**Journal of Information Technology Research**
Volume 15 • Issue 1

# A Robust Authentication System With Application Anonymity in Multiple Identity Smart Cards

Varun Prajapati, National Institute of Technology, Kurukshetra, India
Brij B. Gupta, National Institute of Technology, Kurukshetra, India

## ABSTRACT

User authentication plays a crucial role in smart card-based systems. Multi-application smart cards are easy to use as a single smart card supports more than one application. These cards are broadly divided into single identity cards and multi-identity cards. In this paper, the authors have tried to provide a secure multi-identity multi-application smart card authentication scheme. Security is provided to user data by using dynamic tokens as verifiers and nested cryptography. A new token is generated after every successful authentication for the next iteration. Anonymity is also provided to data servers which provides security against availability attacks. An alternate approach to store data on servers is explored, which further enhances the security of the underlying system.

**KEYWORDS**

Anonymity, Authentication, AVISPA, Multiple Identities Smart Card, Security

## 1. INTRODUCTION

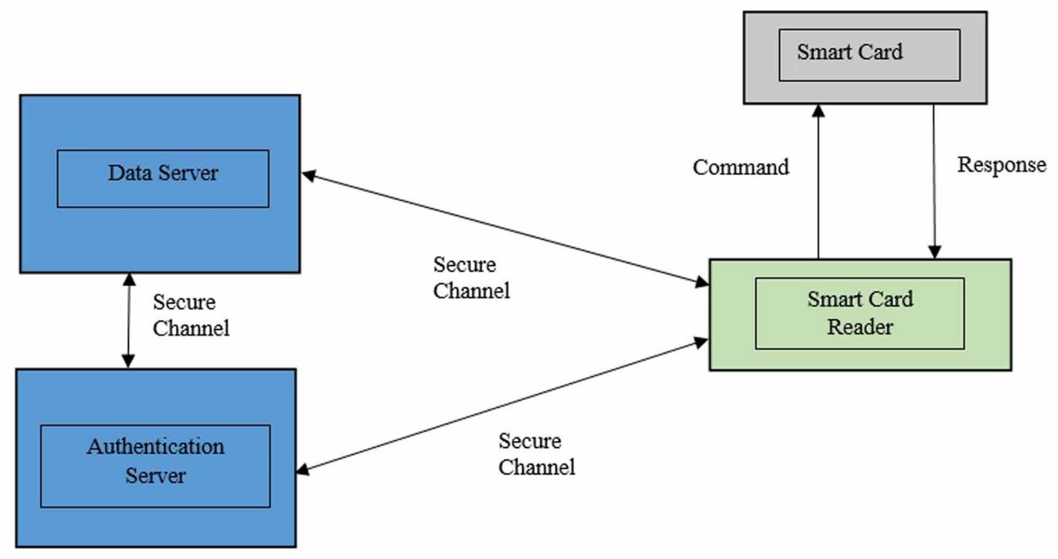
The concept of using Integrated Circuits (IC) in plastic card is very old and initial patents can be traced back to late 1960s. Technological advancement over last two decades (increase in terms of storage space, power, and processing speed along with reduction in terms of size of processor) enabled us to add further functionalities into smart card like Operating System, Authentication Mechanisms and Cryptography which led to mass implementation and usage of the system. The evolution of smart cards made them useful for wide range of applications (Rankl & Effing, 2004). Authentication is the process of verifying identity of a user (El-Latif et al., 2018; Nedjah et al., 2017; Nedjah et al., 2019; Tewari & Gupta, 2018; Zheng et al., 2017). There are 3 basic factors that can be used to authenticate users, i.e., Knowledge, Ownership and Inheritance. Knowledge consists of knowing a secret like a password, PIN, etc. Ownership consists of possessing an object like smartcard (Gupta & Quamara, 2019), software token in mobile, smart watch, etc. Inheritance consists of elements exclusive to the user like fingerprint, voice, DNA, etc.

Smart card system is one which uses smart card at its core and performs certain actions like authentication, cryptography, data storage, etc. in order to obtain the results that are desired by the user. This system consists of various entities, each assigned a specific role to carry out successful implementation of the execution. Most commonly used entities of a smart card system include Smart Card, Smart Card Reader, Server(s), and Communication Channel (Rankl & Effing, 2004).

DOI: 10.4018/JITR.2022010107

There are a few basic steps that are a part of every smart card system. Smart card is connected to the smart card reader either directly or wirelessly. This gives power and clock pulse and the smart card is activated. After activation, command and response Application Protocol Data Units (APDUs) are transmitted between card and reader. After activation of smart card, a secure channel is established between card and reader. This channel can be based either on contact point or contactless media. Usually the established channel is encrypted, especially when medium is contactless. Architecture of a basic smart card system is given in Figure 1. Next step is to validate user identity by authenticating user. User authentication can be based on one or more factors. Validating identity of user can be performed locally at reader or special server called as authentication server. After authenticating user respective application on the smart card is executed. The application can either be providing access to some service or data. Once all the operations are performed, application is terminated.

**Figure 1. Basic Smart Card System**



## 1.1 Identity Based Smart Card Classification

In this paper, we focus on multi-application smart cards. These smart cards are further classified based on total number of identities stored, i.e., Single Identity and Multiple Identity. Main differences between these cards are provided in Table 1.

Major issue lies in providing a secure authentication mechanism for multiple identities. Importance of this issue is higher because if the authentication mechanism is compromised, system will consider attacker as a legitimate user and give access of data. Resource sharing and reduced storage increases need for a lightweight and efficient authentication mechanism. Main objective and contribution of this research is to provide a platform which provides support to run multiple applications on a single smart card with minimum level of dependency or trust between applications.

The remaining section of this paper is organised as follows: Section 2 shows some related work which is relevant to our scheme. In section 3, we have described our proposed model. We evaluate security of our scheme and compare it with other schemes in Section 4. Finally, we conclude this paper in Section 5.

**Table 1. Comparison of multi-application smart cards**

| Multiple Identity | Single Identity |
|---|---|
| • Different type of identity can be used for different authentication mechanism | • Single type of identity is only stored and thus only single authentication mechanism is supported |
| • No interdependencies amongst applications as each application has a separate identity | • Applications are dependent on each other for security as compromising shared identity affects every application |
| • No mutual trust is required amongst application as neither user identity nor authentication mechanism is shared | • Mutual trust amongst the applications present on smart card is key for successful implementation as user identity is shared with all the application present on smart card |
| • High Storage space is required for storing different identities and authentication mechanisms | • Less Storage space is required as only single identity is stored and processed on smart card |
| • Security mechanisms are relatively less secure because of resource sharing | • Security mechanisms are relatively more secure because of no resource sharing |
| • Every application has its own identity and storage responsibility and thus can be less secure | • Every application works in sync and tries to secure the shared identity and authentication mechanism and thus more secure |
| • Single Identity or authentication mechanism failure leads to failure of single application | • Single Identity or authentication mechanism failure leads to failure of all applications present on system |
| • Single rogue application cannot affect performance or security of other applications | • Single rogue application affects the performance and security of other applications if core identity is accessed without processing |

## 2. RELATED WORK

Smart cards are being used to authenticate users from the time when they were not even able to update elements present in their storage. Many researchers have proposed various models so far to utilize smart cards for providing user authentication in secure manner. In this section we list some of these authentication models that use smart cards along with the list of problem that they address and solutions they provide in Table 2. Evolution in smart card technology have made it possible to use more than one factor for the purpose of user authentication. This led to development of various two-factor and three-factor authentication schemes. Inheritance factor generally is considered as the most secure factor as it is something that user cannot forget and mostly cannot be forged. Major disadvantage of this factor is the fact that it is supposed to be transferred on an insecure channel (Internet) by little or no processing from a low processing device, i.e., Smart card reader. Authentication schemes consisting of Knowledge and Ownership factors are preferred in two-factor authentication schemes.

Li et al. [3] identified problem of lack of mutual authentication as a major problem in the smart card systems. They proposed a solution which required system and user to authenticate each other before providing access to the system. Hence, user can also confirm identity of the system. Li et al. [3] and Amin et al. [9] identified that there is an extra overhead in systems with global clock. The problem of global clock synchronization existed when global clock was included in system to check freshness of received message. Random nonce generation was used as a replacement mechanism for checking freshness of the message. Lack of user anonymity was also identified as major problem in various systems [6, 7, 8]. Doshi et al. [6] used the concept of bilinear mapping to maintain anonymity of user identity. Cui et al. [7] used dynamic identity generation which was limited to a specific session to maintain user anonymity in their system. Odelu et al. [8] used the concept of masking thus not revealing actual identity of user to server.

Major identified issues comprise anonymity of user identity from system [6, 7, 8], Global clock or clock synchronization [3, 9], and lack of mutual authentication [3]. These issues along with single authentication mechanism for different identities will be addressed in this paper.

Table 2. Related Work

| Author | Problems Identified | Proposed Solutions |
|---|---|---|
| 2010, Li et al. (Doshi & Patel, 2018) | ● High computational cost<br>● No mutual Authentication<br>● Need clock synchronization<br>● Repudiation | ● One-way hash function for low computational cost<br>● Nonce for key generation<br>● Non Repudiation through Biometrics |
| 2012, Savari et al. (Mayes & Markantonakis, 2008) | ● Single Application Smart cards are difficult to carry and process<br>● Limited resources leads to weakness in cryptographic system in terms of security and efficiency | ● Single smart card to support multiple applications<br>● Cryptographic systems are combined to meet security requirements of various applications |
| 2013, Li et al. (Rankl & Effing, 2004) | ● Lack of forward secrecy<br>● No Password verification in login phase | ● No disclosure of session key even after leaking of Master key<br>● Password validation in smart card and not on server |
| 2015, Amin et al. (Li et al., 2013) | ● User and Server impersonation attack<br>● Global clock for message freshness verification<br>● Dependence on security of cryptographic symmetric key algorithms | ● Secrecy of various login parameter(Di) and Di dependent dynamic keys<br>● Random nonce to replace global clock<br>● Complete dependency on hashing for authentication of user |
| 2015, Odelu et al. (Savari et al., 2012) | ● Impersonation attack<br>● Lack of user anonymity | ● Separate authentication of user and server<br>● Identity of user is not directly revealed to server |
| 2018, Cui et al. (Amin & Biswas, 2015) | ● Denial of Service attack<br>● Anonymity attack<br>● Stolen smart card | ● Local authentication for preventing DoS attack<br>● Dynamic ID for Anonymity attack<br>● Biometric data for stolen smart card attack |
| 2018, Doshi et al. [6] | ● Man-In-The-Middle (MITM) attack<br>● No user anonymity | ● Bilinear mapping is used to maintain user anonymity<br>● Parameter secrecy to withstand MITM attack |
| 2019, Lwamo et al. [10] | ● Different schemes for single-server and multi-server environments | ● Smart card and biometric data is required for secure user authentication |

## 3. PROPOSED MODEL

The system consists of Smart Card, Card Reader, Authenticating server, Data Server and a Mediator. There are multiple Data Servers along with a single Authenticating Server. Each Data Server represents a specific type of data like Academics, Permits, Ownerships, Achievements, etc., for each and every authority. Authorities are distinct group of people who are classified into different parts and each one of them is entitled to at least one type of data depending on their requirements. Each Data Server implements different type of security mechanisms to store data statically and uses same mechanism for transfer of data. There are different types of cryptographic techniques that are used here i.e., symmetric encryption and key exchange algorithm.

### 3.1 Reason for Selection of AES and DH-EKE

In the proposed scheme, AES and Diffie-Hellman Encrypted Key Exchange (DH-EKE) algorithms are adopted. In DH-EKE, at least one party encrypts the public key using a password and sends it to other party. This message is then decrypted by the second party and is used by second party to negotiate a common shared key with the first party. This is a secure version of Diffie-Hellman Key Exchange Protocol and is used to authenticate the entities involved in communication. Main advantage of EKE is that it implies that all the parties (devices) involved in communication are authenticated and genuine.

This enhanced security is achieved only due to encryption of public keys. While Diffie-Hellman was found vulnerable to attacks like Man-In-The-Middle attack, DH-EKE could withstand such attacks. This led to developments of improved variations like Password-Authenticated Key (PAK) family in IEEE P1363 (IEEE standardization project for public-key cryptography)

Block ciphers are considered as obsolete as the encryption and decryption keys are same, but they provide more efficient and reliable cryptographic standards if key is transmitted securely. AES is one of the most secure secret key block cipher and is available in different variants of keys and security rounds like 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. In our implementation, we are using AES with 128-bit keys for simplicity. As our model is proposed for multiple application smartcards, different ciphers can be implemented depending on the level of security needed.

**Table 3. Notations used in Proposed Scheme**

| Notation | Description |
|---|---|
| SC | Smart Card |
| SCI | Smart Card Issuer |
| AS | Authentication Server |
| DS | Data Server |
| A | Authority Password |
| U | User Password |
| Med | Mediator |
| $SC_{no}$ | Secret Smart Card number uniquely assigned to each smart card |
| $U_{ID}$ | Unique identification number of each user |
| L | Available free location on Data Server selected at random |
| $E_K()$, $E_K[]$ | Symmetric Key Encryption using Key K or Key from Entity K |
| $D_K()$, $D_K[]$ | Symmetric Key Decryption using Key K or Key from Entity K |
| $e_K()$, $e_K[]$ | Asymmetric Key Encryption using public key of node K |
| $d_K()$, $d_K[]$ | Asymmetric Key Decryption using private key of node K |
| \|\| | Concatenation operation |
| SK | Secret key generated for Symmetric Key cryptography operation |
| TS | Timestamp |
| $KW_{\#}$ | Keyword no # |
| $DSa_{ddr}$ | Address of Data Server given to Mediator by Authentication Server |
| $SCR_{addr}$ | Address of Smart Card Reader given to Mediator by Authentication Server |
| $Med_{addr}$ | Address of Mediator given to Data Server and Smart Card Reader by Authentication Server |
| D1 | Personnel Symmetric Key of Data Server stored at Data Server |

## 3.2 Entities Involved And Their Role

Smart card, Smart card reader, Smart card issuer, Authentication server, Data server and Mediator are key entities that are involved in our system. Our system depends on Smart card, Smart card reader and Authentication server to authenticate user. Smart card issuer is responsible to register new user.

Role of Data server is to securely store user's data on Cloud and provide it to user after authentication. Role of Mediator is to maintain anonymity between Data server and Smart card reader by acting as a data exchange point. We explore detailed specification and role for each of these entities below.

Smart Card:

Smart Card is a device which is possessed by user. It contains dynamic token required to authenticate user in the system and mechanism to access data stored on data server. Elements stored on smart card are as follows:

1. Each smart card will have a unique Identification Number which will remain static and consists of 1024 bits of data.
2. Each smart card consists a unique one-way hash function.
3. Smart Card will comprise multiple segments of data blocks. Each of these segments will be encrypted in three layers-
    a. Outer layer will be encrypted using the pin of respective authority which is concerned with that specific type of data.
    b. Intermediate layer will consist of pin of user which may or may not vary with each segment.
    c. Innermost layer of encryption will consist of a symmetric encryption using personnel key of Authenticating Server.
4. Beneath this encryption will lie beginning location of the allotted storage space on server and a unique timestamp value required for further operations.

Smart Card Reader:

Smart Card Reader is used to access elements stored on smart card and perform various operations on them. Elements of reader and its functionality is given as follows:

1. Smart Card Reader will be having following elements stored in them:
    a. Address of authenticating server
    b. A secure tamper resistant memory unit which will act as buffer. Main role of this buffer will be to store files temporarily that are sent from server along with passwords of user and Authority.
2. Reader will also provide clock pulse and power to the card which will be used for various operations.

Smart Card Issuer:

Smart Card Issuer is responsible for providing smart cards to new users in registration phase. Their responsibilities are as follows:

1. Smart card issuer will be responsible for updating entities stored in smart card.
2. They are responsible for storing of elements on smart card and are used in registration phase.

Authentication Server:

Authentication Server is a key entity of our system. It will be responsible for validating identities and providing access to the system. The responsibilities of and operations performed by Authentication Server are as follows:

1. Authenticating Server will have a unique identity and will be key point in establishing further transactions.

2. Authentication Server will implement a symmetric key cipher operation before transmitting data to the smart card.
3. Authentication server will verify identities of Authority and User.
4. The data which will be encrypted will include following elements appended in linear way.
   a. Timestamp of the encryption
   b. Location of the file after recent transaction which will be given to the Authentication Server by the respective Data Server. One-way hash ($h_1$) of timestamp will be used to generate a random string of characters of specified length which will be appended to the location. A different one-way hash ($h_2$) will be used to generate a dynamic secret key from the timestamp which will be used to encrypt above appended data using symmetric encryption system.
   c. The timestamp will be appended to this newly generated cipher text and will be used as the input and will be encrypted using personnel key of Authentication Server.
   Data Server:

Data server is responsible for storing a specific type of data, relevant to one or more type of authorities. The responsibilities of and operations performed by Data Server are as follows:

1. Data Server will be responsible for storing data which will be of a specific category.
2. All Data Servers will maintain an encrypted channel with the Authentication Server.
3. Data Server will maintain a personnel key (D1) for symmetric cipher operations on location of data.
4. After every successful transaction where data was accessed (irrespective of the fact whether data was modified or not) following steps will be performed:
   a. Data will be allocated one of the available free slots at random and will be shifted to the new location from its current location.
   b. This new location will then be encrypted using D1 and then will be sent to the Authentication Server.
   c. New location will then be mapped to the user ID value in database and old location will be made free to store new elements.
   Mediator:

Mediator is a new entity which we have introduced in the system. The main responsibilities of Mediator are as follows:

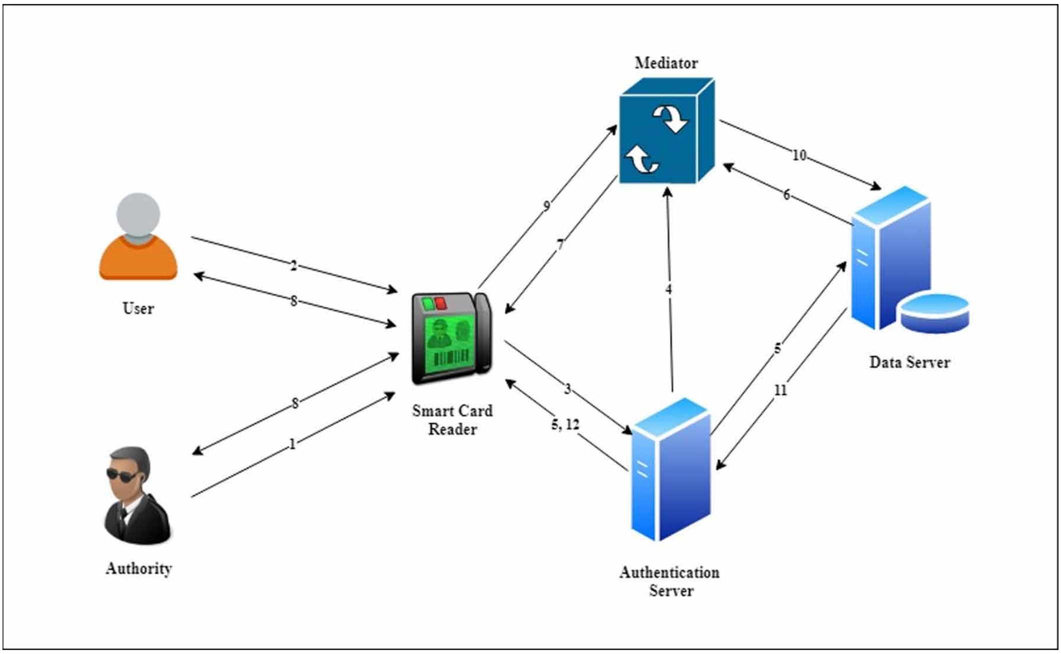
1. Mediator will be responsible for acting as a middle man to transfer data between Smart Card Reader and Data Server.
2. Mediator will be selected at random from a pool of large number of available devices by Authentication Server at random.
3. Main role of mediator is to provide mutual anonymity to Data Server and Smart Card Reader.

## 3.3 Architecture of Proposed Model

System architecture of proposed model is described in Figure 2. This figure explains the overview of model for a specific session. The flow of data and its relation with various phases are explained in the steps mentioned below:

1. In this step identity of Authority is verified when authority enters his password after SC is entered in SCR.
2. In this step identity of User is verified and SCR generates a new authentication token.

Figure 2. Architecture of proposed model



3. Message is sent from SCR to AS and final step of authentication is performed. If authentication is successful, then we move to next step otherwise the session is terminated.
4. Session Initiation Phase starts at this step. A mediator is chosen at random and identities of relevant DS and SCR are sent.
5. Identity of mediator is sent to SCR and DS in this step.
6. DS sends user data to mediator whose identity is received in previous step.
7. Mediator sends data received from DS to SCR.
8. Authority and User accesses the received data and modifies it as per the requirement.
9. Session Termination Phase is initiated and SCR sends modified data to mediator.
10. DS receives modified data from mediator and performs necessary operations.
11. DS passes message to AS which is required for future operations.
12. AS performs some computations and sends message to SCR.
13. SCR encrypts the received message with User and Authority key and returns SC to User.

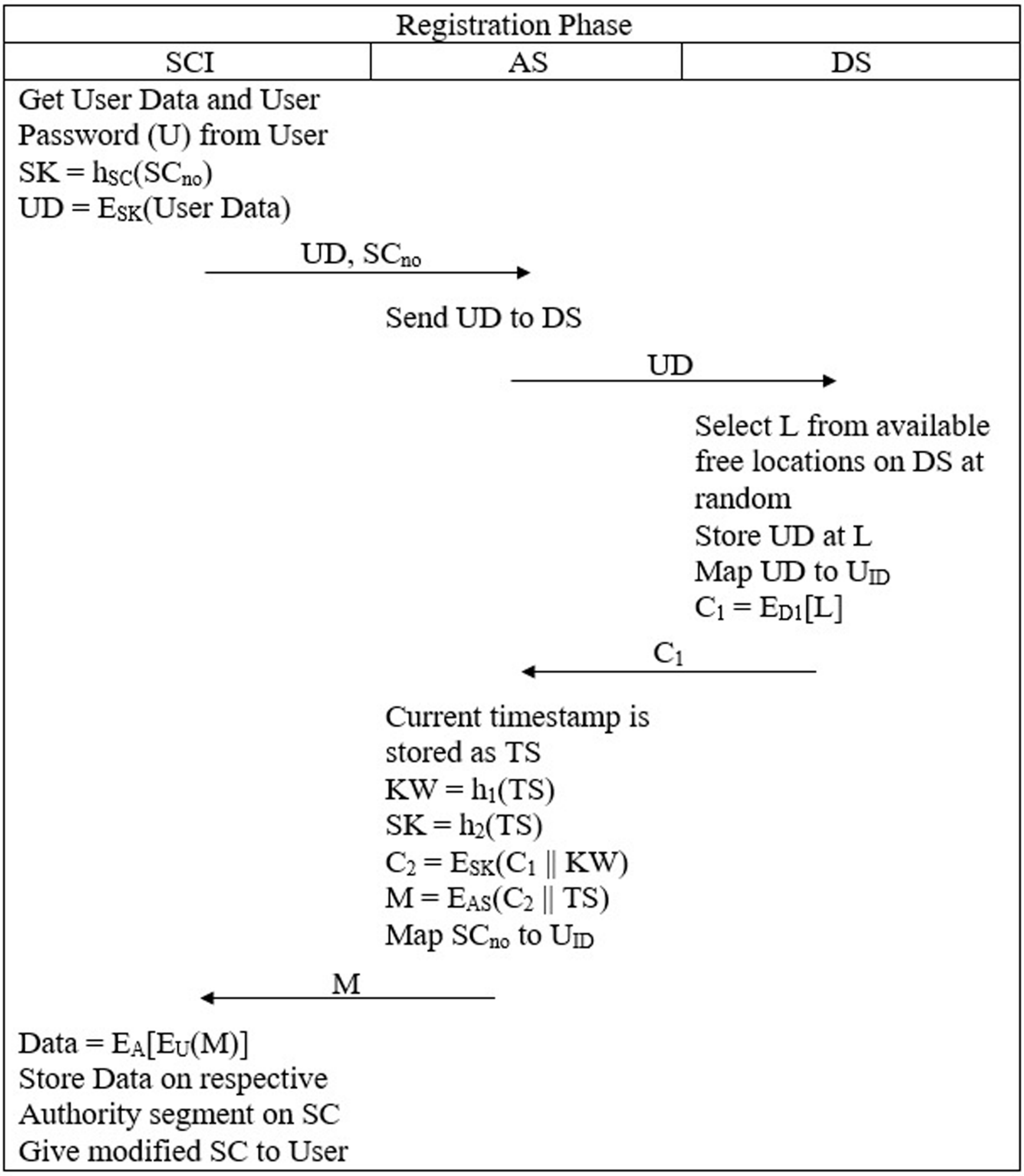
## 3.4 Various Phases of Proposed Model

In this section, flow of data is explained in detail along with the entities involved in communication within that phase. Different phases of system are as follows:

Registration Phase:

User is present physically at SCI in this phase. Entities involved in system include SCI, AS and DS. After submitting the user data to SCI, steps mentioned in Figure 3 are performed by the system. Once data is uploaded to the DS, all the required parameters are stored on SC and returned to the User. In this phase it is assumed that all the connections are secure.

**Figure 3. Flow of data in Registration Phase**



Registration Phase

SCI | AS | DS

Get User Data and User Password (U) from User
$SK = h_{SC}(SC_{no})$
$UD = E_{SK}(\text{User Data})$

$UD, SC_{no}$ →

Send UD to DS

UD →

Select L from available free locations on DS at random
Store UD at L
Map UD to $U_{ID}$
$C_1 = E_{D1}[L]$

← $C_1$

Current timestamp is stored as TS
$KW = h_1(TS)$
$SK = h_2(TS)$
$C_2 = E_{SK}(C_1 \parallel KW)$
$M = E_{AS}(C_2 \parallel TS)$
Map $SC_{no}$ to $U_{ID}$

← M

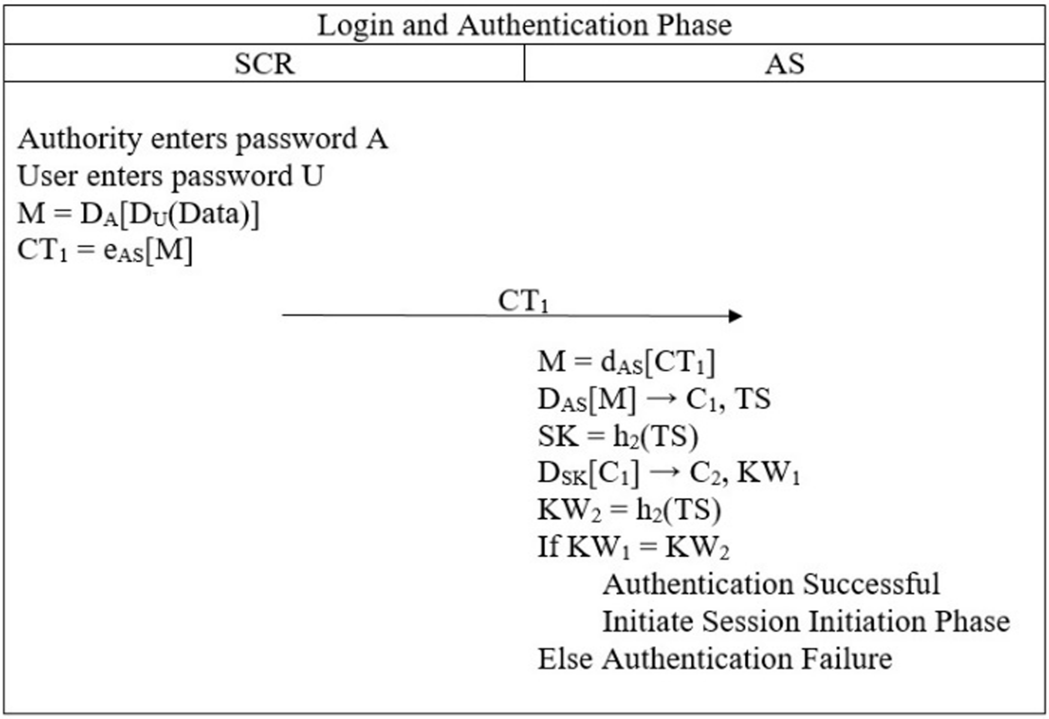
$\text{Data} = E_A[E_U(M)]$
Store Data on respective Authority segment on SC
Give modified SC to User

In case of multiple DS's, data is classified into different categories and uploaded to their respective DS's. AS gets a new $C_1$ from each DS and computes a new M accordingly for each segment on SC. All connections in this phase are considered secure because there is no power or computational constraints on any device in this phase. Secure asymmetric cryptography is performed without adding any large processing burden on any device.

Login and Authentication Phase:

In this phase, User and Authority enter their respective password at SCR. After performing preliminary operations at SCR, message $CT_1$ is sent to AS. Equal values of $KW_1$ and $KW_2$ implies that identity of User and Authority are genuine and SC is also authenticated. Steps involved in this phase are shown in Figure 4 in detail. After successful authentication, system moves to Session Initiation Phase. In case of authentication failure, failure counter is incremented at AS for specific user ID. SC is blocked when this counter reaches value 3.

Figure 4. Steps performed in Login and Authentication Phase



Login and Authentication Phase

SCR | AS

Authority enters password A
User enters password U
$M = D_A[D_U(Data)]$
$CT_1 = e_{AS}[M]$

$CT_1$ →

$M = d_{AS}[CT_1]$
$D_{AS}[M] \rightarrow C_1, TS$
$SK = h_2(TS)$
$D_{SK}[C_1] \rightarrow C_2, KW_1$
$KW_2 = h_2(TS)$
If $KW_1 = KW_2$
    Authentication Successful
    Initiate Session Initiation Phase
Else Authentication Failure

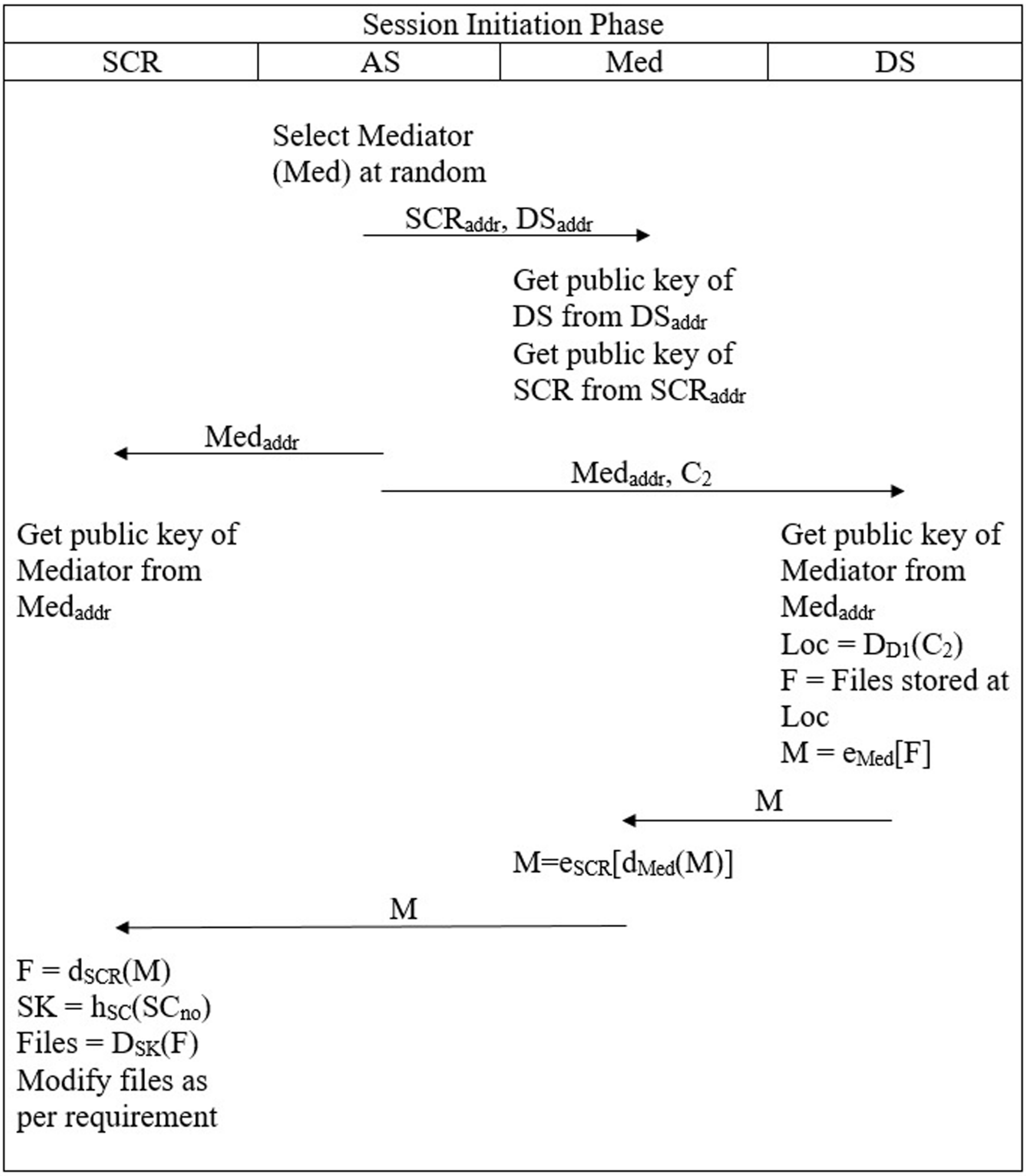
Session Initiation Phase:

In this phase, AS selects one Med from a pool of available Med at random. The address of various entities are passed by AS to other entities. This address is then used to deduct public key of other entities. For example, AS shares address of Med ($Med_{addr}$) with DS. This is used by DS to find public key of Med to communicate securely. At the end of this phase, all user files relevant to authority are displayed. These files may or may not be modified by authority. Steps involved in this phase are shown in detail in Figure 5.

Session Termination Phase:

This phase is initiated after authority has done all the desired modifications (if any) to user data. Figure 6 shows the steps involved in this phase in detail. Med waits for timeout duration of 10 minutes after which it terminates connection with DS and SCR. Med then changes its address to a

**Figure 5. Steps performed in Session Initiation Phase**

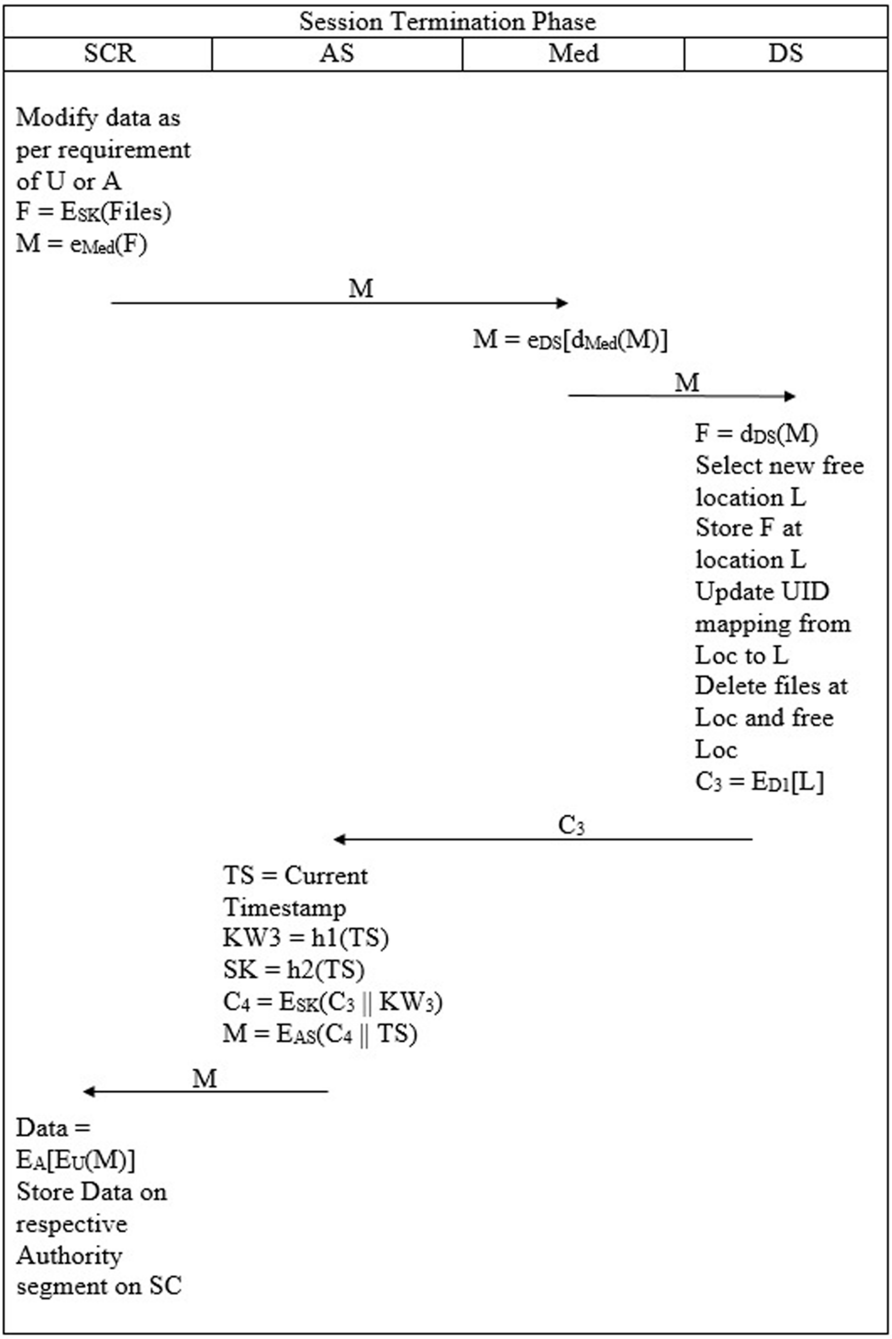

Session Initiation Phase
SCR
AS
Med
DS
Select Mediator (Med) at random
$SCR_{addr}$, $DS_{addr}$
Get public key of DS from $DS_{addr}$
Get public key of SCR from $SCR_{addr}$
$Med_{addr}$
$Med_{addr}$, $C_2$
Get public key of Mediator from $Med_{addr}$
Get public key of Mediator from $Med_{addr}$
$Loc = D_{D1}(C_2)$
F = Files stored at Loc
$M = e_{Med}[F]$
M
$M=e_{SCR}[d_{Med}(M)]$
M
$F = d_{SCR}(M)$
$SK = h_{SC}(SC_{no})$
$Files = D_{SK}(F)$
Modify files as per requirement

new random address thus making it impossible to track Med from previous connection. Med then goes back to pool of free Med and waits till it is assigned a new connection.

Password Change Phase:

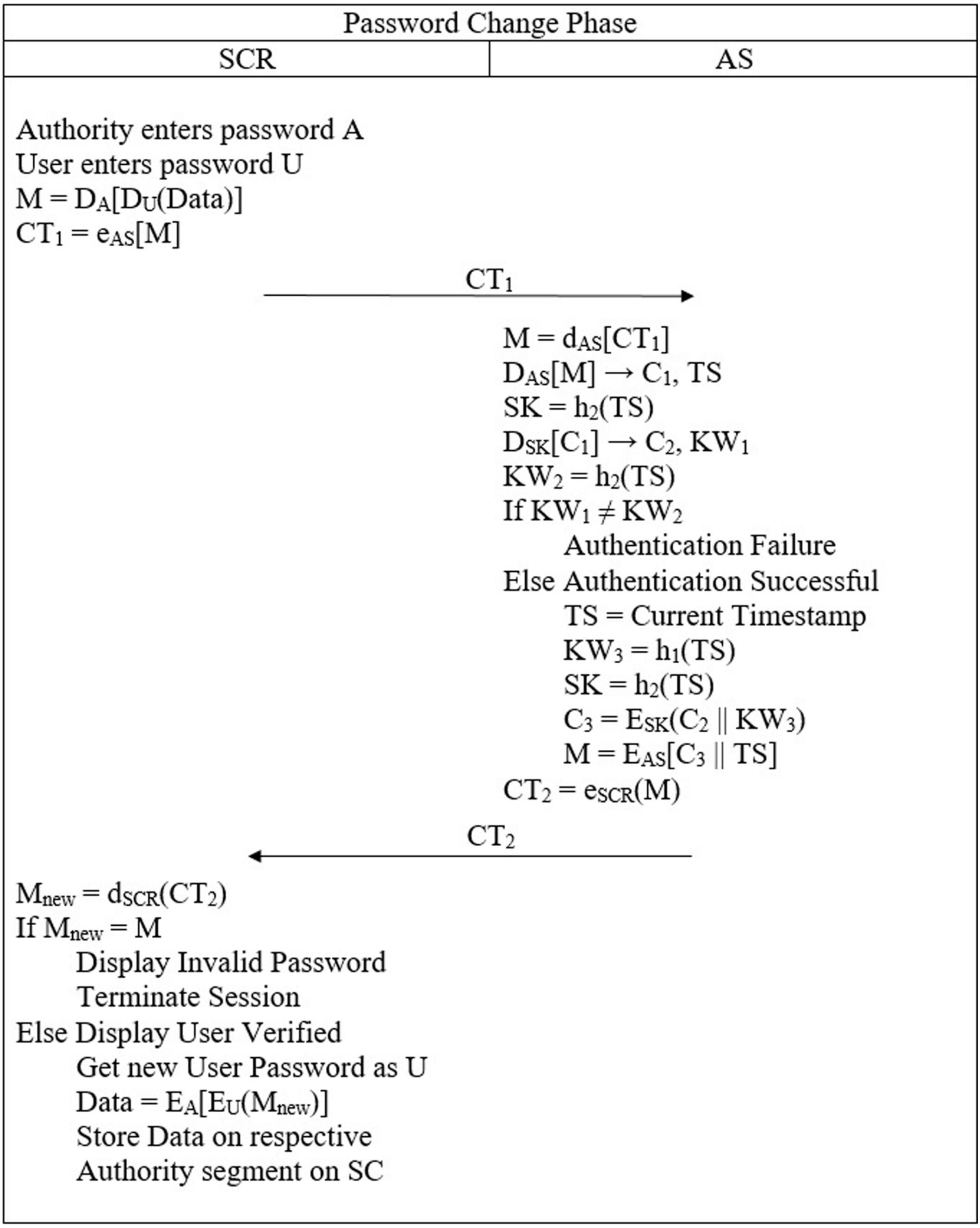
This phase is initiated as a new phase when user wants to change his password. This phase is similar to Login and Authentication Phase as before changing password, it is essential to verify identity of user and authority. This makes it impossible for authority to change user password without

**Figure 6. Steps performed in Session Termination Phase**


Session Termination Phase
SCR
AS
Med
DS
Modify data as per requirement of U or A
$F = E_{SK}(Files)$
$M = e_{Med}(F)$
M
$M = e_{DS}[d_{Med}(M)]$
M
$F = d_{DS}(M)$
Select new free location L
Store F at location L
Update UID mapping from Loc to L
Delete files at Loc and free Loc
$C_3 = E_{D1}[L]$
$C_3$
TS = Current Timestamp
$KW3 = h1(TS)$
$SK = h2(TS)$
$C_4 = E_{SK}(C_3 \| KW_3)$
$M = E_{AS}(C_4 \| TS)$
M
Data = $E_A[E_U(M)]$
Store Data on respective Authority segment on SC

presence of user and attacker to pose as user in front of authority. Steps performed in this phase are shown in detail in Figure 7.

**Figure 7. Steps performed to change password of user**



Password Change Phase

SCR | AS

```
SCR:
Authority enters password A
User enters password U
$M = D_A[D_U(Data)]$
$CT_1 = e_{AS}[M]$

        $CT_1$ →

AS:
$M = d_{AS}[CT_1]$
$D_{AS}[M] \rightarrow C_1, TS$
$SK = h_2(TS)$
$D_{SK}[C_1] \rightarrow C_2, KW_1$
$KW_2 = h_2(TS)$
If $KW_1 \neq KW_2$
        Authentication Failure
Else Authentication Successful
        $TS$ = Current Timestamp
        $KW_3 = h_1(TS)$
        $SK = h_2(TS)$
        $C_3 = E_{SK}(C_2 \| KW_3)$
        $M = E_{AS}[C_3 \| TS]$
$CT_2 = e_{SCR}(M)$

        ← $CT_2$

SCR:
$M_{new} = d_{SCR}(CT_2)$
If $M_{new} = M$
        Display Invalid Password
        Terminate Session
Else Display User Verified
        Get new User Password as U
        $Data = E_A[E_U(M_{new})]$
        Store Data on respective
        Authority segment on SC
```

## 4. EVALUATION OF PROPOSED MODEL

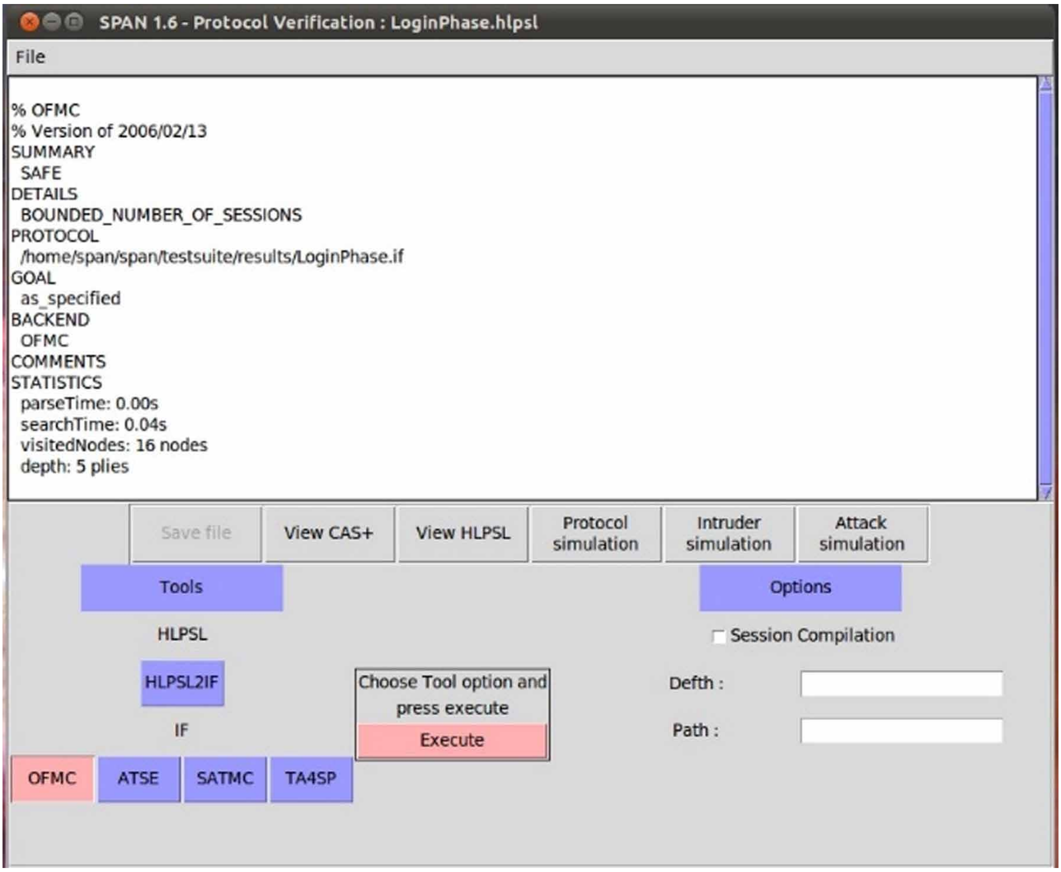
In this section, first we evaluate security of our proposed model in Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. Then we show how our proposed model provides security against various common attacks. Later we evaluate our model against different functionalities. Finally, relative comparison of model is shown against attacks and functionalities in this section.

## 4.1 Security Evaluation With AVISPA

We use AVISPA tool [12] to evaluate security of our model. It is used to implement and evaluate security of different cryptographic protocols and applications. We use OMFC (On-the-fly Model Checker) back-end model and ATSE (Constraint-logic-based Attack Searcher) back-end model to evaluate security of our protocol. The results of this evaluation are displayed in Figure 8 for OFMC back-end and Figure 9 for ATSE back-end.

We also display the simulation of our model in Figure 10. Important point to note here is that our model is secure as each smartcard has unique value assigned to itself in terms of location at DS. So a replay attack won't work.

Figure 8. Simulation under OFMC back-end

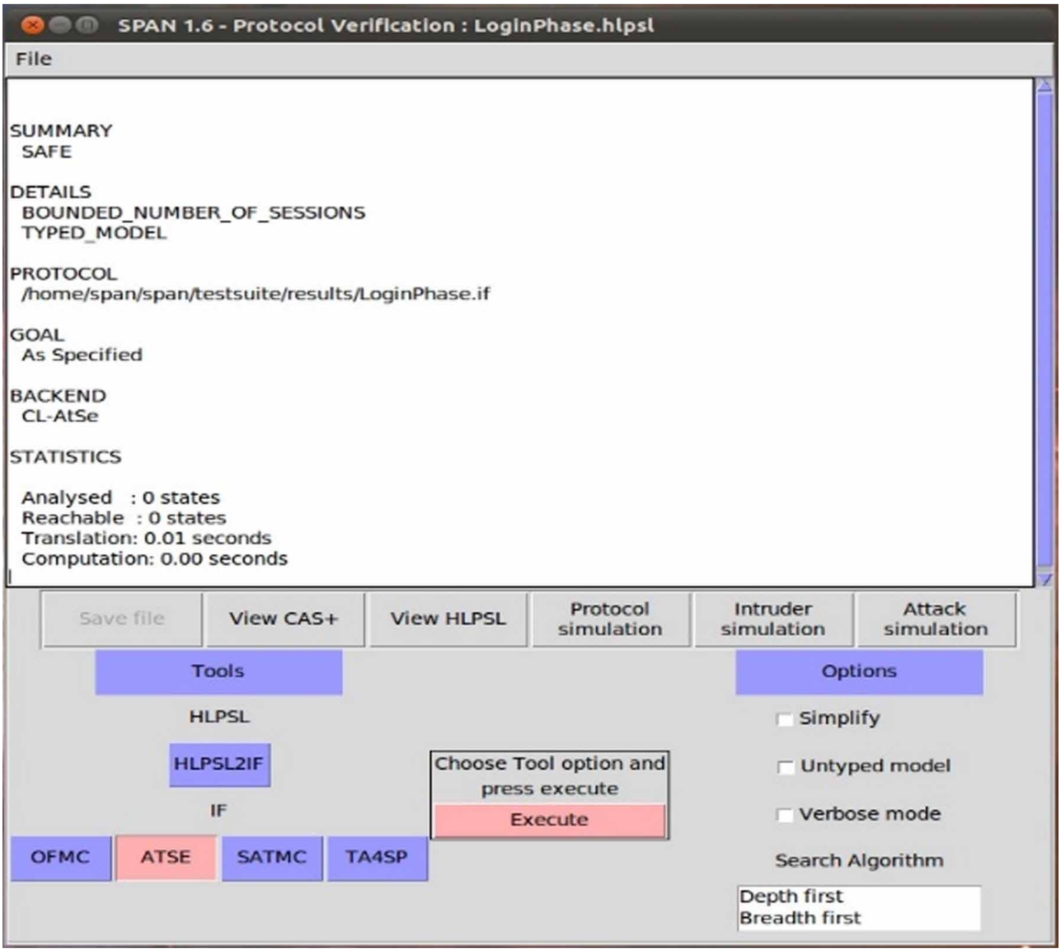


```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/LoginPhase.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.04s
  visitedNodes: 16 nodes
  depth: 5 plies
```

## 4.2 Security Evaluation Against Attacks

We consider different attacks that are possible on authentication mechanism and list out already proposed countermeasures along with countermeasures used in our system in Table 4 [11].

Figure 9. Simulation under ATSE back-end



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/LoginPhase.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.01 seconds
  Computation: 0.00 seconds
```

## 4.3 Evaluation of Functionalities

In our model, we address some other key security issues apart from attacks mentioned in Table 4 that are discussed as follows:

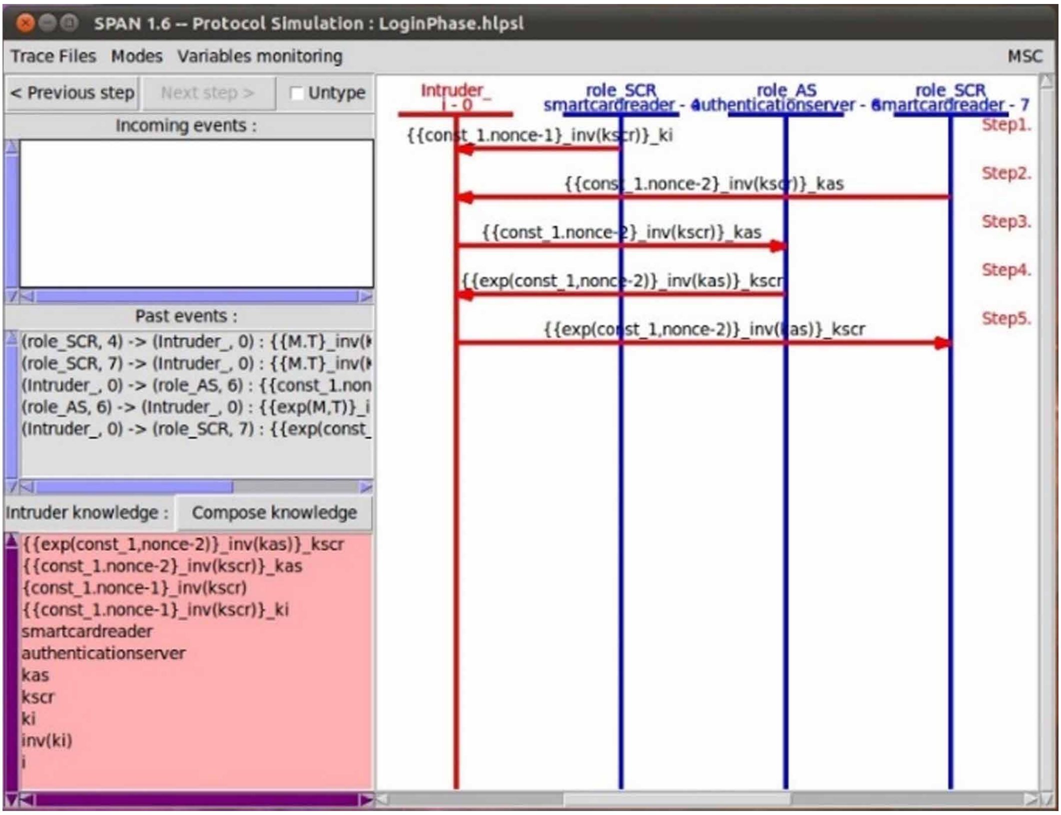
Authentication without global clock:

In our model, we generate a future token for each application during termination of current session. This token is generated by authentication server using timestamp and hash function. This timestamp keeps on changing after every successful authentication and new token contains the value of timestamp. Hence it is not necessary for authentication server to remember it. As a result, there is no need of a global clock to authenticate user.

Anonymity of user identity from system:

In our model, entire user authentication is done by the help of a token which is generated at authentication server and stored on smartcard. Data stored on data server is encrypted by a special

Figure 10. Simulation of proposed model



Step1. {{const_1.nonce-1}_inv(kscr)}_ki

Step2. {{const_1.nonce-2}_inv(kscr)}_kas

Step3. {{const_1.nonce-2}_inv(kscr)}_kas

Step4. {{exp(const_1,nonce-2)}_inv(kas)}_kscr

Step5. {{exp(const_1,nonce-2)}_inv(kas)}_kscr

```
(role_SCR, 4) -> (Intruder_, 0) : {{M.T}_inv(
(role_SCR, 7) -> (Intruder_, 0) : {{M.T}_inv(
(Intruder_, 0) -> (role_AS, 6) : {{const_1.non
(role_AS, 6) -> (Intruder_, 0) : {{exp(M,T)}_i
(Intruder_, 0) -> (role_SCR, 7) : {{exp(const_
```

```
{{exp(const_1,nonce-2)}_inv(kas)}_kscr
{{const_1.nonce-2}_inv(kscr)}_kas
{const_1.nonce-1}_inv(kscr)
{{const_1.nonce-1}_inv(kscr)}_ki
smartcardreader
authenticationserver
kas
kscr
ki
inv(ki)
i
```

key only accessible to smart card. The location at which data is stored on the server also keeps on changing continuously after every successful authentication. Hence, the user never needs to reveal his identity to the system. Data stored on smart card is also encrypted by password and hence it cannot be accessed by anyone even in case the smart card is stolen. Therefore, our system maintains anonymity of user without compromising security. This provides protection from various attacks like insider attack, privileged insider attack, and stolen verifier attack.

Lack of mutual authentication:

Mutual authentication is required so that attacker cannot disguise himself as either a legitimate user or as a legitimate system interface. Instead of providing authentication of system separately, we incorporated it in our system in authentication phase. Our system consists of nested symmetric key encryption at user, authority and server level. In order for successful authentication, all these keys must be accurate. Since all these keys are secret, only a combination of valid user and system can access the data. Thus, we provide mutual authentication in our system.

Single authentication mechanism:

In our system, we have given a general authentication mechanism where we accept input of user in the form of text. However, there are different methods like accessing biometric data, verifying geographical location, challenge response mechanisms, etc. which can be used by each application

Table 4. Security Analysis

| Attacks on Proposed Model | Proposed Countermeasures | Countermeasures in our system |
|---|---|---|
| Stolen smart card Attack | ● Biometrics<br>● OTP based authentication schemes | ● OTP based user authentication scheme |
| DNS spoofing | ● Updated software versions<br>● security patches | ● Double authentication mechanism<br>● Client side data encryption |
| Session key disclosure | ● Secure protocols<br>● Encryption technique | ● Encryption technique |
| Stolen verifier attack | ● No storage of credentials<br>● OTP based authentication | ● OTP based authentication |
| Brute force attack | ● Use of strong passwords | ● Dynamically generated passwords<br>● Nested cryptography |
| Parallel Session attack | ● PKI based authentication schemes<br>● Prevention of multiple access | ● Prevention of multiple access by blocking access of data from data server after single read |
| Plain text attack | ● Cryptography<br>● Random selection of Keys | ● Cryptography<br>● Random selection of Keys |
| Insider attack | ● Access control<br>● Hashing<br>● Encrypted data storage | ● Encrypted data storage |
| Replay attack | ● Dynamic or unique data for authentication such as OTP, Timestamp or Nonce | ● OTP generated from timestamp for authentication |
| Remote server attack | ● Encryption<br>● Access control policies | ● Encryption |
| Reflection attack | ● Challenge-Response based authentication mechanism<br>● Hash based computation<br>● Timestamp facility | ● Timestamp facility<br>● hash based keyword generation for authentication |
| Relay attack | ● Two factor authentication<br>● Distance bounding protocols | ● Two factor authentication<br>● (Authority and User password) |
| Denial of Service Attack | ● PKI<br>● Server Anonymity<br>● Large number of Servers | ● Server Anonymity |
| Masquerade attack | ● OTP<br>● Biometric based schemes | ● OTP for authentication |
| Modification Attack | ● Encryption<br>● Digital Signatures | ● Encryption |
| Man-in-the-middle attack | ● Encryption<br>● Hashing | ● Encryption |

separately so that the authentication parameters can be modified by application developers as per their need.

Anonymity of data server:

Table 5. Relative Analysis of Security Attacks

| Attacks | Doshi et al. [3] | Cui et al. [4] | Odelu et al. [5] | Amin et al. [7] | Lwamo et al. [10] | Our Scheme |
|---|---|---|---|---|---|---|
| DNS spoofing | No | Yes | Yes | Yes | Yes | Yes |
| Session key disclosure | Yes | No | Yes | Yes | Yes | Yes |
| Stolen verifier attack | Yes | Yes | Yes | No | Yes | Yes |
| Plain text attack | Yes | Yes | Yes | No | Yes | Yes |
| Reflection attack | No | No | No | No | No | Yes |
| DoS Attack | No | Yes | Yes | Yes | No | Yes |
| Masquerade attack | Yes | Yes | No | Yes | Yes | Yes |

Yes = Model can defend against this attack
No = Model is vulnerable to this attack

Attacks are performed mainly to directly affect the security of the system or otherwise weaken the performance of system. Some of them are Denial of Service (DoS) attack, Spoofing attack, etc. Their primary requirement is to know the identity of destination. In our system, we added an extra node between data server and the smart card reader labelled as mediator. This mediator provides anonymity to data server and prevents these attacks.

No verification table:

In our system, we verified users based on tokens. These tokens are generated at the end of each successful session and acts as key parameter for verification of next iteration. Hash functions are static one-way hash functions. Hence, there is no need to store them with respect to every user. Timestamps are taken dynamically and stored in tokens so no necessity to store them. As there is no information about the user which must be retained on server, verification tables are not required.

## 4.4 Relative Analysis

In Table 5, we perform relative analysis of attacks on different models from Table 2. These attacks are mentioned in Table 4. We showed how our system can withstand these attacks. In Table 6, we perform relative analysis of various functionalities listed in previous subsection for different models from Table 2.

Table 6. Relative Analysis of Functionalities

| Functionalities | Doshi et al. [3] | Cui et al. [4] | Odelu et al. [5] | Amin et al. [7] | Lwamo et al. [10] | Our Scheme |
|---|---|---|---|---|---|---|
| Server Anonymity | No | No | No | No | No | Yes |
| Global Clock Synchronization | Yes | No | No | No | No | No |
| Simple Password changing mechanism | No | Yes | Yes | Yes | Yes | Yes |
| Verification Table Required | Yes | Yes | Yes | No | No | No |

## 5. CONCLUSION AND FUTURE WORK

In recent years, smart card technology has developed quickly and is currently in use for various applications. However, there is little or no cooperation between different applications and hence, many applications require different cards for individual applications. In this paper, we proposed a new approach for combining these different cards. We provided a system where applications are not required to share any information with each other. To further enhance security, applications can add another layer of security like biometric details or personal identification number as an extra layer of security. In order to maintain anonymity of data server of different applications, communication with data server happens via a mediator. Later we analysed security of the proposed system against various attacks, along with key functionalities provided by the system and their advantages to show how we provide a better security when compared with other existing schemes.

### 5.1 Applications

This system can be incorporated by governments to digitally verify identity of citizens and to further ease the system of document verification. Also, people need not carry multiple documents with them as those documents can then be verified remotely. Another implementation can be done in the private sector like insurance where insurance can be provided instantaneously after seeing documents like health records for health insurance. Medical practitioners can be benefited from this system if all the historical data of patients like previous treatments, illness and prescriptions can be stored centrally and accessed by the doctors.

### 5.2 Limitations and Future Research Directions

Further exploration can be done to detect tampering of smart cards in an application and implement blocking in all other applications to prevent misuse of the data stored on the card. Current implementation requires smart cards to store location where data is stored on data server. However, research can be done to design a system which maps location on server based on user identity and available free locations at the data server. This will provide a failsafe mechanism for the loss of smart card.

For anonymity purposes, data server currently has no mapping of data stored to the user. However, this leads to redundant data storage as the user can come for multiple registrations. A special controller can be created so that every user is provided a secure identity based on the unique document which can be either one that the user stores, or a new identification provided to user by the system after proper verification.

Currently the scheme supports single authentication server for verifying identity of user. Further research can be done to provide support for multi-server architecture.

## REFERENCES

Amin, R., & Biswas, G. P. (2015). A novel user authentication and key agreement protocol for accessing multi-medical server usable in tmis. *Journal of Medical Systems*, *39*(3), 33. doi:10.1007/s10916-015-0217-3 PMID:25681100

Cui, J., Sui, R., Zhang, X., Li, H., & Cao, N. (2018, June). A Biometrics-Based Remote User Authentication Scheme Using Smart Cards. In *International Conference on Cloud Computing and Security* (pp. 531-542). Springer. doi:10.1007/978-3-030-00015-8_46

Doshi, N., & Patel, C. (2018, September). A Novel Approach for Biometric Based Remote User Authentication Scheme using Smart Card. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 2093-2097). IEEE. doi:10.1109/ICACCI.2018.8554825

El-Latif, A. A. A., Abd-El-Atty, B., Hossain, M. S., Rahman, M. A., Alamri, A., & Gupta, B. B. (2018). Efficient quantum information hiding for remote medical image sharing. *IEEE Access : Practical Innovations, Open Solutions*, *6*, 21075–21083. doi:10.1109/ACCESS.2018.2820603

Gupta, B. B., & Quamara, M. (2018). A taxonomy of various attacks on smart card–based applications and countermeasures. *Concurrency and Computation*, e4993. doi:10.1002/cpe.4993

Gupta, B. B., & Quamara, M. (2019). Smart Card Security: Applications, Attacks, and Countermeasures. CRC Press, Taylor & Francis.

Li, C. T., & Hwang, M. S. (2010). An efficient biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer Applications*, *33*(1), 1–5. doi:10.1016/j.jnca.2009.08.001

Li, X., Niu, J., Khan, M. K., & Liao, J. (2013). An enhanced smart card based remote user password authentication scheme. *Journal of Network and Computer Applications*, *36*(5), 1365–1371. doi:10.1016/j.jnca.2013.02.034

Lwamo, N. M., Zhu, L., Xu, C., Sharif, K., Liu, X., & Zhang, C. (2019). SUAA: A Secure User Authentication Scheme with Anonymity for the Single & Multi-server Environments. *Information Sciences*, *477*, 369–385. doi:10.1016/j.ins.2018.10.037

Mayes, K. E., & Markantonakis, K. (Eds.). (2008). Smart cards, tokens, security and applications: Vol. 2. *No. 3*. Springer.

Nedjah, N., Wyant, R. S., Mourelle, L. M., & Gupta, B. B. (2017). Efficient yet robust biometric iris matching on smart cards for data high security and privacy. *Future Generation Computer Systems*, *76*, 18–32. doi:10.1016/j.future.2017.05.008

Nedjah, N., Wyant, R. S., Mourelle, L. M., & Gupta, B. B. (2019). Efficient fingerprint matching on smart cards for high security and privacy in smart systems. *Information Sciences*, *479*, 622–639. doi:10.1016/j.ins.2017.12.038

Odelu, V., Das, A. K., & Goswami, A. (2015). A secure biometrics-based multi-server authentication protocol using smart cards. *IEEE Transactions on Information Forensics and Security*, *10*(9), 1953–1966. doi:10.1109/TIFS.2015.2439964

Rankl, W., & Effing, W. (2004). *Smart card handbook*. John Wiley & Sons.

Savari, M., Montazerolzohour, M., & Thiam, Y. E. (2012, June). Combining encryption methods in multipurpose smart card. In *Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)* (pp. 43-48). IEEE. doi:10.1109/CyberSec.2012.6246120

Tewari, A., & Gupta, B. B. (2018). A lightweight mutual authentication approach for RFID tags in IoT devices. *International Journal of Networking and Virtual Organisations*, *18*(2), 97–111. doi:10.1504/IJNVO.2018.091573

The AVISPA Project. (2019). Retrieved from http://www.avispa-project.org./

Zheng, Q., Wang, X., Khan, M. K., Zhang, W., Gupta, B. B., & Guo, W. (2017). A lightweight authenticated encryption scheme based on chaotic scml for railway cloud service. *IEEE Access : Practical Innovations, Open Solutions*, *6*, 711–722. doi:10.1109/ACCESS.2017.2775038

*Varun Prajapati received B.E. degree in Computer Engineering from TCET in 2017 and is pursuing M.Tech from NIT Kurukshetra. His research interests are in cryptography and system security. Currently he is working on Smart Card Authentication Systems.*

*B. B. Gupta received PhD degree from Indian Institute of Technology Roorkee, India in the area of information security. He has published more than 350 research papers in international journals and conferences of high repute. He has visited several countries to present his research work. His biography has published in the Marquis Who's Who in the World, 2012. At present, he is working as an Assistant Professor in the Department of Computer Engineering, National Institute of Technology Kurukshetra, India. His research interest includes information security, cyber security, cloud computing, web security, intrusion detection, computer networks and phishing.*