AN ENHANCED USER AUTHENTICATION FRAMEWORK IN CLOUD COMPUTING

Hasan Al-Refai, Khaldoun Batiha, Ahmad M. Al-Refai

Department of Computer Science, Philadelphia University, Jordan

ABSTRACT

Recently, there are several studies have proposed user authentication frameworks to defend against different types of attacks such as phishing, replay attack, man in the middle attack and denial of service attack, etc. Most of these frameworks consist of three main phases, which are the registration phase, login phase, and authentication phase. Most of them have the changing password process as an additional activity.Many problemshave been noticed in the performance of these frameworks. For example, the registration phase is valunerable to internal attack such as SYN flood attack. In this work, we aim to propose a robust user authentication framework that overcomes the previous framework shortages. The proposed framework provides many security aspects such as remote authentication, mutual authentication, session key establishment,to mention a few. Besides, to ensure the security through all phases of this framework, we add a new phase called a Service Access Authentication Phase (SAAP).This phase is resposable of the internal verification .

KEYWORDS

user authentication framework, phishing, replay attack, man in the middle attack, denial of service attack, remote authentication, mutual authentication, session key establishment.

1. INTRODUCTION

Cloud computing is strong and applicable, but it suffer from some issues, such as resiliency, performance, interoperability, the transition from legacy systems, and data migration. Security is considered one of the most issues conducted in the last few years [1], [2]. Cloud computing, as a modern technology still suffering from many security threats, where the highest impact threat is authentication breakthrough. User authentication is one of the most sensitive security issues in cloud computing environments because if the unauthorized user accesses the cloud server, he/she can cause significant damage to the data and services provided by the server. There are a lot of research works that covered the issue of user authentication by developing a user authentication frameworks [4], [5], [6], [7], [8], [9]. These solutions denote multi phases approach [3]. The first user authentication framework is proposed by Choudhury [4]. Then there are lot of enhancedversions have been eported from this approach [5], [6], [7], [8], [9]. However, all of theseframeworks suffer several shortages including:

1- Weak control over user behavior after logged to the cloud system. Thus, many attacks can occur internally (Attacks after logged to the system) such as internal DoS attack.
2- No protection technique includes the registration phase. They assume all of the client and server are honest on this phase, which reflects an unrealistic hypothesis.
3- For the registration and login phases, the service provider and user need to have a smartcard reader. They use it either to prepare or read the tcard. So for the service

provider and the user are required to have an extra deviceand this is unpopular recently [10].

In this research work, we are moving to improve the security of cloud computing, especially in the area of user authentication. There is a subsequent research work present a robust user authentication framework to defense against several types of user authentication threats in cloud computing. Their primary goal is to ensure user authentication before accessing the cloud server. The objective of this research work is to analyze the detected problem in Choudhury et al., framework [4], and the frameworks that come after it. Then, we propose a novel framework that overcomes the problems of the studied frameworks. Besides, we have given attention to the defense against the DoS attack.This is due to the fact that attack causes a lot of damage on the level of resources and the network.We mean to clear each stage in the proposed framework to ensure that the DoS attack will not occur. The rest of the paper is organized as follow: In Section 2, we introduce the previous studies that handle user authentication threats and DoS attacks. Section 3 Reflects the proposed mobile user authentication framework. In section 4, we clarify how is the proposed framework fortified against the SYN flood attack. In section 5, we perform security analysis and compare the resulted framework with the most recent previous framework. Section 6 illustrates conclusion and future work.

2. LITERATURE REVIEW

This research work mainly based on the sequence of research works [4], [5], [6], [7], [8], [9], that handle user authentication by developing a framework to protect the system from several possible attacks. The proposed frameworks have provided strong user authentication protection against several types of an attack like DoS attack, replay attack, man in the middle attack etc. In this section, we review the most related previous work and discuss their shortages.

Choundhuey et al. [4] propose the first user authentication framework for cloud network. The framework ensures user authentication before access cloud server by two-steps: verification based on password, smartcard and out of bond (i.e., strong two factors) authentication. Moreover, they provide identity management, user privacy, session key establishment, and mutual authentication.

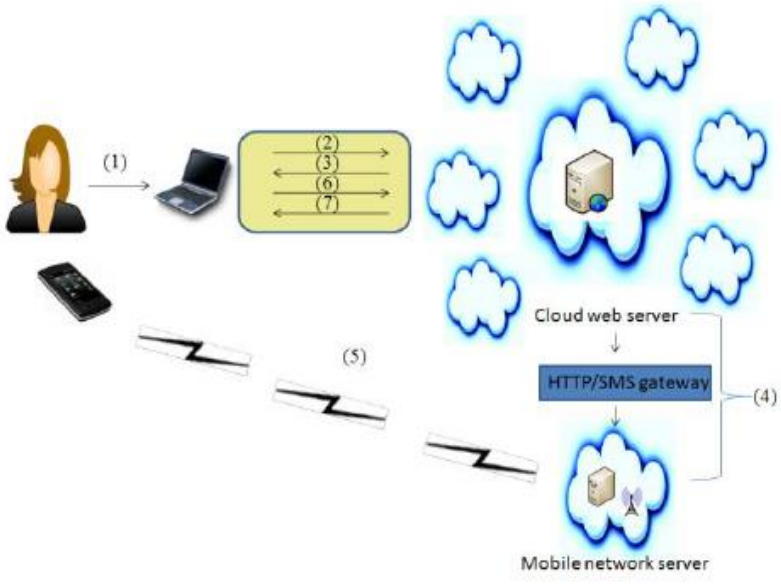

Figure 2.1: Choundhuey et al. Authentication Protocol

Figure 2.1, graphically illustrats the basic idea of Choundhuey et al. [4] framework:

1. The user logged using the smartcard and user identification (ID) and password(PW). The local system checks user legitimacy.
2. When the user is verified in the local system, the login request released to the cloud server.
3. According to receiving the login request, authentication data will be released by a cloud server based on the profile of the user.
4. The cloud server sends a onetime key verification to mobile network through HTTP/SMS gateway.
5. Via SMS, the mobile network will send that onetime key to the mobile device of the end user.
6. The user trusts and authenticates the server and sends confirmation messages according to the smartcard, ID, and onetime key.
7. The server authenticates the user based on data transmitted by the user in step 6.

There are lot of research works that came after Choundhuey et al. [4] aimed to analyze and prove the correctness framework.

Rui Jiang [6] claimed that the Chondhury framework suffers from weakness in facing some of the attacks, such as the OOB (Out Of Bond) attack, the masquerading attack, and the password change flaw, where these points appear after analyzing Chondhury framework. So he proposed an advance user authentication framework based on remote authentication schemes and apply two-factor authentication technology to overcome above security shortages.

Chen and Jiang [5] analyzed the Chondhury framework and claimed that there were some weaknesses in it. So, they made some improvements by taking into account cryptographic standards to enhance the communication security between users and the cloud server.

Chen and Jiang [7] also present an extended security analysis. Chen and Jiang enhanced the Chondhury framework. They proved the mutual authentication of their protocol formally by using space model theory and the authentication test tools, in addition to that they made performance simulation to prove their works.

Mun, Kim, and Won [9] claimed that Chondhury framework and the research works come after still suffering from weaknesses in defense against some of the attacks such as server impersonation attack, outsider attack, off-line password guessing attack and smart card stolen attack. In addition to that there was no mechanism to detect the password correctly. So, they built a secure and robust framework by using a remote user authentication scheme.

Hasan, M. et al. [14] propose a multi-technique model to ensure user authentication, while they are moving between cloud and mobile cloud computing. Where they use several authentication approaches such as a device-based approach, image and biometric-based, token-based approach, and text-based approach. Their the proposed model chooses the best approach to use according to user device capabilities.

However, Hasan, M. et al., [14] and Birkmann, J. [11] did not introduce the stage of user registration, where there is no verification on the registered users.

Tsai, J. L., & Lo, N. W [20] proposed a privacy-aware authentication scheme that ensures user authentication through a distributed mobile cloud system. The proposed scheme built under the dynamic nonce generation and bilinear pairing cryptosystem protocols. They argue

that their scheme allows users to access several mobile cloud services from several cloud providers by using one private key and a robust smart card generator. The proposed scheme supports key exchange, mutual authentication, user intractability, and user anonymity.

Huszti, A, and Ol´ah, N. [21] present an authentication scheme based on two-security factors, which are password and smartcard. They seek through using mutual authentication to protect the cloud server from the insider attack. Merkle tree used to hash user password when it was shared.

Nayak, S. K. et al. [22] propose a mutual authentication framework based on user ID, password, and user E-mail. The framework consists of three phases, which are the initialization phase, registration phase, and authentication phase. Through the authentication phase, the user sends his password and ID to the server. The server will send a token to the used E-mail and ask the user to enter the value of the token. Furthermore, they introduce flexible change of the password activity and the session agreement between user and server.

Roy, S. et al. [15] build a mobile cloud computing framework based on a lightweight remote authentication scheme. They used several techniques to ensure user authentication, such as bitwise XOR, cryptographic hash, and fuzzy extractor functions. User authentication grunted remotely before access the cloud server through authenticates the user on the level of the network operator. The request sent through the internet to the cloud service provider. They take in their account the resource-constrained on the mobile user device and design the framework to be such that a lightweight scheme.

Darwish, M. et al. [10] propose an adaption framework to defend against DoS attack. The proposed framework consists of three phases, which are the registration phase, the adaptive DoS defender protocol, and the authentication phase. Where they argue that using a complex model or scheme could exhaust the cloud's resources and can cause a weakness in defense against a DOS attack. They used a cost-based model approach to calculate each arrived request values then detect the malicious requests.

Chang, V. et al. [23] propose an adaption framework to defense against viruses and trojans threat, deal with SQL injection through intrusion prevention system. They use a multi-layered framework that consists of three layers, which are identity management layer, firewall layer, and encryption.

Al-Attab, B. S., & Fadewar, H. S. [19] proposed an authentication framework that can handle several types of attacks such as a denial of service (DOS), password guessing, replay, man-in-the-middle, insider attacks and so on. They argue that their framework built under coherent techniques such as hash function, USB token, and Diffie-Hellman. The proposed framework consists of three phases, which are the registration phase, login phase, and authentication phase, and two activities, which are USB token backup and change of password.

Fan, K. et al. [2] propose a mutual authentication scheme based on smartcard and hash function. The scheme analyzes A. Singhal & M. Ramaiva Scheme [12] and proves that their work vulnerable to offline password guessing attacks and lost smartcard attacks, so Kai, F. et al. [2] proposed an enhanced scheme to overcome these threats. The scheme consists of three phases, which are the registration phase, login phase, and mutual authentication phase.

Based on an analysis of the literature review, we found that the previous work suffers from many problems are shown in Table 2.1.

Table 2.1: Previous Work Shortages

Research work	Criteria		
	Used bio factor	**Registration phase refinement**	**Internal attack detection**
Choundhuey et al [4]	X	X	X
Jiang, R. [6]	X	✓	X
Chen, N., & Jiang, R. [7]	X	✓	X
Mun, J. et al.[9]	X	X	✓
Andrea and Norbert [13]	X	✓	✓
Hasan, M. et al., [14]	✓	X	✓
Roy, S. et al, [15]	✓	✓	X
Darwish, M. et al., [10]	X	✓	✓
Fan, K. et al., [2]	X	✓	X

3. ENHANCED USER AUTHENTICATION FRAMEWORK

In this research work, we built a robust user authentication framework for cloud networking to defend against several types of attacks over the cloud networks, in particular, the DoS attack. The framework is built based on a remote authentication technique, where the user must identify himself before access the cloud server. This is by using two factors to authenticate each user: password and fingerprint. The framework consists of four main phases and one activity: Registration Phase, Login & Authentication Phase, Service Access Authentication Phase (SAAP) and Change The Password Activity. The notations used in this research work are illustrated in (Table 3.1)

Table 3.1: Description Of Notations Used in This Thesis

Notation	**Description**
A	Denote a specific user
a	Denote basic user information
b	Denote Server authentication information
C	Denote CAPTCHA text
FP	Denote fingerprint
h(.)	Denote to the hash function
ID	Denote user identity
K	Denote onetime shared key between user and server
Enc_k	Encrypt a massege using the shared key k
LA	Denote to the local authentication process
M	Denote mobile phone
PW	Denote user password
Q	Denote QR code
R(FP)	Denote Read fingerprint
S	Denote cloud server
SA	Denote to server authentication process
SE	Denote a specific session
Si	Denote Service identifier
T	Denote terminal (desktop, laptop)
t	Denote time stamp

3.1. Registration Phase

In the registration phase, the user needs to register to the server by providing appropriate identification data such as ID, PW, and FP. The server processes this data of each user and stores it in an internal database. After that, it sends the processed data into the connected Mobile phone. The registration phase occurs through two steps. First, the user has to insert user basic information such as first name, last name, E-mail..etc., and CAPTCHA "Completely Automated Public Turing test to tell Computers and Humans Apart" recognition through his/her PC device. Second, the user will insert his fingerprint, password, and ID through his/her mobile device. Indeed the session will be transferred from the user terminal to the mobile phone to complete the registration phase securely. The server provides the terminal by QR " Quick Response " code, where the terminal asks the user to read QR code through his mobile to move the session to his mobile phone. We assumed that the user is using a mobile phone that supports a fingerprint reading.

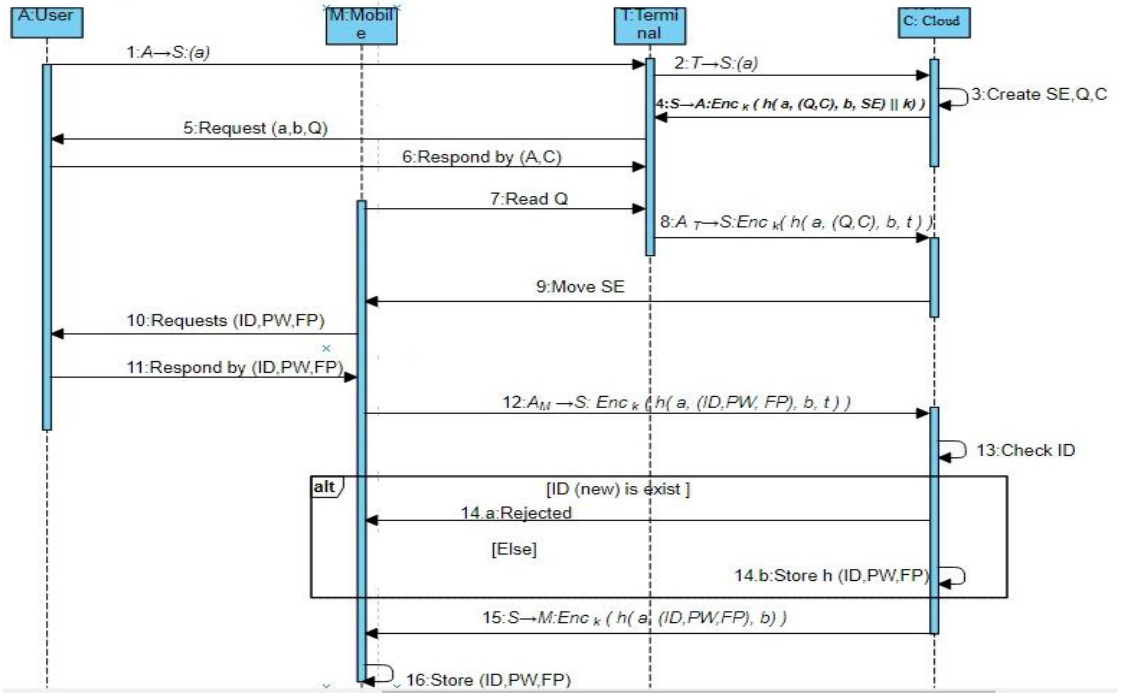

Figure 3.1: Registration Phase Sequence Diagram

The procedure of this phase is as follows (Figure 3.1) :

1- User sends a registration request through his terminal (Desktop/Laptop) with his basic information *a*.
A→S: h(a)
2- The terminal sends the request to the cloud server.
T→S: h (a)
3- The server received the request and generated onetime shared key *k*, and created *SE,Q, C* for user *A*.
4- The server sends *SE* and *h((Q,C, b) || k)* to terminal and requests *C* and *Q* from user *A*:
S→A: Enc k (h(a, (Q,C), b, SE) || k))
5- Through terminal, theuser is asked to insert *a,C* and read *Q*.
6- User inserts *a* and *C*.
7- When the user inserts *a* and *C*, the mobile phone is used to read *Q*.

8- Terminal computes *h(b, C, Q)* and sends them to server:
$A_T \rightarrow S: Enc_k (h(a, (Q,C), b, t))$
9- Server stores *a, b,* and moves *SE* from terminal to mobile phone with *k*.
10- Mobile requests *(ID,PW,FP)* from user.
11- User inserts *ID,PW,* and *FP*.
12- Mobile sends *h(ID,PW,FP)* to the server:
$A_M \rightarrow S: Enc_k (h(a, (ID,PW, FP), b, t))$
13- The server checks whither IDnew does not exist in the ID table.
14 a- if *IDnew* exists in the *ID* table, the *ID* should be rejected.
14 b- if IDnew does not exist in the ID table, the server saves h(ID,PW,FP).
15- Server save *ID,PW,FP* in the identity table and sends *h(ID,PW,FP)* to a mobile phone:
$S \rightarrow M: Enc_k (h(a, (ID,PW,FP), b))$
16- Mobile saves *h(ID,PW,FP)* locally for local authentication.

3.2. Login and Authentication Phase

This phase is invoked when the user wants to log in to the cloud server, where the users are verified before access to the cloud. The login phase accomplished through two scenarios. The first scenario presents the behavior of the user, mobile, terminal, and the server when the user accesses the cloud network through the terminal. The second scenario presents the behavior of the user, mobile, and the server when the user accesses the cloud network through his mobile. For each login session, the server generates a random number that denotes to the session key.

There are two authentication steps included in this phase. The first one is local authentication, according to registration phase user *ID, PW* and *FP* are recorded at the cloud server and sent to the mobile phone. Whenever the user wants to login into the cloud server, he must first prove his identity locally before sending the request to the cloud server. The user inserts his identity data (*ID, PW, and FP*), and the mobile phone will compare them with the recorded data that has been sent previously from the server. The second step is server authentication when then the local authentication passed, the login request can reach the cloud server, and the server reauthenticate that user through his *ID, PW, and FP*.

If the user went to login into the cloud server through a personal computer or a mobile device. We assumed that the server can distinguish the device type using a certain context-aware technique.

A- Login Through Terminal (Personal Computer)

When the user sends a login request to the server through the terminal, the mobile phone is required to authenticate himself using fingerprint. Thus, the session should be transferred to the mobile phone. The *QR* code is used as a precondition to transfer the session. The basic flow of this scenario is as follows (Figure 3.2).

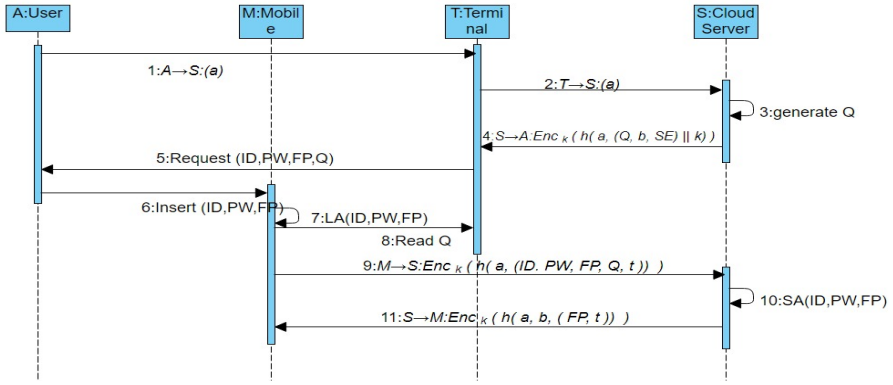

Figure 3.2: Login and Authentication Phase Terminal Scenario Sequence Diagram

1- User sends login request through his/her terminal(Desktop/Laptop) besides the authentication information *a:*
A→S :h (a)
2- The terminal sends the request to the cloud server:
T→S: h(a)
3- upon receiving the request and the server generates Q and *k* as a onetime shared key.
4- The server sends SE and h(Q) to the requested terminal.
S→A: Enc $_k$ (h(a, (Q, b, SE) || k))
5- The terminal computer, computes *Enc $_k$ (h(a, (Q, b, SE)*) and asked user *A* for ID,PW,FP, and read Q using a mobile phone camera.
6- User inserts ID,PW and FP on a mobile phone.
7- Local authentication performed based on user data stored on mobile.
8- Through mobile phone Q will be scanned from the terminal.
9- Mobile sends h(ID,PW,FP,Q) to the server:
M→S: Enc $_k$ (h(a, (ID. PW, FP, Q, t)))
10- The server authenticates the user.
11- The server responds by sending *SE* that has a user profile.
S→M: Enc $_k$ (h(a, b, (FP, t)))

B- Login Through Mobile Phone

The basic flow of this scenario is as follows (Figure 3.3).

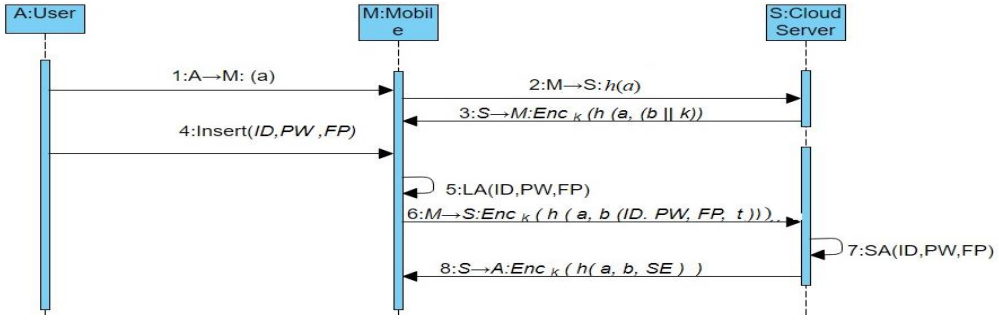

Figure 3.3: Login and Authentication Phase Mobile Scenario Sequence Diagram

1- User sends login request through Mobile including basic information *a* :
A→M: a
2- Trough mobile the request will be sent to the server *M* :
M→S :h (a)

3- Server responds by generating a onetime shared key with a server using authentication information *b*, and send it to user *M:*
$S \rightarrow M: Enc_k (h (a, (b \| k)))$
4- User inserts *ID,PW, and FP* on the mobile phone.
5- Local authentication is verified based on user data stored on mobile.
6- Mobile sends h(a, b, ID,PW,FP,Q, t) to the server:
$M \rightarrow S: Enc_k (h (a, b (ID. PW, FP, t)))$
7- The server authenticates the user.
8- Server responds by sending *SE* that have user profile:
$S \rightarrow A: Enc_k (h(a, b, SE))$

3.3. Service Access Authentication Phase (SAAP)

In this phase, in each user request, the identity is proven via fingerprint. Whereby this verification, the server is protected from any internal attack such as the SYN Flood attack. SAAP phase accomplished through two scenarios, where the first scenario presents the behavior of the user, mobile, terminal, and the server when the user accesses the cloud through the terminal. The second scenario presents the behavior of the user, mobile, and the server when the user accesses the cloud through his mobile.

A- SAAP Through Terminal

When the user requests a specific service from the server through the terminal, the authentication is required from the mobile device using the fingerprint. The session is transferred to the mobile phone to allow the user to insert the fingerprint; we used a QR code as a precondition to transfer the session.

The basic flow of this scenario is as follows (Figure 3.4).
1- From terminal user requests a service from the cloud server:
$A \rightarrow S: h (a)$
2- Server responds by generating *Q* and a fresh onetime shared key *k*, and sends it to user *A:*
$S \rightarrow A: Enc_k (h (a, (b, Q) \| k)))$
3- Through mobile user needs to read Q from the terminal.
4- Via mobile Q will be read.
5- Mobile sends *h(FP,Q)* to the server:
$M \rightarrow S: Enc_k (h (a, (b, Q, FP)))$
6- Server checks *Q,FP,* and response by the service with its *Si (service identifier)*:
$S \rightarrow T: Enc_k (h (a, (b, Si, SE)))$

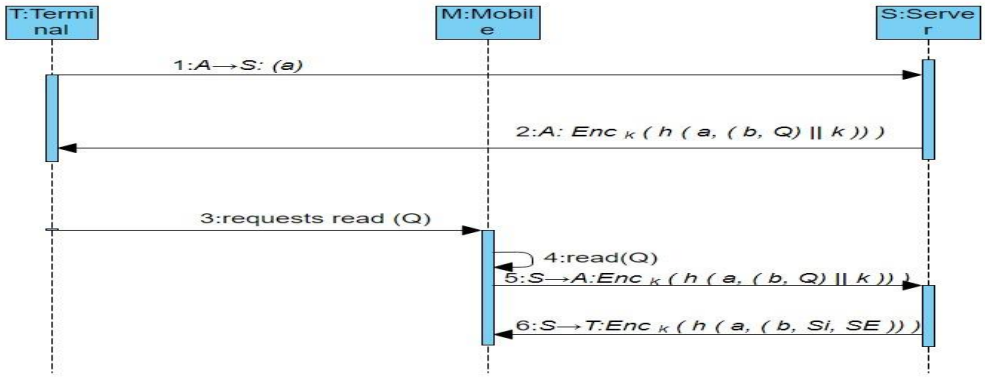

Figure 3.4: SAAP Terminal Scenario

B- SAAP Through Mobile Phone

The basic flow of this scenario is as follows (figure 4.5).

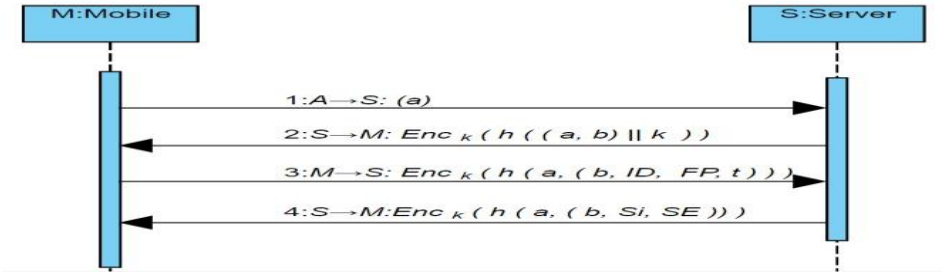

Figure 3.5: SAAP Mobile Scenario

1- From mobile, the user requests a service from the cloud server:
A→S: h(a)
2- Server responds by generating a fresh onetime shared key *k*, and requests FP from mobile*:*
S→M: Enc $_k$ (h ((a, b) || k))
3- Mobile sends *h(FP,ID)* to the server:
M→S: Enc $_k$ (h (a, (b, ID, FP, t)))
4- Server checks ID, FP, and response by the service with its *Si (service identifier)*:
S→M: Enc $_k$ (h (a, (b, Si, SE)))

3.4. Change The Password Activity

The presented framework is considered a flexible, this is due to the fact that the user can,at any time, change the password.

The presumption of this activity is:
A: User can change his password through mobile phones only.
B: User is already logged in to cloud system.
The basic flow of this activity is shown in Figure (3.6):

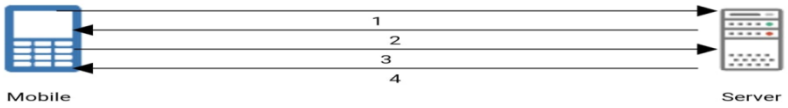

Figure 3.6: Change the Password Flow

1- User insert his/her *ID*, *PW* and *FP* in order to change his password.
2- The server checks the correctness of *ID*, *PW* and *FP* then ask the user to insert the PW_{new}.
3- The user sends the PW_{new} to the server.
4- The server saved the PW_{new} and sent it to the mobile phone for local authentication purposes.

4. THE DEFENSE AGAINST SYN FLOOD ATTACK

SYN Flood attack is a type of internal Denial of Services (DoS) attack. The attacker exploits the usage of TCP protocol, where according to TCP protocol, there are three ways handshake to guarantee reliability as shown in Figure (4.1). The server releases an SYN-ACK to the user and keeps waiting

for an ACK from the user. If the request is sent from the attacker, then no ACK will be sent back to the server, the server will keep that connection open and expect to receive messages from the user. A certain channel and resources are assigned to that user [16].

Figure 4.1: TCP Three-Way Handshake Protocol

The presence of an SYN flood attack (Figure 4.2), when the attacker sends a large number of concurrent requests; without sending ACKs to the server. Then the buffer will be flooded by fake requests. If a good user sends a request to the server, the service will be denied [17].

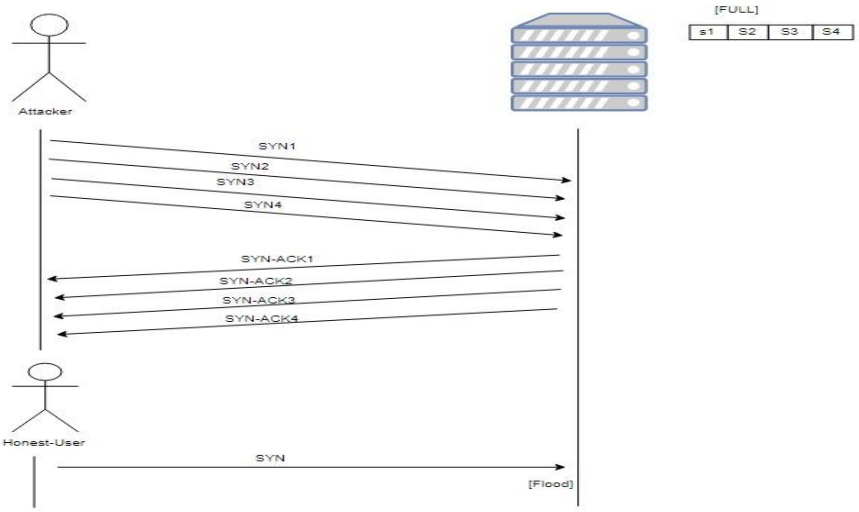

Figure 4.2: SYN Flood Attack

In our proposed framework, the SYN Flood attack has been tackled. From SAAP (Figure 4.3) the request is not accepted or inserted into the buffer until the user sends his fingerprint with each request. However, the attacker cannot send any malicious request because the fingerprint cannot be generated [18].

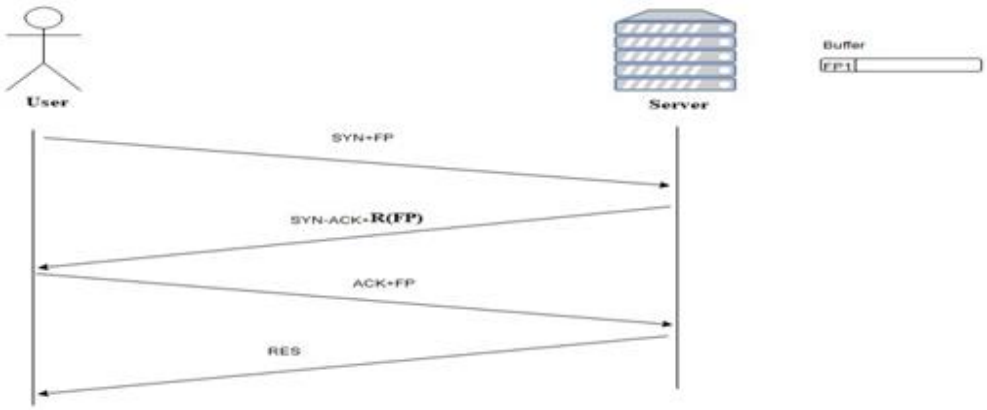

Figure 4.3: SAAP Against SYN Flood Attack

Firstly, the user requests a specific service from the cloud server, this requests inserting fingerprint to authenticate the user. The server responds by inserting the requested ID and

user fingerprint in the buffer and send SYN_ACK + R(FP), then the user sends ACK and waiting for server response.

5. EVALUATION AND SECURITY ANALYSIS

In this section, we present an evaluation of the presented framework. According to [14], there is no clear standard or technique to test the authentication property under the concept of a framework. We use the same evaluation technique that has been used in [19], where they used a security analysis that contains two-steps. The first one is functionality requirements, which define the main function or property that used to defend against several types of attacks. The second one is security requirements that determine the attacks that the framework defense against it. Then a comparison table will show how the presented framework is more robust and efficient compared to previous frameworks.

5.1. Security Analyses

In this section, the analyses will be divided into functional requirements and security requirements.

5.1.1. Functionality Requirements

F1: Mutual Authentication

It is a technique used for both sides (user and server) in a specific communication channel to authenticate each other. The user authenticates the server and vice-versa. In the presented framework, the user verifies him/herself through a two-step, which is the local authentication and server authentication. The server authenticates itself to the user by using server authentication information (*b)*. In all phases, the user and the server authenticate each other by checking the equality of the user fingerprint *FP* and the server authentication information *b* at each side.

F2: Identity Management

When the user inserts the ID on the registration phase step 11, the server will check whether the ID new is unique (registration phase step 13) through the ID management table, where the table contains all registered IDs.

F3: User Privacy

According to this work, each message transmitted under the cryptographic mechanism by encrypting the messages between parties by the onetime shared key for each opened session, where it is hard to be decoded. Hence, the scheme provides user privacy.

F4: Session Key Agreement

For each session, the server generates onetime shared key *k* between the user and the server. This key is concatenated with a specific session token established to prevent reuse that key. Therefore, in each login session, there is a session token generation. The generated token is established between the user and the server after the authentication process is finished. When the session is expired, the token will not be repeated.

F5: Password Change

The presented framework includes password change activity, where on-demand users can change the password, as shown in Section four, section 4.2.4.

F6: Portability

This research work presents an authentication framework that able to be applied to cloud computing and mobile cloud computing everywhere at any time.

F7: Authenticate the User at the Registration phase

In this work, we provide several techniques to immunize the registration phase, such as CAPTCHA, onetime shared key establishment, session token, hashing, and QR code.

5.1.2. Security Requirements

R1: Replay attack

A replay attack occurs when the attacker eavesdrops on two-points of communication in order to repeat the request when the session is ended. The presented framework has a session key establishment where the session key is random and will never repeat. In addition, to the use of a timestamp technique.

R2: Password Guessing Attack

The framework stores the password in encrypted form in addition to use a one-way hash function (h(.)). The framework relies on two factors to determine user authentication, which is password and fingerprint. If the password has been disclosed, the fingerprint prevents the intruder from accessing the server.

R3: External DoS

The framework guaranteed user authentication before access the cloud server in the login phase (chapter four, section 4.2.2, (A) step 7). Thus there is no way to damage the server by external DOS.

R4: Internal DoS

After the authentication phase, the Service Access Authentication Phase (SAAP) is started, in this phase, the server asked the user to prove himself/herself by sending his fingerprint with each request, he/she sends it to the server. Thus there is no way to send malicious requests to the server.

R5: Man in the Middle Attack

If the attacker has the message sent from the user and the server or vice versa, the malicious attempt is not successful. This is due to the fact that the attacker cannot modify the messages since it was encrypted in addition to used timestamp and fingerprint mechanisms.

R6: Insider Attack

An insider attack is considered as one of the most dangerous threats to any inter-networking system. In this research work, we prevent insider attacks through SAAP, where the user must identify his/her self by using fingerprint.

R7: Stolen Verifier Attack and Data Modification Attack

In this research work, we used a mobile phone to verify user authentication. Where if the mobile phone is stolen, it is impossible to use it for accessing the cloud server because the attacker needs the fingerprint to access the service.

R8: Impersonation Attack

The presented framework does not transmit user ID and PW in the plaintext form; instead, user ID and PW will be hashed and encrypted before they were transmitted. Moreover, the framework usesa onetime sheared key, the key delivered to the user through a secret channel.

R9: Phishing Attack

The presented framework, includes Mutual authentication between the user and the server (Section 5.2.1), only authenticated server can send *b* and QR which consider the server identifier, that will be verified by the user.

R10: Server Masquerade Attack

In a masquerade attack, the attacker forges the identity to get access to the cloud server or to get higher privileges than they are authorized. The attacker needs to steal another user identity to performed a masquerade attack. According to the presented framework, there is no way to stole user identity, where the user used two security factors, which are the password and fingerprint, to verify himself/herself to the server. This made the identity very hard to be stolen.

5.3. Comparative analyses between different frameworks

In the following tables (Table 5.1 and Table 5.2), the comparative analyses of this research work with previous frameworks are given, which indicate the contribution and enhancement of this work.

- Functionality Requirements

Table 5.1: Comparative Analyses Based on Functionality Requirement

Functionality Requirement	Previous Framework					Our Frame-work
	Choudhury, A. J. et al.[4]	**Nayak, S. K. et al.[22]**	**Al-Attab, B. S., & Fadewar, H. S. [19]**	**Fan, K. et al., [2]**	**Raina, P., & Patel, B. [24]**	
F1	✓	✓	✓	x	✓	✓
F2	✓	✓	✓	x	x	✓
F3	✓	X	✓	x	✓	✓
F4	✓	✓	✓	x	x	✓
F5	✓	✓	✓	x	x	✓
F6	x	X	x	✓	✓	✓
F7	x	X	x	x	x	✓

- Security Requirements

Table 5.2: Comparative Analyses Based on Security Requirements

Security Requirements	Previous Framework					Our Frame-work
	Choudhury, A. J. et al.[4]	**Nayak, S. K. et al.[22]**	**Al-Attab, B. S., & Fadewar, H. S. [19]**	**Fan, K. et al., [2]**	**Raina, P., & Patel, B. [24]**	
R1	✓	✓	✓	✓	x	✓
R2	✓	x	✓	✓	✓	✓
R3	x	x	✓	x	x	✓
R4	x	x	x	x	x	✓
R5	✓	✓	✓	✓	✓	✓
R6	✓	x	✓	x	✓	✓
R7	✓	x	✓	✓	x	✓
R8	✓	x	x	✓	x	✓
R9	✓	x	x	x	✓	✓
R10	x	x	x	✓	x	✓

6. CONCLUSION AND FUTURE WORK

In this work, we present an enhanced framework to overcome the threat that may face the authentication property in cloud networking. The presented framework analyzed the most previous work shortages and resolved them where the framework construct under the concept of remote authentication, mutual authentication, identity management, etc. The framework defends against several types of attacks such as Replay attack, Impersonation attack, Phishing attack, Denial of Service attack, and others, which were mentioned in Section five. The presented framework consists of four main phases, which are the registration phase, login phase, authentication phase, and service access authentication phase (SAAP). We reinforced the registration phase to be more robust by using CAPTCHA recognition and session key agreement through QR code. The login and authentication phase accomplish through two levels of authentication, which are local authentication and server authentication, to guarantee only legitimate users can access the cloud server. Moreover, we provide the framework by SAAP to detect and defense against an internal attack such as an SYN flood attack. To evaluate the presented framework, we firstly prepare security analyses that justify each property and defense technique in the framework; then, we compare the presented framework with the most recently produced framework. The result shows that the presented framework exceeds all the comparative research. In future work, we plan to provide an instance of the presented framework to be able to implement it. So then we can test the efficiency of the presented framework.

REFERENCES

[1] D. Zissis and D. Lekkas, "Addressing cloud computing security issues", Future Generation Computer Systems, Vol.28, No.3, pp.583–592, 2012.

[2] Fan, K., Deng, H., Li, H., & Yang, Y. (2018). Privacy Protection Smartcard Authentication Scheme in Cloud Computing. *Chinese Journal of Electronics*, *27*(1),pp 2-5.

[3] Gao, Y., Fischer, R., Seibt, S., Parekh, M., & Li, J. (2017).Integrated Security *INFORMATIK,28(1),pp 4-9.*

[4] Choudhury, A. J., Kumar, P., Sain, M., Lim, H., & Jae-Lee, H. (2011, December). A strong user authentication framework for cloud computing. In Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific (pp. 110-115).

[5] Chen, N., & Jiang, R. (2013). Analysis and improvement of user authentication framework for cloud computing. In *Advanced Materials Research*,756, pp. 3482-3486.

[6] Jiang, R. (2013). ADVANCED SECURE USER AUTHENTICATION FRAMEWORK FOR CLOUD COMPUTING. *International Journal on Smart Sensing & Intelligent Systems*, *6*(4),pp 2-7.

[7] Chen, N., & Jiang, R.(2014). Security analysis and improvement of user authentication framework for cloud computing. *Journal of Networks*, *9*(1),pp2-5.

[8] Patel, S. C., Singh, R. S., & Jaiswal, S. (2015, February). Secure and privacy enhanced authentication framework for cloud computing. In *Electronics and Communication Systems (ICECS)*,pp. 1-3.

[9] Mun, J., Kim, J., & Won, D. (2016). An Improvement of User Authentication Framework for Cloud Computing. *JCP*, *11*(6), pp 3-7

[10] Darwish, M., Ouda, A., & Capretz, L. F. (2015). A cloud-based secure authentication (CSA) protocol suite for defense against Denial of Service (DoS) attacks. *Journal of information security and applications*, *20*,pp 3-7.

[11] Birkmann, J. (2006). Measuring vulnerability to promote disaster-resilient societies: conceptual frameworks and definitions. *Measuring vulnerability to natural hazards: Towards disaster resilient societies*, *1*, 9, pp 3-7.

[12] A. Singhal & M. Ramaiya, (2015). "A novel safe and efficient smartcardauthentication scheme using hash function", EngineeringUniverse for Scientific Research and Management, Vol.7, No.1,pp.1–6.

[13] Andrea Huszti &Norbert Olah. (2016). A simple authentication scheme for clouds.Conference: 2016 IEEE Conference on Communications and Network Security (CNS). DOI:10.1109/CNS.2016.7860549

[14] Hasan, M., Riaz, M. H., & Rahman, M. A. (2017). Authentication Techniques in Cloud and Mobile Cloud Computing. *International Journal of Computer Science and Network Security*, *17*(11),pp 28-32.

[15] Roy, S., Chatterjee, S., Das, A. K., Chattopadhyay, S., Kumar, N., & Vasilakos, A. V. (2017). On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services. *18*(*5)*,pp 2-4.

[16] Halagan, T., Kováčik, T., Trúchly, P., & Binder, A. (2015). Syn flood attack detection and type distinguishing mechanism based on Counting Bloom Filter. In *Information and Communication Technology*,12(1) pp. 30-39.

[17] Hussain, K., Hussain, S. J., Dillshad, V., Nafees, M., & Azeem, M. A. (2016). An Adaptive SYN Flooding attack Mitigation in DDOS Environment. *International Journal of Computer Science and Network Security*, *16*(7), pp5-8.

[18] Krämer, L., Krupp, J., Makita, D., Nishizoe, T., Koide, T., Yoshioka, K., & Rossow, C. (2015, November). Amppot: Monitoring and defending against amplification ddos attacks. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 615-636).

[19] Al-Attab, B. S., & Fadewar, H. S. (2016, December). Authentication scheme for insecure networks in cloud computing. In *Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), 2016 International Conference on* (pp. 158-163).

[20] Tsai, J. L., & Lo, N. W. (2015). A privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE systems journal*, *9*(3), pp 2-4.

[21] Huszti, A and Ol´ah, N. (2016). A Simple Authentication Scheme for Clouds, The 2nd IEEE Workshop on Security and Privacy in the Cloud, pp 1-4

[22] Nayak, S. K., Mohapatra, S., & Majhi, B. (2012). An improved mutual authentication framework for cloud computing. *International Journal of Computer Applications*, *52*(5), pp 2-4.

[23] Chang, V., Kuo, Y. H. & Ramachandran, M. (2016). Cloud computing adoption framework: A security framework for business clouds. *Future Generation Computer Systems*, *57*,pp 24-32.

[24] Raina, P., & Patel, B. (2017). Authentication Scheme in Cloud Computing Environment. International Journal of Advanced Research in Computer Science, 8(3),pp2-5.

AUTHOR

Dr. Hasan Al-Refai is an assistant professor and head of the Computer Information Systems department, faculty of IT for three years 2009 to 2012 at Philadelphia University, Jordan. He got his Ph.D. from the National University of Malaysia, 2005. He joined Philadelphia in the fall (first) semester 2006 after one year of experience as an assistant professor at Yarmouk University, Jordan. His research interests include Cryptography, Mobile Cryptographic protocols, E-Commerce Security, Formal Methods, Multimedia, Wireless networks & Mobile Computing, Software Engineering. He has written several journal articles and conference papers; he also had been the main advisor for many projects of undergraduate students as well as the main advisor for many thesis of Master students. He is a member of the IJOPCM Editorial Board, International Journal of Open Problems in Computer Science, **Program Committee of the** Third International Symposium on Innovation in Information & Communication Technology - ISIICT 2009 (From 15 - 17 December 2009) Jordan. A reviewer at Third International Symposium on Innovation in Information & Communication Technology - ISIICT 2009 (From 15 - 17 December 2009), Member of the steering committee of the fourth International Symposium on Innovation in Information & Communication Technology - ISIICT 2011 November 2011