Novel Mutual Authentication Protocol for Cloud Computing using Secret Sharing and Steganography

Nimmy K, M. Sethumadhavan
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham
Coimbatore, India
{m_sethu, k_nimmy}@cb.amrita.edu

ABSTRACT: *Proper authentication is an essential technology for cloud-computing environments in which connections to external environments are common and risks are high. Here, a new scheme is proposed for mutual authentication where the user and cloud server can authenticate one another. The protocol is designed in such a way that it uses steganography as an additional encryption scheme. The scheme achieves authentication using secret sharing. Secret sharing allows a part of the secret to be kept in both sides which when combined becomes the complete secret. The secret contains information about both parties involved. Further, out of band authentication has been used which provides additional security. The proposed protocol provides mutual authentication and session key establishment between the users and the cloud server. Also, the users have been given the flexibility to change the password. Furthermore, strong security features makes the protocol well suited for the cloud environment.*

Keywords: Mutual Authentication, Secret Sharing, Steganography, Cloud Computing, Session Key Establishment, OOB

Received: 12 March 2014, Revised 18 April 2014, Accepted 24 April 2014

1. Introduction

Cloud computing is essentially composed of a large-scale distributed and virtual machine computing infrastructure. This new paradigm delivers a large pool of virtual and dynamically scalable resources including computational power, storage, hardware platforms and applications to users via Internet technologies. Private and public organizations alike can make use of such cloud systems and services and many advantages may be derived when migrating all or some information services to the cloud computing environment. Examples of these benefits include increases in flexibility and budgetary savings through minimization of hardware and software investments.

One of the major challenges in cloud computing is mutual authentication so that both the parties authenticate themselves to the other before the communication begins. Authentication can be accomplished in many ways. User authentication can be handled using one or more different authentication methods. Some authentication methods such as plain password authentication are easily implemented but are in general weak and primitive. The fact that plain password authentication is still by far the most widely used form of authentication. Even though it doesn’t provide enough security on the Internet and even within private networks. Other methods of authentication, that may be more complex and require more time to implement and maintain, provide strong and reliable authentication (provided one keeps its secrets secret, i.e. private keys and phrases) [1]. There are protocols which are specifically designed to serve this purpose. Lin and Chang [2] proposed a countable and time-bound password-based user authentication scheme, which is a possible method for solving this problem. Hao, Zhong and Yu [3] proposed a new ticket-based mutual authentication scheme which uses smart card in the mutual authentication scheme. Each client has his/her own unique smart card. The clients' tickets are associated with her smart card, so that even when her tickets are lost, they cannot be used by other clients. A Albeshri and William Caelli [4] introduced an architecture which was a new approach to the problem identified as Mutual Protection for Cloud Computing (MPCC). The main concept underlying (MPCC) is based on a philosophy of Reverse Access Control, where clients control and attempt to enforce the means by which the cloud service providers control authorization and authentication within the dynamic environment, and the cloud provider ensures that the client organization does not violate the security of the overall cloud structure itself. Alman et al. [5] proposed a reliable and strong user authentication framework for cloud computing, where a legitimate user will prove his/her authenticity before entering into the cloud. This scheme verifies user authenticity using a two-step verification, which is based on a password, a smartcard and out of band (i.e. strong two factor) authentication. Nan Chen and Rui Jiang [6] proposed an advanced authentication protocol. Their improved protocol ensures user legitimacy before entering into the cloud This new protocol is designed by combining steganography and secret sharing. The rest of this paper is organized as follows: a description of the proposed protocol is available in section 2, section 3 provides the security analysis of the proposed scheme and section 4 concludes this paper.

Notation	**Description**
U_i	i^{th} user
ID_i	Identity of the ith user
PW_i	Password selected by user
H	One way hash function
S	Secret calculated by authentication server
HU_i	Hash of user information
b	Random number generated by user
ts	Time stamp
VP	Valid period
pwc	Password check
$share1_i$	Secret share of the user
$share2_i$	Secret share for the cloud server corresponding to the user
H_{sinfo}	Hash of authentication server information
E_{k_S}	Encryption using the secret key shared between the user and the authentication server
T_i	Onetime key
MAC_{T_i}	Message authentication code calculated using the key
K_{acs}	Secret shared between authentication server and cloud server
K_{cs_i}	Session key established between the cloud server and the user
$stego\ (x)$	Function that converts a cover image to a stego image

Table 1. Description of notations

2. The proposed protocol

The proposed protocol, depicted in Figure 1., achieves mutual authentication using steganography and secret sharing. The protocol consists of 4 phases: registration, login phase, mutual authentication and password change phase. There are mainly three participating entities in this protocol, the user, Authentication Server (AS) and Cloud Server (CS). Moreover, the protocol has been designed with the assumptions that the Cloud Service Providers and the users are assumed to be honest during the registration phase, the user and server are not trusted after the registration phase, and the communication channel between CS and AS is assumed to be a secure channel.

The description of notations used in this protocol is provided in Table 1.

2.1 Registration Phase

When the user U_i initially registers to *AS*, the registration phase is invoked. The user selects his/her unique user *id* and password. Hash of user information will be then embedded into an image and will be sent to the AS. The AS verifies the request and extracts information from the image. It then generates the secret, *S*, using the information sent by the user as well as the information of the AS. This secret will be embedded into an image using Pixel Value Differencing [8] and will be further divided into two shares using (2, 2) secret image sharing scheme based on addition [7]. One share will be sent to the user and the other will be sent to the CS. The share which has to be sent to the user will be written into a smart card along with other valuable information such as valid period. When the user receives the smart card he/she writes the random number *b* into the smart card. The registration phase, depicted in Figure 2 consists of the following steps:

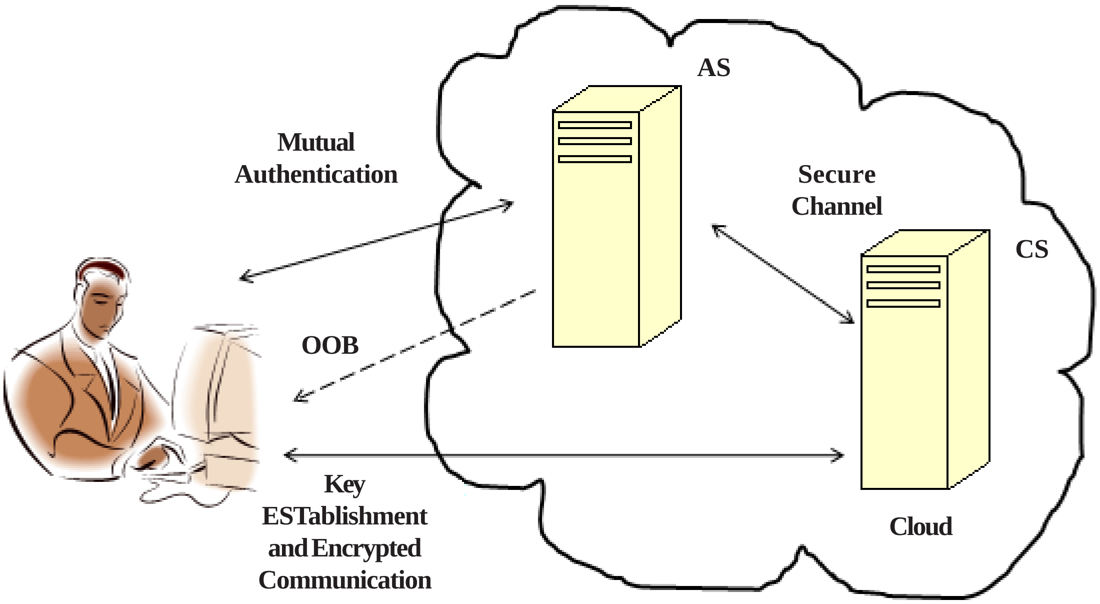

Figure 1. The Protocol

Step 1: U_i selects a user id ID_i and sends to *AS*

Step 2: AS checks whether $ID_{new} = ID_{existing}$. If doesn't exist, go to Step 1.

Step 3: U_i selects a password PW_i . Generates a random number *b* and computes $H(PW_i \oplus b)$.

Step 4: U_i computes: $HU_i = H(ID_i \,||\, H(PW_i \oplus b))$ and embeds HU_i , ID_i and timestamp *ts* into a cover image and sends to *AS*.

Step 5: *AS* extracts the message from the stego image, decides the Valid Period, *VP*. for the user and computes: Secret,

$$S = HU_i \oplus H_{sin\,fo} \quad \text{Also, } pwc = HU_i$$

i

Step 6: *AS* embeds *S* into a secret image which will be used to produce two shares using secret sharing, $share1_i$ and $share2_i$.

Step 7: *AS* $share1_i$, *VP*, *pwc*, ID_i , HU_i writes and a secret key, K_s, generated by the server into a smart card and sends to U_i.

$$share\,2_i \,||\, \{ID_i\}_{K_{acs}} \text{ will be sent to CS}$$

Step 8: User writes *b* into the smart card.

2.2 Login Phase

This phase is invoked when the user wants to login to the cloud. The user uses the smart card for this purpose. The user enters the user *id* and password for local system verification. The local system embeds the user information and sends to AS. The AS generates a one-time key and sends to the user using secure OOB channel. The user enters received key into the local system and the local system sends the share of the user to AS for verification. The login phase is depicted in Figure 3. The procedure is described below:

Step 1: U_i inserts the smart card and enters ID_i and PW_i.

Step 2: The local system computes: $HU_i\ (new) = H\ (ID_i\ ||\ H\ (PW_i \oplus b))$ and checks if $HU_i\ (new) = pwc$, then proceeds to next step else aborts.

Step 3: Local system embeds $E_{K_{si}}\{ID_i, ts\}$ into an image and sends $stego\ (E_{K_{si}}\{ID_i, ts\})$ to *AS*.

Step 4: AS extracts and decrypts the message. Generates onetime key $E_{K_{si}}\{\ T_i\ \}$and sends to U_i using secure OOB (Out of band) channel to user's mobile phone.

Step 5: U_i enters the onetime key $E_{K_{si}}\{T_i\}$ to the local system and the local system decrypts the message

Step 6: The local system computes $MAC_{T_i}\{share1_i\}$ and $share1_i\ ||\ E_{K_{si}}\{MAC_{T_i}\{share1_i\}, ID_i, ts, VP\}$ and sends to *AS.*

2.3 Mutual Authentication Phase

This phase is invoked when the server receives the share from the user. The AS sends a request to CS for the second share. Once it receives the share from CS combines and produces the secret image from which the secret *S* will be extracted. The authentication succeeds when the hash of user information matches with the information stored. At this moment authentication of the user is completed. Also, AS calculates the session key and sends that to CS. The CS uses the session key to send an acknowledgement to the user for further communication. The local system, when it receives the acknowledgement message from the CS, calculates the session key at its end and uses the key to decrypt the message sent by the CS. Here the authentication of the server to the

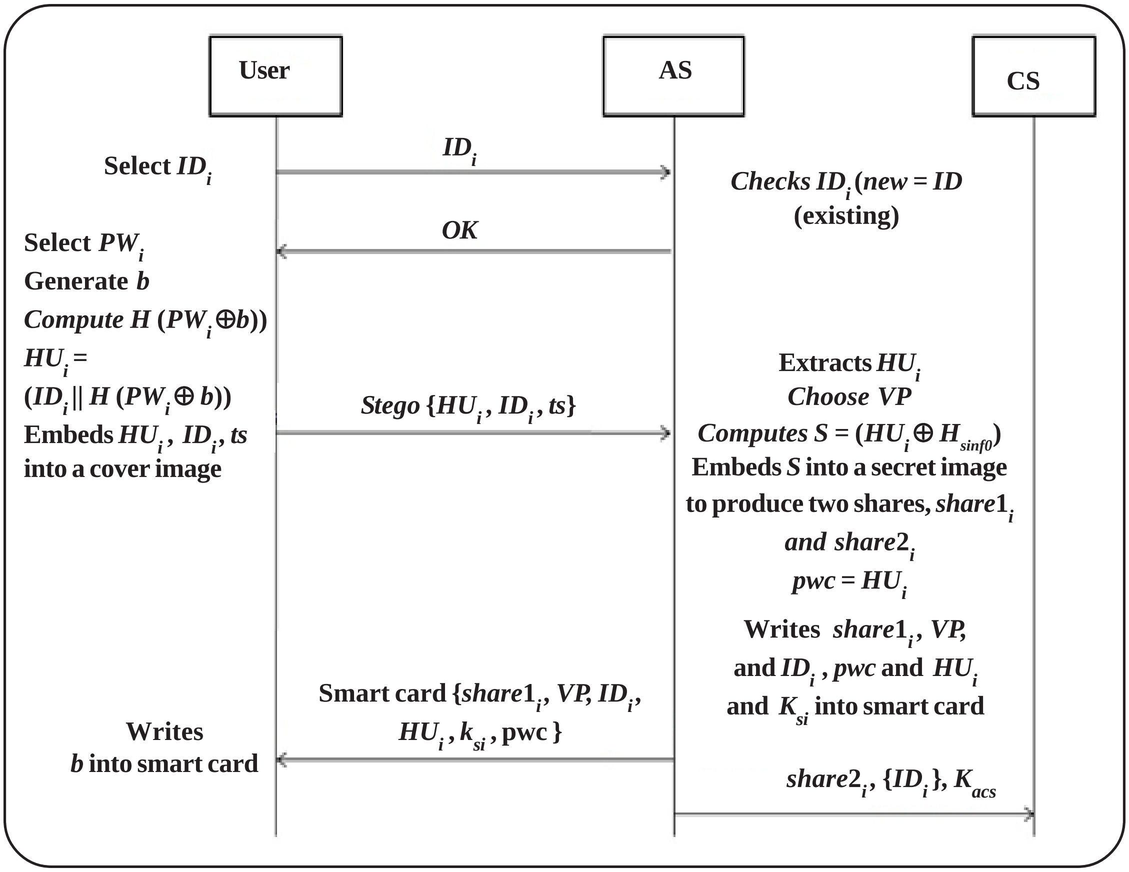
User
AS
CS
Select ID_i
ID_i
Checks ID_i(new = ID (existing)
OK
Select PW_i
Generate b
Compute $H\ (PW_i \oplus b))$
$HU_i =$
$(ID_i\ ||\ H\ (PW_i \oplus b))$
Embeds HU_i, ID_i, ts into a cover image
Stego $\{HU_i, ID_i, ts\}$
Extracts HU_i
Choose VP
Computes $S = (HU_i \oplus H_{sinf0})$
Embeds S into a secret image to produce two shares, $share1_i$ and $share2_i$
$pwc = HU_i$
Writes $share1_i$, VP, and ID_i, pwc and HU_i and K_{si} into smart card
Smart card $\{share1_i, VP, ID_i, HU_i, k_{si}$, pwc $\}$
Writes b into smart card
$share2_i, \{ID_i\}, K_{acs}$

Figure 2. Registration Phase

user happens and that completes the mutual authentication phase. The mutual authentication phase, depicted in Figure 4., is as follows:

Step 1: *AS* decrypts the message and checks the VP, if finds expired sends a message to user and aborts, else calculates $MAC_{T_i}\{share1_i\}$and checks whether $MAC_{T_i}\{share1_i\} = MAC_{T_i}\{share1_i\}$ (*received*), also checks the time stamp *ts*. If succeeds proceeds to next step or aborts.

Step 2: AS sends $E_{K_{acs}}\{ID_i, ts\}$ to the *CS* over the secure channel.

Step 3: CS sends $share2_i \parallel E_{K_{acs}}\{MAC_{T_i}\{share2_i\}\}$ sends to AS.

Step 4: Calculates $MAC_{K_{acs}}(share2_i)$ and checks: and checks $MAC_{K_{acs}}(share2_i) = (received)$ and check *ts*

Step 5: *AS* combines $share1_i$ and $share2_i$ to obtain the secret image Further extracts secret from the secret image

Step 6: AS extracts HU_i using the equation $HU_i = S \oplus H_{sinf0}$. Also calculates the Session Key, $K_{cs_i} = H(HU_i \oplus T_i)$ and sends the message $E_{K_{acs}}\{ID_i, ts, K_{cs_i}\}$ to *CS*

Step 7: CS embeds that into a an image and sends $stego(E_{K_{acs}}\{ID_i, ts\})$ to U_i

Step 8: The local system also calculates K_{cs_i}, extracts, decrypts and checks the ID_i and *ts*. The user can further communicate with *CS* using the session key K_{cs_i}.

2.4 Password Change Phase

The password change phase helps the user to change his/her password at his/her will. This is a very important requirement in user authentication schemes when it comes to security. The user inserts the smart card and enters user id and password for local verification. Once it is done user can select the change password option which will enable the user to change the password. The user has to select a new password for this purpose. The local system generates a random number and computes hash value of the new password xored with the random number and this is written into the smart card. The procedure for password change phase is discussed below:

Step 1: U_i inserts the smart card and enters ID_i and PW_i.

Step 2: U_i selects the change password option in the local system.

Step 3: U_i selects a new password $PW_i(new)$. Generates a random number and computes $b(new)$ and computes $H(PW_i(new) \oplus b(new))$.

Step 4: U_i computes: $pwc(new) = H(ID_i \parallel H(PW_i(new) \oplus b(new)$ and deletes the old values, writes it into the smartcard.

The password used only for local verification to login to the system.

3. Security Analysis

In this section, we examine the security of this proposed scheme and analyze the resistivity of this scheme to various known attacks.

3.1 Out-of-Band Authentication:

Out-of-Band authentication provides human interaction which makes the protocol stronger. Phone based out-of-band authentication works because no additional hardware, software or training is required for the end user. Since most users already carry mobile phones, phone communication can occur in true real time, phone authentication can require interaction with a human being, the authentication process can be a "*closed-loop*" with certainty of completion and a strong, humanly understandable audit trail of the transaction is captured.

3.2 Secret Sharing

The protocol is implemented using (2, 2) secret sharing scheme where 2 shares are needed simultaneously to retrieve the secret.

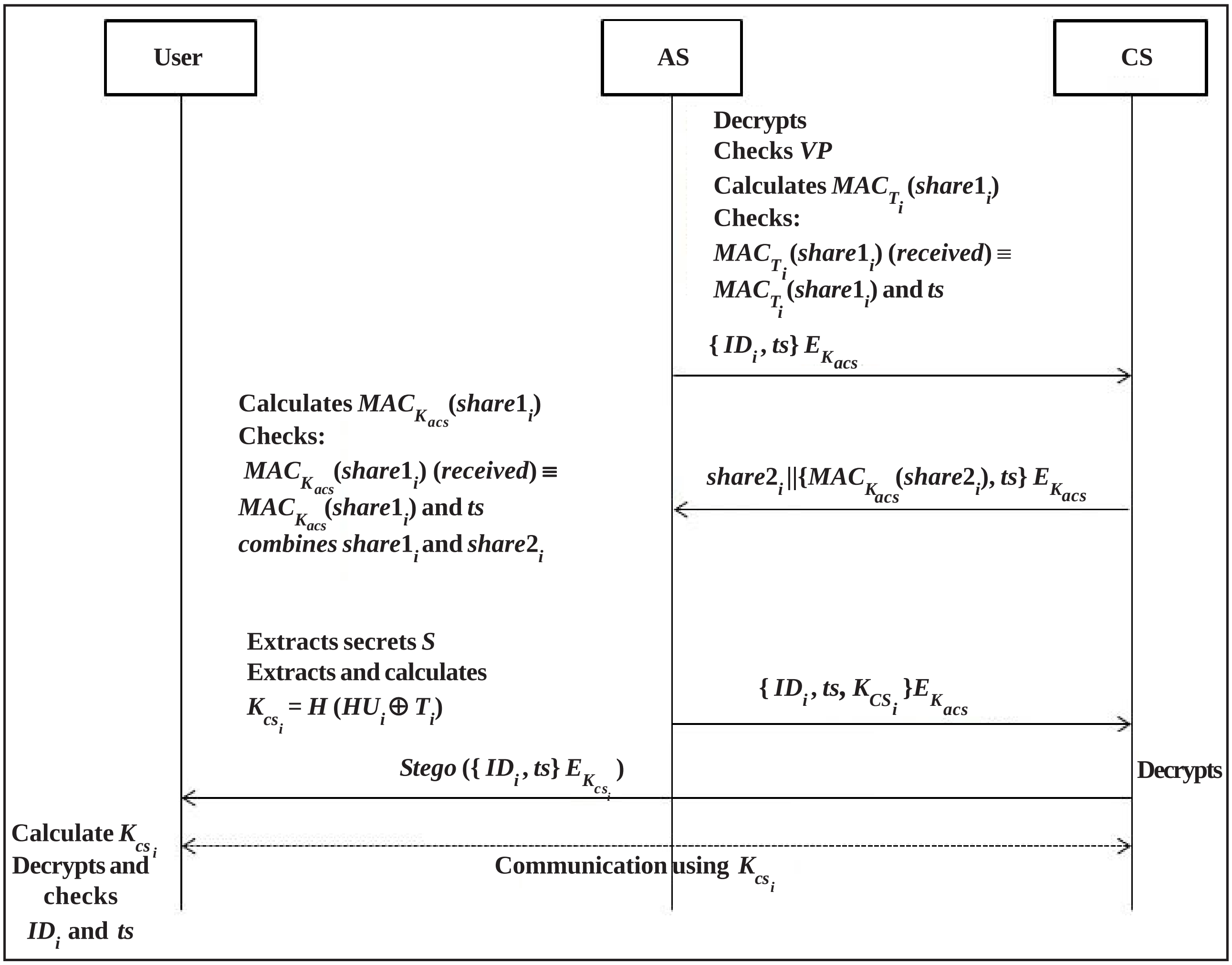
User
AS
CS
Decrypts
Checks *VP*
Calculates $MAC_{T_i}(share1_i)$
Checks:
$MAC_{T_i}(share1_i)$ (*received*) ≡
$MAC_{T_i}(share1_i)$ and *ts*
$\{ID_i, ts\} E_{K_{acs}}$
Calculates $MAC_{K_{acs}}(share1_i)$
Checks:
$MAC_{K_{acs}}(share1_i)$ (*received*) ≡
$MAC_{K_{acs}}(share1_i)$ and *ts*
combines $share1_i$ and $share2_i$
$share2_i \| \{MAC_{K_{acs}}(share2_i), ts\} E_{K_{acs}}$
Extracts secrets *S*
Extracts and calculates
$K_{cs_i} = H(HU_i \oplus T_i)$
$\{ID_i, ts, K_{CS_i}\} E_{K_{acs}}$
Stego $(\{ID_i, ts\} E_{K_{cs_i}})$
Decrypts
Calculate K_{cs_i}
Decrypts and
checks
ID_i and *ts*
Communication using K_{cs_i}

Figure 4. Mutual Authentication Phase

The AS embeds the secret into an image and the image will be split into two secrets namely *share*1 and *share*2. The first share *share*1 is kept by the user and *share*2 by *CS*. Secret sharing ensures that the secret which contains both the user information and authentication server information is available to neither the user nor the cloud server. Also, since partial information is sent across the network, the Man in The Middle (MITM) attack will not succeed to obtain the secret. Also, secret sharing ensures that both parties, user and the cloud server, take part in the authentication process which results in mutual authentication.

3.3 Steganography

This provides additional security and acts as an additional encryption. So far no known steganalysis scheme can extract the information from an image embedded using some sophisticated algorithms [9].

3.4 Mutual Authentication

Mutual Authentication is a security feature in which a client process must prove its identity to a server, and the server must prove its identity to the client, before any application traffic is sent over the client-to-server connection. A well-designed mutual authentication solution protects against other forms of online fraud such as phishing, man in the middle attacks, shoulder surfing, Trojan horses, keyloggers and pharming [10]. In the proposed protocol, the user will be authenticated when the *AS* produces the secret and retrieves the user information out of it. Also the cloud server *AS* is authenticated when user id encrypted using the session key is decrypted successfully by the user using the key calculated at user’s end. So the proposed system achieves mutual authentication successfully.

3.5 Resistance to Replay Attack

The presence of time stamp in all messages avoids replay attack. Also the session key is valid only for that particular session and it changes for each login of the user.

3.6 Resistance to Masquerade attack

Even if an attacker gains access to the local system by stealing the password and the smart card, the session key establishment will not happen as the one time key is sent to the user's mobile phone. Also server authentication prevents server masquerading or spoofing.

3.7 Resistance to Denial of Service Attack (DoS)

The local system verification avoids multiple requests being sent to the server and hence prevents DoS attacks.

3.8 Password Change

Proposed scheme facilitate users to change password any time as shown in password change phase. Password change facility makes a scheme inherently stronger compared to static password based schemes. Moreover it gives flexibility to users such that, users can change password if they forget or if they get hacked. Hence, it is user friendly. Also the password change phase does not affect the whole process. Only the password, used for local system verification, is changed.

3.9 Resistance to Insider Attacks

Insider attack is the most hazardous threat to any inter-networking system. In the proposed scheme, the secret *S* is partitioned into two shares and one is kept by the cloud server *CS*. An insider, if succeeds to obtain the partial secret, can't make use of it as it will not help the attacker to authenticate himself/herself to the *AS*. Authentication always requires the other share also from the user.

4. Conclusion

This paper proposes a significant mutual authentication scheme for cloud computing with many security features such as mutual authentication, session key agreement between the users and the cloud server and password change option. The proposed scheme introduces a new way of authentication using secret sharing. Out-of-band authentication provides human interaction which makes the protocol stronger as no additional hardware or software or training is required for the end user. In addition, cloud computing being a combination of computing resources; resource constrains are given less priority to provide high security to the cloud. Hence, this paper has not performed any performance comparison with some existing schemes. The proposed protocol can resist many popular attacks such as replay attack, man in the middle attack, and denial of service attack.

References

[1] Okuhara, M., Shiozaki, T., Suzuki, T. (2010). Security Architecture for Cloud Computing, *FUJITSU Sci. Tech. J*, 46 (4) 397–402.

[2] Lin, I. C., Chang, C.C. (2009). A countable and time-bound password-based user authentication scheme for the applications of electronic commerce, *Information Sciences*, 179 (9) 1269–1277.

[3] Hao, Z., Zhong, S., Yu, N. (2006). A Time-Bound Ticket-Based Mutual Authentication Scheme for Cloud Computing, *Int. J. of Computers*, *Communications & Control*, VI (2011), No. 2 (June), p. 227-235.

[4] Albeshri, A. A., Caelli, W. (2010). Mutual protection in a cloud computing environment, *In*: IEEE 12th International Conference on High Performance Computing and Communications (HPCC). p. 641–646.

[5] Choudhury, A. J., Kumar, P., Sain, M., Lim, H., Jae-Lee, H. (2011). A Strong User Authentication Framework for Cloud Computing, in Services Computing Conference (APSCC), IEEE Asia-Pacific, p. 110–115.

[6] Wu, D. C., Tsai, W. H. (2003). A steganographic method for images by pixel-value differencing, *Pattern Recognition Letters*, 24, p. 1613–1626.

[7] Nan Chen, Rui Jiang. (2014). Security Analysis and Improvement of User Authentication Framework for Cloud Computing, *Journal of Networks*, p. 198-203.

[8] Dong, L., Ku, M. (2010). Novel (*n*, *n*) secret image sharing scheme based on addition, in Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Sixth International Conference on, p. 583–586.

[9] Provos, N., Honeyman, P. (2002). Detecting steganographic content on the Internet, *In*: Proceedings of Network and Distributed System Security Symposium (San Diego, Feb. 6–8). *Internet Society*, Reston, VA.

[10] SearchSecurity, On the web at *http://searchsecurity.techtarget.com/opinion/Time-for-a-closer-look-at-software-security.*