), and then encrypts the session key by using (K_{cr}=K_{usr}) and sends back to the user's workstation. This session key will be used to encrypt and decrypt all communications between the server and the client.

$$\text{Encrypts } (K_{sess}) \rightarrow Kcr\{K_{sess}\} \rightarrow \text{ send to user's workstation}$$

12. The user's workstation (user B) decrypts the value to get the session key (K_{sess}), and then use it to encrypt the information to be sent.

All the above steps, from (1 to 12) are summarized in Figure 4.

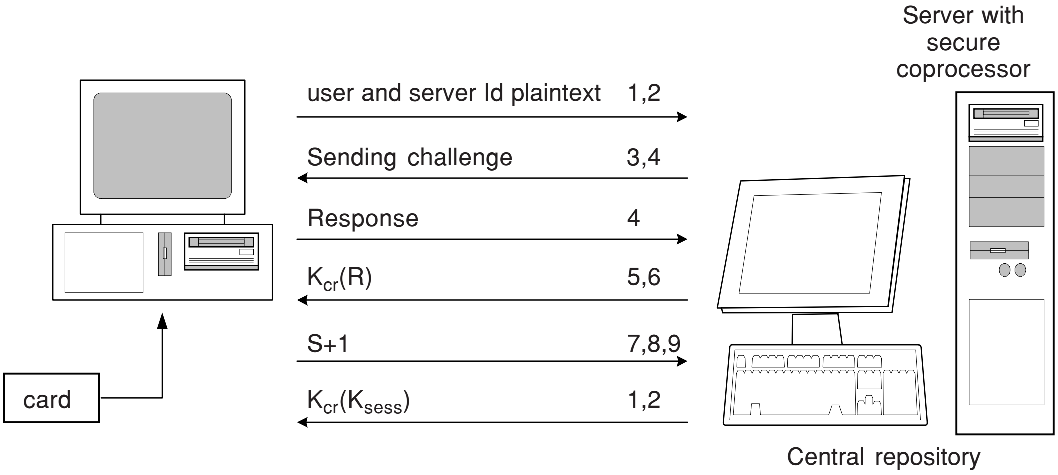


Figure 4 Steps of Authentication of Case 1

Case 2 (See Appendix B for the flow chart)
In this case we consider two Central Repositories A and B, which are connected with each other by communication links. The user who holds a smart card issued by Central Repository (B), wants to use that card in a machine connected with Central Repository (A). The steps of the procedure summarized in the next section.

Authentication Procedure

1. When the user who has a card issued by Central Repository (B) inserts his card to a machine connected to Central Repository (A), user & Server ID, contained in the card are sent to Central Repository (A) in plaintext (see Figure 5).
2. Central Repository (A) must checks whether the server (ID) sent by the user (B) is in its hashed database. If it is then the card belongs to one of other trusted central repositories that are connected to this central repository (A). Otherwise the request is rejected.

3. Central Repository (A) starts to authenticate the Central Repository (B), before doing any thing to the user (B).

4. Central Repository (A) sends its own certificate, and its public key to the Central Repository (B).

5. Central Repository (B) decrypts the certificate of Central Repository (A) using public key of certificate authority (CA), and extracts the public key in the certificate. It then compares with the public key sent from Central Repository (A) in plaintext. If they matches then it goes to perform step 6 below, otherwise the request is rejected.

6. Central Repository (B) replays by sending its certificate and its public key to Central repository (A). Central Repository (A) extracts and compares the two public keys. If they matches then the mutual authentication between the two central repositories is performed correctly, otherwise the authentication process is rejected.

The above six steps provide mutual authentication between Central Repository (A) and Central Repository (B). After finishing steps 1-6, we will start the next steps required for completion of the authentication process.

7. Central Repository (A) encrypts the message sent by user (B), (e.g. I am user (B) I want to talk to Central Repository (B) + user ID & server ID), using the public key of Central Repository (B) and sends it through the communication channel

(virtual channel) decided by Central Repository (A) to connect, the Central Repository (B) and the machine belongs to Central Repository (A) where the user inserts his card (see Figure 5).

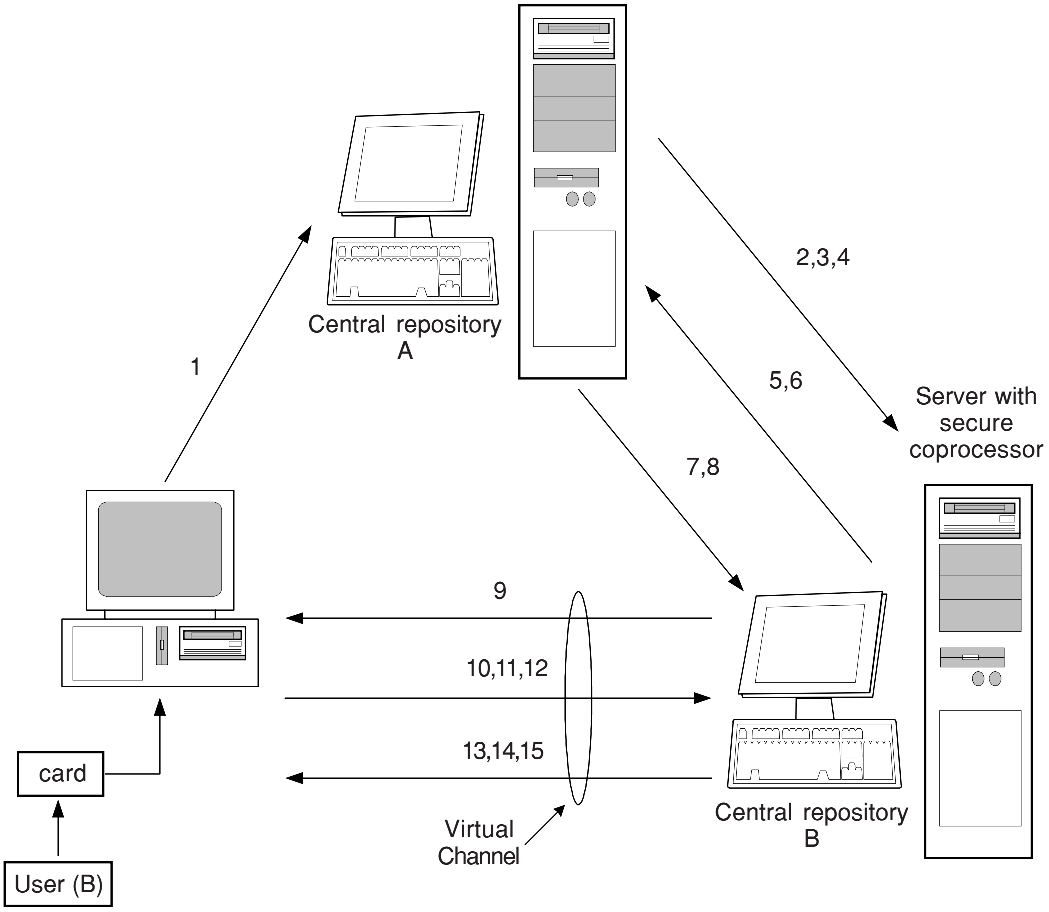


Figure 5 Authentication Using Virtual Channel

Notice here that the Central Repository (A) acts as a bridge or a router on the communication link between the user at workstation connected to Central Repository (A) and the issuer central repository (B).

8. The Central Repository (B) receives the message sent by user (B). It decrypts the message using its private key. Then it hashes the user ID, searching its hashed database and getting the hashed PIN number. Then it generates a challenge question, encrypts it and sends it to user (B). If the user (B) replied the response correctly then the Central Repository combines the response with user (ID) and (PIN) number to generate a key using DES key generation as mentioned above.

The generated key is say (K_{cb}).

To complete the procedure the rest of steps that will follow are similar to step 6 to 12 of Case 1. All steps above are summarized in Figure 5.

The Effect of Major Attacks on Case 1

Case 1 concerns the authentication of the user to the Central Repository, which issues his smart card. We will show the effects under major attacks techniques such as the following:

a. Micro Probing Techniques

These techniques can be used to access the chip surface to get secret information stored in the smart card.

In our proposed methods, even if the attacker accesses the chip surface, he could not observe or manipulate any secret values. This is because the smart card contains only user ID, which is not secret and is available to the public. Thus Micro Probing has no adverse effect on our methods.

b. Eavesdropping Techniques

It is a technique that monitor with high time resolution, the analog characteristics of all supply and interface connections and any other electromagnetic radiation produced by the processor during normal operation.

In our proposed methods, eavesdroppers couldn't get any secret information, because we don't load secret keys to workstation main memory through I/O interfaces. The generation and loading of keys is done inside the smart card.

c. Software Attacks

These attacks concerned with vulnerabilities found in the protocols and encryption algorithms. The type of smart card we use save from this attack, because we are not going to save inside the smart card any non standardized encryption, hashing algorithms that uses the PIN number, which is memorized by the user himself.

d. Fault Generation Techniques

It is a technique used to generate malfunctions in the smart card processor. These techniques have no effect on our smart card; since the information saved in the smart card are the user identification information, the standardized encryption and hashing algorithms. Therefore, no malicious programs or protocols must be inside the card. Even if there is a flaw in the crypto card's algorithm, it is not going to leak the sensitive information to the workstation.

e. Attacks by Card Issuer

The card issuer, the card manufacturer and software manufacturer are the same and must be trusted, and they have no effect to our authentication methods.

f. Attacks by the Reader Against the Cardholder or Data owner

These attacks have no effect to our proposed methods, because the reader cannot get any secret or important information from the card. Even we change the reader. The reader reads the non-secret information (ID) and returns the hashed value from the central repository.

g. Attacks by the Cardholder Against the Reader

These attacks involves, fake or modified cards running rogue software, with the intent of subverting the protocol between the card and the reader.

This has no effect on our proposed methods, because no critical or secret information stored in the card and the reader just reads the user identifications from the card, which is not secret.

h. Attacks by the Cardholder Against the Data Owner

There are two essential characteristics of these attacks. One, the card must act as a secure perimeter, preventing the cardholder from accessing the data inside the card. In this context, the card may need to be fairly confident that it will detect and respond to attacks with a minimum of control over its environment. And two, the attacker has access to the card on his own terms. He is allowed to take the card into his laboratory and perform whatever experiments he wants to. He is allowed to take cards and destroy them in order to learn how they work.

In our proposed methods the cardholder and data owner are the same, and no secret information held on the card, so no attack by cardholder against its own card.

i. Attacks by the Cardholder Against the Issuer

These attacks are made possible by the issuer's decision to use a smart card system where the cardholder holds data for the issuer or other party.

In our proposed methods, no information about the issuer would store in the card, and issuer must be a trusted party.

j. Effect of Impersonation Attacks

These attacks include the stolen cards by a third party, compromising workstations and reflection attacks.

- **Attacks by Third Parties Using Stolen Cards**
  There are two differences between this attack and an attack by the cardholder.

One, the thief does not have access to any secret information required to activate the card. And two, the thief has only a limited amount of time to carry out his attack before the cardholder will notice that his card has been stolen.

This attack has no affects in our authentication process. The thief could know the identification information of the user only, which is not secret. The cardholder is responsible for safekeeping his PIN number.

- **Compromised Workstation Attacks**

  When a Central repository issues a challenge, the user must answers the challenge question correctly. The central repository will uses the response as a part of its generated key and the user uses the information to generate the equivalent key.

  This we alleviate the workstation being compromised, because we will use only new response to be included in the generated key in each new session.

k. Reflection Attack

This attack occurs when the central repository (server) sends a challenge. The smart card's user must replies a response. This response may be caught by a reader that will try to store and use later by opening new session. This can be alleviated by the followings:

The responses used to answer the challenges must not be stored in the smart card, and must be memorized by the user himself. The challenges chosen randomly by central repository need to be answered by user correctly each time.

The workstation must decrypts the random value {R} sent by server, by using the key generated by the user. Suppose the decrypted value at the workstation is, (S), then the workstation sends the value (S+1) in plaintext to the Central Repository to avoid reflection attacks.

The Effect of Major Attacks on Case 2

In this case, we consider two Central Repositories named (A) and (B), which are connected with each other by means of communication links. The user who holds a smart card issued by Central Repository (B), and wants to use the smart card in a machine connected with Central Repository (A).

The effects under major attacks to the method of Case 2 are approximately identical to Case 1. Since we use the same authentication process sequence, the only difference is that the user authenticates the issuer Central Repository (CR) through another intermediate trusted Central Repository, which acts as a router or a bridge between the user and the issuer Central repository.

The micro probing techniques, software attacks and fault generation technique have no effects for the method of Case 2. This is because standardized encryption algorithms, protocols and hashing algorithm are saved inside the smart card. Therefore, no malicious programs or protocols are inside the card that would harm the smart card processor. Even if there is a flaw in crypto card's algorithms, it is not going to leak the sensitive information to the workstation.

The effects of the attacks by different parties of smart card system for the method of Case 2 are the same as the effects of these attacks on Case 1. The impersonation attacks include the stolen cards by a third party, compromised workstation attacks and reflection attacks. These attacks have no affects in our authentication process of Case 2. The thief knew the identification information of the user only, which is of no secret.

3.0 COMPARISON OF SET AND THE PROPOSED METHOD

Another class of electronic payment methods involves the Financial Institutions (FIs) in the protocol. The most well known example is the Visa and MasterCard joint effort, the Secure Electronic Transaction (SET) protocol [11].

1. The interaction among FIs in the settlement network is not a part of SET. Communication between FIs and consumer/merchant is defined in the protocol.
2. The authentication and non-repudiation requirements require the use of digital signatures and consequently of digital certificates for each message.
3. Privacy and integrity are also attained. Each SET cardholder must have a digital certificate issued by a trusted Certificate Authority (CA).
4. The cardholder's public key is certified via a digital certificate. This is necessary, because otherwise no one can be sure of the legitimacy of a cardholder's identity or of the public key.
5. SET provides all necessary security requirements, unfortunately by sacrificing "convenience".

Currently, SET is not widely deployed and we believe that it will not be in the near future. We also believe that the FIs are less than eager to deploy SET, for reasons mentioned below:

1. SET requires the registration of consumers by their FIs. They need to have certificates in order to use the protocol. However, SSL based solutions do not require such registration [11].
2. SET requires a PKI (Public Key Infrastructure). A PKI is a complete system for certificates. The FIs, the payment brands and the end users come together in a hierarchical manner in this PKI. Independent CAs issues the certificates. We

think that this PKI, as all other distributed and large CA-based PKIs, is unlikely to be used, because the implementation and maintenance cost of this PKI, which is to be paid to CAs, would be an extra expense for the FIs [11].

3. SET is only for payment-card (credit or debit) based transactions. Account based transactions, like electronic check (e-check), are not included in SET [11].

Comparing the proposed method of authentication in this paper to the listed disadvantages of SET mentioned above, we conclude the following:

1. Central Repository maintains a large database, which contains the information about users and servers. Therefore, every new user to the system must register in this central repository database without needs for certificates. Certificates needed only for all trusted Central repositories.
2. The proposed method of authentication does not require a Public Key Infrastructure because the implementation and maintenance cost of this PKI, which is to be paid to CAs is very expensive and increases the system cost.
3. The proposed method of authentication can be used for account-based transactions, like electronic check (e-check).
4. SET systems are not portable for the cardholder, since they require both software and certificates to be installed on one of the cardholder's local machines. This means that in order to shop using SET, the customer must have access to that particular machine. While, the proposed method of authentication provide portability for the cardholder, since the cardholder can use his smart card in any machine belongs to the system.

4.0 CONCLUSION

Authentication of user and authentication of a computer is one of the oldest research area in the world. Currently many researchers use conventional methods such as password-based authentication, address-based authentication Conventional Cryptographic and public key cryptography. A smart card-based authentication also being proposed and proved more superior than conventional approaches. In this paper, the main contribution is the attempt to investigate the use of a smart card and secure coprocessor to bring out new methods of authentication for a secure Internet transaction.

The methods of authentication developed in this paper use a cryptographic smart card that offers the medium for cryptographic operations. A secure coprocessor offers very high security that provides secure Internet transaction. Both cases considered in this paper based on storing non secret information on smart card and secret information on non volatile RAM of secure coprocessor. Keys used for authentication must be generated inside a coprocessor and the cryptographic smart card. When the user's workstation connected directly to the issuer CR, the authentication process done immediately between them. When the user's workstation connected indirectly through another trusted CR, using a virtual channel that allows a direct connection between user's workstation and the issuer central repository must do the authentication process. In this case the intermediate trusted CR acts as a router.

A series of steps must be followed to perform the authentication procedure, and each has strong effect on the security of communication between the parties of the authentication system.

The major attacks that affect the smart card and the server are applied to our proposed methods of authentication. The effects could be minimized, if we use a proper system preparation and a proper authentication process sequence as explained in previous sections.

REFERENCES

[1] David Chadwick. 1999. Smart Cards Aren't Always the Smart Choice. *IEEE Computer*. 32(12): 142-144

[2] Phantom Menace. Online Credit Card Fraud for Consumers, 2001. At: URL://www.noie.gov.au/publications/NOIE/consumer/creditcardfraud.pdf.

[3] Meridien Reserach. Statistics for General and Online Card Fraud. Statistics for General and Online Card Fraud; Date: Fri, 30 Nov 2001 10:06:28. URL://www.mailarchive.com/ansiepay@lists.commerce.net/msg00122.html.

[4] Canadian Bankers Association DB 38. PUBLIC Credit Card Statistics - VISA and MasterCard. 2001. At: URL://www.cba.ca/eng/Statistics/stats/db38_eng.PDF.

[5] Rachel Konrad and Staff Writer. Visa Program will Target Online Fraud. CNET News.com, March 24, 2000. At : URL://www.news.cnet.com/news/0-1007-200-1583717.html.

[6] Oliver Kommerling and Markus G. Kuhn. Principles for Tamper-Design Resistant Smart card Processors. *Proceedings of the USENIX Workshop on Smart Card Technology (Smart card '99)*, Chicago, Illinois, USA, May 10-11, 1999,USENIX Association, pp. 9-20, ISBN 1-880446-34-0.

[7] Schneier, B. and Adam Shostack. 1999. Breaking Up Is Hard To Do: Modeling Security Threats for Smart Cards. *USENIX Symposium on Smart Cards*, October 19.

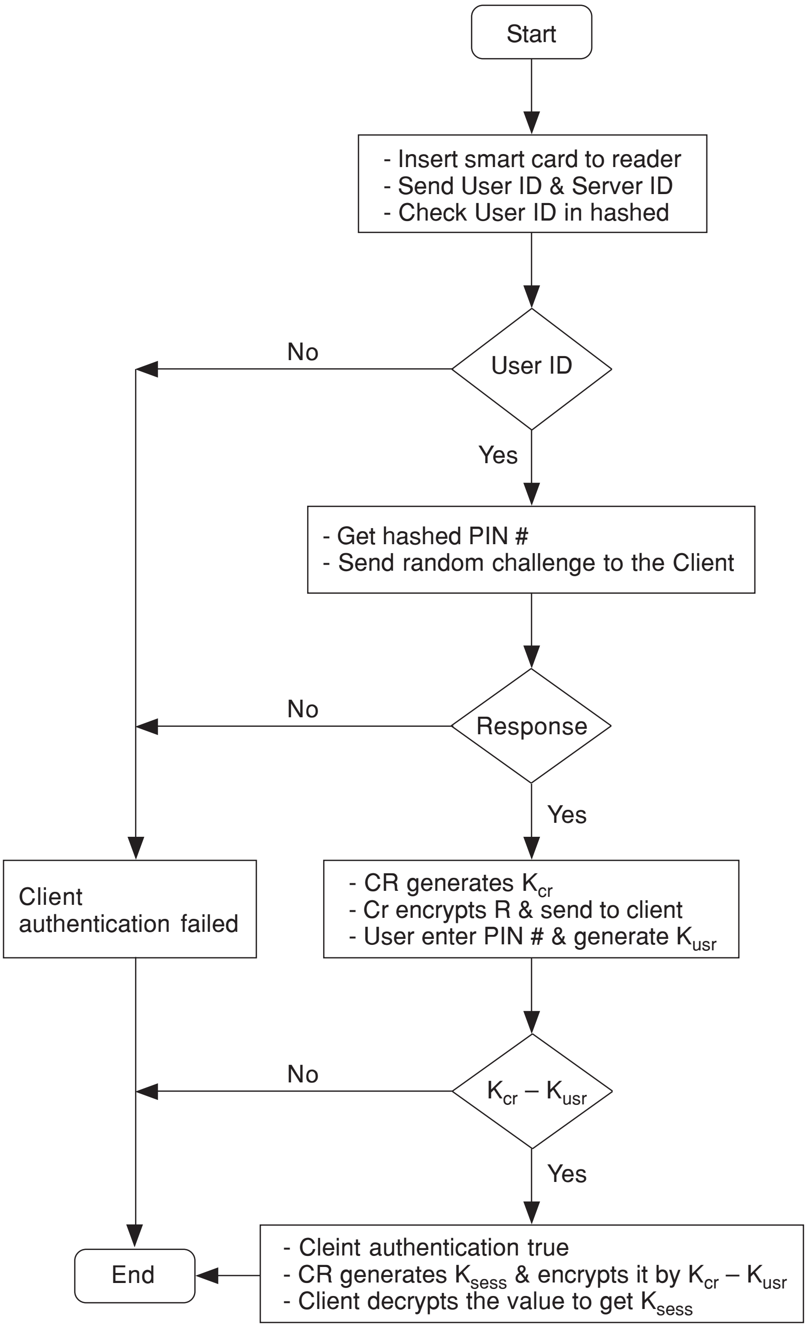
[8] B. Yee. 1994. Using Secure Coprocessor, CMU-CS-94-149 School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213. *PhD Thesis*. May.

[9] S. W. Smith, Elaine R. Palmer and Steve Weingart. 1998. Using a High-Performance, Programmable Secure Coprocessor. *Proceedings of the Second International Conference on Financial Cryptography*. Springer-Verlag Lecture Notes in Computer Science.

[10] W. Stallings. 1999. *Crytographic and Network Security: Principles and Practise.* 2nd Edition .Prentice Hall.

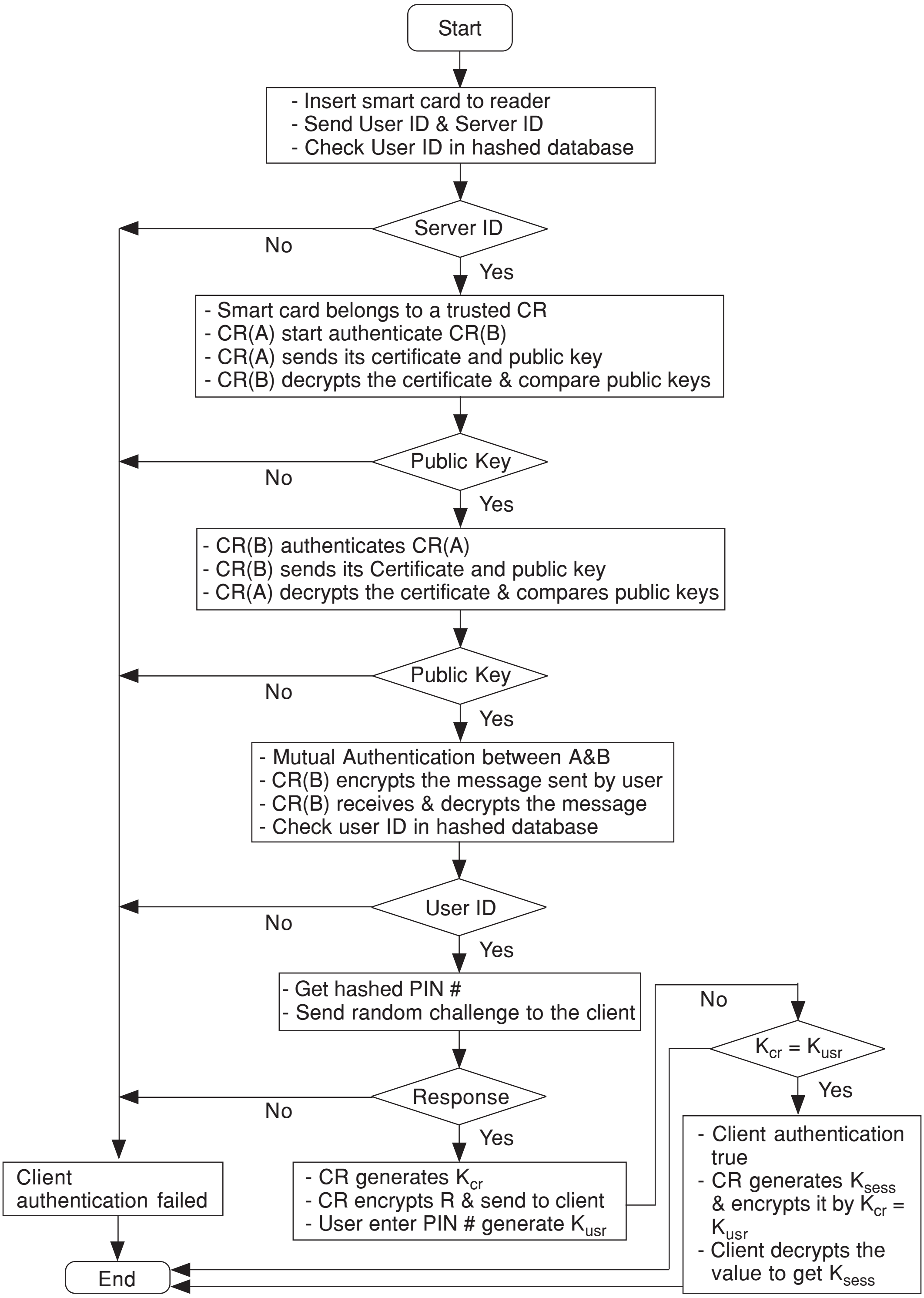
[11] Levi Albert and Cetin Kaya Koc. CONSEPP:CONvenient and Secure Electronic Payment Protocol Based on X9.59. *Proceedings, The 17th Annual Computer Security Applications Conference*, pages 286-295, New Orleans, Louisiana, IEEE Computer Society Press, Los Alamitos, California, Dec 2001.

APPENDIX A


Start
- Insert smart card to reader
- Send User ID & Server ID
- Check User ID in hashed
User ID
No
Yes
- Get hashed PIN #
- Send random challenge to the Client
Response
No
Yes
Client authentication failed
- CR generates K_{cr}
- Cr encrypts R & send to client
- User enter PIN # & generate K_{usr}
$K_{cr} – K_{usr}$
No
Yes
End
- Cleint authentication true
- CR generates K_{sess} & encrypts it by $K_{cr} – K_{usr}$
- Client decrypts the value to get K_{sess}

FLOWCHART OF CASE 1

APPENDIX B


Start
- Insert smart card to reader
- Send User ID & Server ID
- Check User ID in hashed database
Server ID
No
Yes
- Smart card belongs to a trusted CR
- CR(A) start authenticate CR(B)
- CR(A) sends its certificate and public key
- CR(B) decrypts the certificate & compare public keys
Public Key
No
Yes
- CR(B) authenticates CR(A)
- CR(B) sends its Certificate and public key
- CR(A) decrypts the certificate & compares public keys
Public Key
No
Yes
- Mutual Authentication between A&B
- CR(B) encrypts the message sent by user
- CR(B) receives & decrypts the message
- Check user ID in hashed database
User ID
No
Yes
- Get hashed PIN #
- Send random challenge to the client
No
$K_{cr} = K_{usr}$
Response
No
Yes
Yes
- Client authentication true
- CR generates K_{sess} & encrypts it by $K_{cr} = K_{usr}$
- Client decrypts the value to get K_{sess}
Client authentication failed
- CR generates K_{cr}
- CR encrypts R & send to client
- User enter PIN # generate K_{usr}
End

FLOWCHART OF CASE 2