Enhanced Time-Bound Ticket-Based Mutual Authentication Scheme for Cloud Computing

Ravi Singh Pippal
Radharaman Institute of Research and Technology, Bhopal, India
E-mail: ravesingh@gmail.com

Jaidhar C. D.
Defence Institute of Advanced Technology, Girinagar, Pune, India
E-mail: jaidharcd@diat.ac.in

Shashikala Tapaswi
ABV-Indian Institute of Information Technology and Management, Gwalior, India
E-mail: stapaswi@iiitm.ac.in

Keywords: authentication, cloud computing, cryptanalysis, impersonation attack, smart card

Received: December 3, 2012

Cloud computing is a recently developed technology for complex systems with services sharing among various registered users. Therefore, proper mutual authentication is needed between users and cloud server prior to avail the services provided by cloud servers. Recently, Hao et al. [26] proposed time-bound ticket-based mutual authentication scheme for cloud computing. However, this paper shows that their scheme is vulnerable to Denial-of-Service attack and insecure password change phase. Besides, enhanced scheme is proposed to overcome these security pitfalls. Moreover, performance comparison of both the schemes proves that the enhanced scheme is more efficient in comparison with Hao et al.'s scheme.

Povzetek: V tem članku je predlagana okrepljena shema medsebojne avtentifikacije aplikacij v oblaku, ki odpravi nekatere varnostne slabosti.

1 Introduction

Cloud computing is a new computing paradigm and got wide popularity from both industries as well as academia since 2007. It is employed because of its powerful computing and storage capabilities necessary in a distributed environment [1]. Its attractive characteristics include on-demand self-service, measured service, location independent resource pooling, ubiquitous network access and rapid elasticity. Three types of service offered by cloud computing are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Several firms like Google, Amazon, Microsoft, IBM and Yahoo are the ancestors that offer services for Internet users. Some more firms like Facebook, Salesforce, Myspace, Youtube, etc. are also started offering cloud computing services.

Users who are acquainted to use Internet can avail the computing resources, storage space and software services as per their demands to solve their problems. Further, users can also store their data in cloud servers and the same can be accessed from anywhere over the Internet as on-demand. This offers great flexibility for remote users.

Although, it provides a number of advantages such as cost reduction, dynamic resource provisioning, increased flexibility, low capital expenditures and time saving for new service deployment. However, still it is not matured enough to preserve data confidentiality as well as integrity. Many security issues, like data security either in store form or transmission form, application security, monitoring and metering need to be addressed and so on. Number of security issues have been discussed [2, 3, 4, 5, 6] and few research works address the security issues [7, 8, 9, 10].

One of the primary security needs is user authentication. Several authentication schemes have been proposed in the literature but most widely used one is password based authentication scheme [11, 12, 13, 14]. However, single factor password based authentication is not secure enough in the present scenario. Two factor authentication is a better option using password as one and smart card as other factor. Smart card is a tamper resistant integrated circuit card with memory to store personal information and a processor capable of performing computations [15].

In this context, many password based smart card authentication schemes have been proposed in order to avoid the use of the verification tables [16, 17, 18, 19]. Subsequently, authentication based on smart card has been employed continuously in several applications like healthcare [20], key exchange in IPTV broadcasting [21, 22], wireless networks

[23], authentication in multi-server environment [24], wireless sensor networks [25] and many more.

1.1 Contribution of this Paper

Cloud servers authenticate the remote users prior to offer any services to them. Recently, Hao *et al.* [26] proposed time-bound ticket-based mutual authentication scheme for cloud computing. It is claimed that the scheme resists lost smart card attacks, offline password guessing attack, lost ticket attack, masquerade attack and replay attack. In addition, it provides mutual authentication and secure session key generation. This paper shows vulnerabilities of Hao *et al.*'s scheme, i.e. vulnerable to Denial-of-Service attack and insecure password change phase. To resist these weaknesses, this paper proposes an enhancement to Hao *et al.*'s scheme.

The rest of this paper is organized as follows. Section 2 gives review of Hao *et al.*'s scheme. Security pitfalls of Hao *et al.*'s scheme is shown in section 3. Section 4 describes the proposed enhanced mutual authentication scheme. An in-depth security analysis and performance comparison is discussed in section 5. Finally, section 6 concludes the paper.

2 Review of Hao *et al.*'s Scheme

This section describes Hao *et al.*'s time-bound ticket-based mutual authentication scheme for cloud computing [26] (see Figure 1). The scheme consists of four phases: Registration phase, Verification request phase, Mutual authentication phase and Password change phase. The notations used throughout this paper are summarized in Table 1.

Table 1: Notations used in this paper

Symbols	Their meaning
U_i	Remote user
ID_i	Identity of U_i
PW_i	Password chosen by U_i
S	Cloud server
U_a	Attacker
PW_a	Password chosen by U_a
t	Number of digital tickets needed by U_i
$T_i^{(j)}$	j^{th} ticket of U_i
$TID_i^{(j)}$	j^{th} ticket ID
$VP_i^{(j)}$	Valid period of $T_i^{(j)}$
k_1,k_2	Two long term secret keys of S
$H(\cdot)$	Cryptographic hash function
$H_k(\cdot)$	Keyed hash function
$\|$	Concatenation
$\oplus$	Bitwise XOR operation
r_u	Random nonce generated by U_i
r_s	Random nonce generated by S
r_a	Random nonce generated by U_a
K_c/K_s	Shared session key between U_i and S

2.1 Registration Phase

This phase is invoked when a new user registers with the cloud server. The cloud server issues 't' tickets, in which each ticket can be used only once. In this phase, U_i selects ID_i, PW_i and a random number b, computes $IPB_i = H(ID_i \parallel H(PW_i \oplus b))$ and submits $\{ID_i, IPB_i, t\}$ to S over a secure channel, where 't' is the number of digital tickets needed by U_i.

Upon receiving the registration request and ticket fee from U_i, S generates t tickets for U_i. j^{th} ticket of U_i and its validity is represented as $\{(TID_i^{(j)}, VP_i^{(j)}), j = 1, 2, ..t\}$. S computes

$$W_i = IPB_i \oplus H(ID_i, K_1)$$

$$\alpha_i^{(j)} = H_{K_2}(ID_i \parallel TID_i^{(j)} \parallel VP_i^{(j)})$$

$$\beta_i^{(j)} = \alpha_i^{(j)} \oplus IPB_i$$

$T_i^{(j)}$ has two parts,

$$T_i^{(j)} = (T_i^{(j)1}, T_i^{(j)2})$$

in which

$$T_i^{(j)1} = (TID_i^{(j)}, VP_i^{(j)})$$

$$T_i^{(j)2} = \beta_i^{(j)}$$

S also computes $Z_i = H_{K_2}(ID_i) \oplus IPB_i$ and issues a smart card to U_i by storing $\{ID_i, t, W_i, Z_i, T_i^{(j)}\}$ into smart card memory over secure channel. After receiving, U_i stores b into smart card memory.

2.2 Verification Request Phase

As U_i receives t tickets, these tickets can be used to perform data verification at most t times. Suppose, for k^{th} verification request, U_i inserts the smart card to the card reader and keys in ID_i and PW_i. The smart card generates a nonce r_u and computes

$$IPB_i = H(ID_i \parallel H(PW_i \oplus b))$$

$$H_i = W_i \oplus IPB_i$$

$$C_1 = r_u \oplus H_i$$

$$C_2 = H(r_u) \oplus T_i^{(k)2} \oplus IPB_i$$

U_i sends the verification request $\{ID_i, T_i^{(k)1}, C_1, C_2\}$ to S in order to pass the mutual authentication phase.

2.3 Mutual Authentication Phase

Once the verification request has been received, S first checks the validity of ID_i to accept/reject the verification request. S rejects the request when it finds invalidity otherwise checks whether $TID_i^{(k)}$ is on the bulletin board or not. If it's on the bulletin board, S rejects U_i's request and terminates the process. S checks whether the current date is within the range of $VP_i^{(k)}$ or not. If not, S rejects U_i's request and terminates the process.

If all these conditions hold, S computes

$$D_0 = H(ID_i, K_1)$$

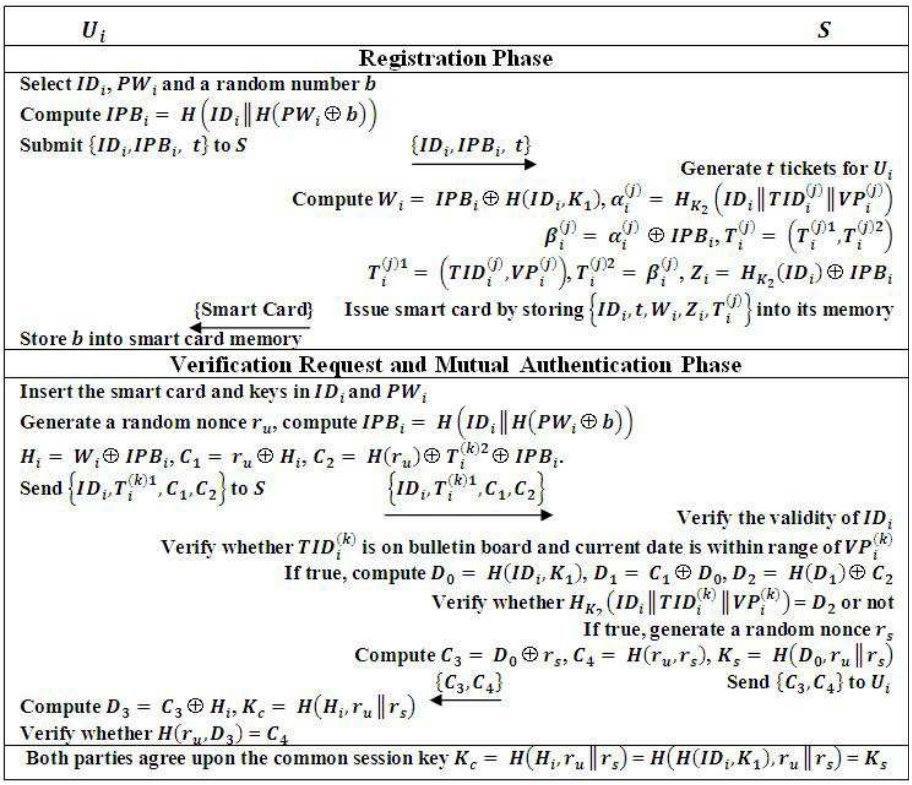

Figure 1: Hao *et al.*'s Scheme

$$D_1 = C_1 \oplus D_0$$
$$D_2 = H(D_1) \oplus C_2$$

S computes $H_{K_2}(ID_i \parallel TID_i^{(k)} \parallel VP_i^{(k)})$ and checks whether it is equal to D_2 or not. If true, S generates a random nonce r_s, computes $C_3 = D_0 \oplus r_s$, $C_4 = H(r_u, r_s)$ and sends the message $\{C_3, C_4\}$ to U_i. S also computes $K_s = H(D_0, r_u \parallel r_s)$ as the session key.

After getting the message $\{C_3, C_4\}$ from S, U_i computes $D_3 = C_3 \oplus H_i$ and compares $H(r_u, D_3)$ with C_4. If true, U_i authenticates S successfully otherwise terminates the session. Subsequently, U_i computes $K_c = H(H_i, r_u \parallel r_s)$. Both parties agree upon the common session key $K_c = H(H_i, r_u \parallel r_s) = H(H(ID_i, K_1), r_u \parallel r_s) = K_s$.

2.4 Password Change Phase

This phase is invoked when U_i wants to change the password. U_i inserts the smart card to the card reader and keys the credentials such as ID_i and PW_i. The smart card generates a nonce r_u and computes

$$IPB_i = H(ID_i \parallel H(PW_i \oplus b))$$
$$C_1 = r_u \oplus W_i \oplus IPB_i$$
$$C_2 = H(r_u) \oplus Z_i \oplus IPB_i$$

The smart card sends $\{update, ID_i, C_1, C_2\}$ to S, in which, $update$ denotes that it's a password change request. After receiving, S checks the validity of ID_i to accept/reject the request. If it is invalid, then S rejects the request otherwise computes

$$D_1 = C_1 \oplus H(ID_i, K_1)$$
$$D_2 = H(D_1) \oplus C_2$$

S computes $H_{K_2}(ID_i)$ and checks whether it is equal to D_2 or not. If true, S generates a random nonce r_s, computes $C_3 = H(ID_i, K_1) \oplus r_s$, $C_4 = H(r_u, r_s)$ and sends the message $\{C_3, C_4\}$ to U_i. Upon receiving the message $\{C_3, C_4\}$, smart card computes $D_3 = C_3 \oplus W_i \oplus IPB_i$ and compares $H(r_u, D_3)$ with C_4. If true, U_i authenticates S successfully otherwise terminates the session. Subsequently, smart card prompts U_i to enter a new password PW_i^{new}. Then, smart card computes

$$IPB_i^{new} = H(ID_i \parallel H(PW_i^{new} \oplus b))$$
$$W_i^{new} = W_i \oplus IPB_i \oplus IPB_i^{new} = H(ID_i, K_1) \oplus IPB_i^{new}$$
$$Z_i^{new} = Z_i \oplus IPB_i \oplus IPB_i^{new} = H_{K_2}(ID_i) \oplus IPB_i^{new}$$

The smart card updates $T_i^{(j)2}$ to $T_i^{(j)2} \oplus IPB_i \oplus IPB_i^{new}$ for all remaining tickets which yields $\alpha_i^{(j)} \oplus IPB_i^{new}$.

3 Weakness in Hao *et al.*'s Scheme

This section provides security flaws in Hao *et al.*'s scheme. They are (a) exposed to Denial-of-Service attack due to lack of early wrong password detection prior to verification request creation and (b) inefficient password change phase. It is assumed that the attacker U_a is able to intercept all the messages exchanged between U_i and S.

3.1 Denial-of-Service Attack

To check whether or not the requested user is a legitimate bearer of smart card, entered password must be verified at the smart card level before login request creation [27]. In this scheme, if U_a gets U_i's smart card by any means, he or she can create invalid login request by entering wrong password which is verified only at the cloud server side not at the user side.

Assume, U_a gets/steals U_i's smart card, inserts the smart card into the card reader and enters the wrong password PW_a as well as ID_a. Smart card creates an invalid login request without verifying the correctness of entered password or identifier. The smart card generates a nonce r_a and computes

$$IPB_a = H(ID_a \parallel H(PW_a \oplus b))$$
$$H_a = W_i \oplus IPB_a = IPB_i \oplus H(ID_i, K_1) \oplus IPB_a$$
$$C_{1a} = r_a \oplus H_a = r_a \oplus IPB_i \oplus H(ID_i, K_1) \oplus IPB_a$$
$$C_{2a} = H(r_a) \oplus T_i^{(k)2} \oplus IPB_a$$

U_a sends the verification request $\{ID_i, T_i^{(k)1}, C_{1a}, C_{2a}\}$ to S. This request fails to pass the authentication phase at the cloud server side. As a result, load on S increases which leads to Denial-of-Service attack. To overcome this attack, both password and identifier must be verified at the user side prior to compute verification request.

3.2 Insecure Password Change Phase

Communication is needed between S and U_i during the password change phase. Password change at the user side without interacting with S strengthen the security and reduces the load on S. Further, password change phase leads to Denial-of-Service attack because of non existence of earlier password as well as identifier verification before the update request creation [27].

4 Proposed Enhanced Mutual Authentication Scheme

This section describes proposed enhanced mutual authentication scheme over Hao *et al.*'s scheme (see Figure 2). The scheme consists of four phases: Registration phase, Verification request phase, Mutual authentication phase and Password change phase. The details of these phases are as follows:

4.1 Registration Phase

In this phase, U_i selects ID_i, PW_i and a random number b, computes $H(PW_i \oplus b)$ and submits $\{ID_i, H(PW_i \oplus b), t\}$ to S over a secure channel, where 't' is the number of digital tickets needed by U_i. Upon receiving the registration request and ticket fee from U_i, S generates t tickets for U_i. j^{th} ticket of U_i and its validity is represented as $\{(TID_i^{(j)}, VP_i^{(j)}), j = 1, 2, ..t\}$. S computes

$$W_i = H(ID_i \parallel H(PW_i \oplus b))$$

$$X_i^{(j)} = H_x(ID_i \parallel TID_i^{(j)} \parallel VP_i^{(j)}) \oplus H(ID_i, x)$$

where 'x' is long term secret key of S. $T_i^{(j)}$ has two parts,

$$T_i^{(j)} = (T_i^{(j)1}, T_i^{(j)2})$$

in which

$$T_i^{(j)1} = (TID_i^{(j)}, VP_i^{(j)})$$
$$T_i^{(j)2} = X_i^{(j)}$$

S issues a smart card over secure channel to U_i by storing $\{ID_i, t, W_i, T_i^{(j)}\}$ into smart card memory. After receiving, U_i stores b into smart card memory.

4.2 Verification Request Phase

As U_i receives t tickets, these tickets can be used to perform data verification at most t times. Assume for k^{th} verification request, U_i inserts the smart card to the card reader and keys the credentials, ID_i' and PW_i'. The smart card computes $W_i' = H(ID_i' \parallel H(PW_i' \oplus b))$ and compares it with the stored W_i. If true, U_i is the valid owner of smart card.

The smart card generates a nonce r_u and computes $Y_i = H_{T_i^{(k)2}}(T_i^{(k)2} \parallel r_u)$. U_i sends the verification request $\{ID_i, T_i^{(k)1}, Y_i, r_u\}$ to S.

4.3 Mutual Authentication Phase

Upon receiving the verification request $\{ID_i, T_i^{(k)1}, Y_i, r_u\}$; S first checks the validity of ID_i to accept/reject the verification request. S rejects the request when it finds invalidity otherwise checks whether $TID_i^{(k)}$ is on the bulletin board or not. If it's on the bulletin board, S rejects U_i's request and terminates the process. S checks whether the current date is within the range of $VP_i^{(k)}$ or not. If not, S rejects U_i's request and terminates the process.

If all these conditions hold, S computes $X_i^{(k)} = H_x(ID_i \parallel TID_i^{(k)} \parallel VP_i^{(k)}) \oplus H(ID_i, x)$. S computes $Y_i' = H_{X_i^{(k)}}(X_i^{(k)} \parallel r_u)$ and checks whether it is equal to received Y_i or not. If true, S authenticates U_i otherwise rejects the request. S generates a random nonce r_s, computes $Z_i = H_{X_i^{(k)}}(r_u \parallel r_s \parallel X_i^{(k)})$ and sends the message $\{ID_i, Z_i, r_s\}$ to U_i. S also computes $K_s = H(ID_i \parallel r_u \parallel r_s \parallel X_i^{(k)})$ as the session key.

After getting the message $\{ID_i, Z_i, r_s\}$ from S, U_i computes $Z_i' = H_{T_i^{(k)2}}(r_u \parallel r_s \parallel T_i^{(k)2})$ and compares it with the received Z_i. If true, U_i authenticates S successfully otherwise terminates the session. Subsequently, U_i computes $K_c = H(ID_i \parallel r_u \parallel r_s \parallel T_i^{(k)2})$. Both parties agree upon the common session key $K_c = H(ID_i \parallel r_u \parallel r_s \parallel T_i^{(k)2}) = H(ID_i \parallel r_u \parallel r_s \parallel X_i^{(k)}) = K_s$.

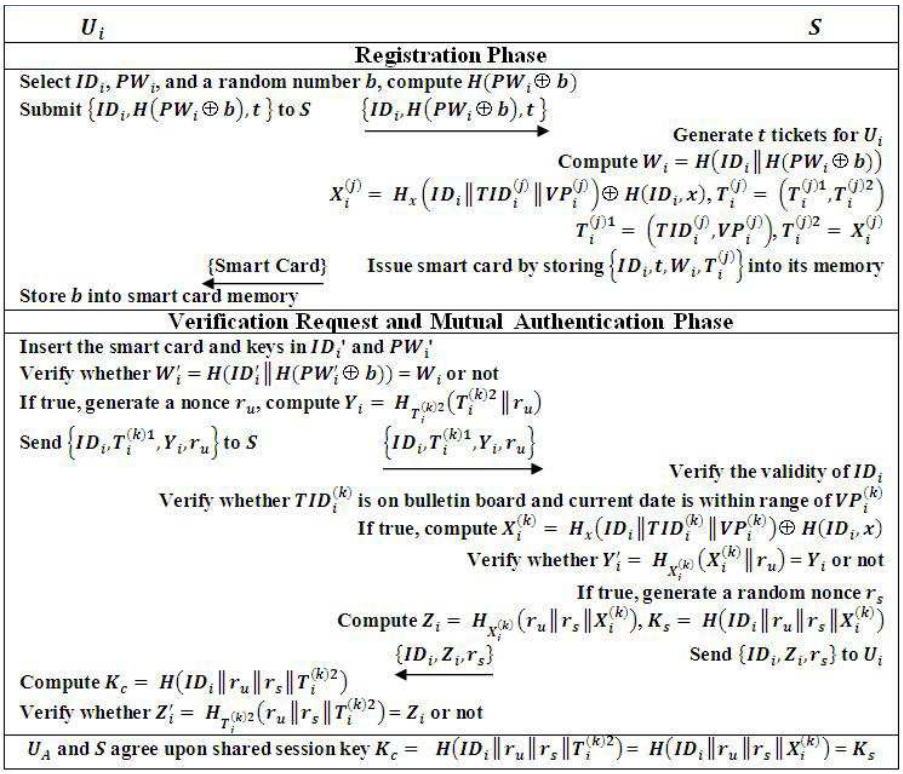

Figure 2: Proposed Enhanced Mutual Authentication Scheme

4.4 Password Change Phase

This phase is invoked when U_i wants to change the password. U_i inserts the smart card to the card reader and keys the credentials such as ID_i' and PW_i'. The smart card computes $W_i' = H(ID_i' \parallel H(PW_i' \oplus b))$ and compares it with the stored W_i. If true, U_i is the legitimate bearer of smart card.

Subsequently, smart card prompts U_i to enter a new password PW_i^{new}. Then, smart card computes $W_i^{new} = H(ID_i \parallel H(PW_i^{new} \oplus b))$. The smart card updates W_i to W_i^{new} in the smart card memory.

5 Security Analysis and Performance Comparison

This section discusses security analysis of the proposed enhanced mutual authentication scheme and provides performance analysis in comparison with Hao *et al.*'s scheme.

5.1 Impersonation Attack

Suppose, U_a has complete hold on the insecure communication channel and can intercept all the communicating messages transmitted between U_i and S. U_a is unable to create a forged verification request as the value of $T_i^{(k)2}$ is needed to compute fake Y_i. Further, it is not possible to get $T_i^{(k)2}$ from intercepted $T_i^{(k)1}$ without knowing 'x', long term secret key of S. Moreover, without the information about $T_i^{(k)2}$, U_a cannot masquerade as a legitimate S. Hence, U_a is unable to forge the verification request to impersonate a valid U_i or forge the response message to impersonate a legitimate S.

5.2 Password Guessing Attack

One of the most important features provided by any authentication scheme is the security of passwords of users. The scheme must be structured in such a way that no one can guess the password. In the proposed scheme, password is used only in the card holder verification. It is not used in the calculation of any of the verification request parameters. Hence, there is no chance of offline password guessing attack. To resist online password guessing attack, the number of attempts made by user can be limited to some fixed value.

5.3 Replay Attack

An adversary may try to act as an authentic user by resending previously intercepted messages. This scheme uses unique ticket ID TID_i and random nonces r_u and r_s which are different from session to session. As a consequence, U_a cannot enter the system by resending previously transmitted messages to impersonate legal U_i.

Assume that the intercepted verification request

$\{ID_i, T_i^{(k)1}, Y_i, r_u\}$ is replayed to pass the mutual authentication phase. Upon receiving the verification request, S first checks the validity of ID_i and then checks whether $TID_i^{(k)}$ is on the bulletin board or not. Obviously, S will find that $TID_i^{(k)}$ is on the bulletin board. S rejects the service request and terminates the process.

5.4 Reflection and Parallel Session Attack

To resist reflection and parallel session attacks, the given scheme employs asymmetric structure of communicating messages, i.e., $\{ID_i, T_i^{(k)1}, Y_i, r_u\}$ and $\{ID_i, Z_i, r_s\}$. There is no symmetry in the values of $Y_i = H_{T_i^{(k)2}}(T_i^{(k)2} \parallel r_u)$ and $Z_i = H_{X_i^{(k)}}(r_u \parallel r_s \parallel X_i^{(k)})$. Hence, U_a is unable to launch parallel session attack by replaying cloud server response message as the user verification request or reflection attack by resending user verification request as the cloud server response message.

5.5 Privileged Insider Attack

For remembrance, many users employ same password to access different servers. Nevertheless, a privileged insider of server can get this password and then try to utilize it for personal benefit. In the given scheme, U_i sends $H(PW_i \oplus b)$ to S instead of PW_i to resist privileged insider attack. Hence, this scheme provides security against privileged insider attack.

5.6 Valid Period Extending Attack

In the proposed scheme, no one can use the ticket after the expiration date. It helps to control the database growth maintained by S. Let us suppose, U_i wants to reuse the k^{th} ticket $T_i^{(k)}$. U_i changes $VP_i^{(k)}$ to $VP_i^{(k')}$ (by including the current date) and sends $\{ID_i, T_i^{(k')1}, Y_i, r_u\}$ to S.

Once received, S computes $X_i^{(k')} = H_x(ID_i \parallel TID_i^{(k)} \parallel VP_i^{(k')}) \oplus H(ID_i, x)$. Obviously, S finds Y_i' = $H_{X_i^{(k')}}(X_i^{(k')} \parallel r_u) \neq Y_i$ and rejects the request. Hence, the enhanced scheme is able to prevent the user from extending the expiration date of any ticket.

5.7 Early Wrong Password Detection

To provide security against Denial-of-Service attack, identity of users must be verified at the user side prior to creation of verification request. The enhanced scheme verifies the entered password and identifier by comparing W_i' with the stored W_i during the verification request phase. If U_i enters either password or identifier incorrect, the smart card prompt U_i to re-enter correct password as well as correct identifier. In addition, it is infeasible to guess correct identifier and password simultaneously by using stolen smart card. Hence, there is no chance for Denial-of-Service attack.

5.8 Efficient Password Change Phase

In the proposed scheme, U_i can choose and change the password without any support from S. The smart card compares the computed W_i' with the stored W_i to verify the legitimacy of U_i before the update of new password. If it holds, smart card asks U_i to enter a new password PW_i^{new}, computes W_i^{new} and updates W_i to W_i^{new} in the smart card memory. It eliminates the role of S during password change phase which diminishes burden on S.

5.9 Performance Comparison

In order to measure the security in terms of possible attacks, proposed scheme is compared with Hao *et al.*'s scheme. From Table 2, it can be clearly seen that the proposed scheme is more secure in comparison with Hao *et al.*'s scheme. It includes early wrong password and wrong identifier detection which resists Denial-of-Service attack either during verification request phase or password change phase.

Table 3 shows comparative results for Hao *et al.*'s scheme and the proposed enhanced scheme in terms of computational complexity. In this table, t denotes the number of tickets issued to user U_i and r denotes the number of tickets remaining. From both the tables, it is clear that the proposed scheme is more efficient in comparison with Hao *et al.*'s scheme.

6 Conclusion

Nowadays, cloud has become one of the most popular business transaction platform. However, the growing security threat emerging due to the present security attacks obfuscates this powerful network. Weak authentication of responses and requests allows the attackers to compromise the cloud infrastructure. Hence, authentication of both the users and the cloud servers is a vital issue. To address this aforementioned issue, Hao *et al.* [26] proposed time-bound ticket-based mutual authentication scheme for cloud computing.

This paper pointed out that Hao *et al.*'s scheme is inadequate to provide security against Denial-of-Service attack. Further, password change phase is also insecure. To overcome these security flaws, this paper proposes an enhanced scheme over Hao *et al.*'s scheme. The enhanced scheme inherits all the merits of Hao *et al.*'s scheme and resists the identified security attacks. In addition, user can choose and change the password securely without any assistance from the cloud server.

Acknowledgement

The authors would like to thank ABV-Indian Institute of Information Technology and Management, Gwalior, India for providing the academic support.

Table 2: Comparison between proposed scheme and Hao *et al.*'s scheme in terms of security properties

Security Properties	**Hao *et al.*'s Scheme**	**Proposed Scheme**
User is allowed to choose and change the password	Yes	Yes
Provides mutual authentication	Yes	Yes
Provides secure session key generation	Yes	Yes
Resists replay attack	Yes	Yes
Resists guessing attack	Yes	Yes
Resists parallel session attack	Yes	Yes
Resists reflection attack	Yes	Yes
Resists privileged insider attack	Yes	Yes
Resists valid period extending attack	Yes	Yes
Resists impersonation attack	Yes	Yes
Resists Denial-of-Service attack	No	Yes
Free from cloud server involvement during password change	No	Yes
Provides early wrong password detection	No	Yes
Provides early wrong identifier detection	No	Yes

Table 3: Comparison between proposed scheme and Hao *et al.*'s scheme in terms of computational complexity

Authentication Schemes	**Name of Phases**	**No. of Hash Functions (H)**	**No. of Exclusive-or Operations (XOR)**	**Total No. of Operations**
Hao *et al.*'s Scheme	Registration Phase	$(4+t)$	$(3+t)$	$(24+t)$ H
	Verification Request Phase	(3)	(5)	
	Mutual Authentication Phase	(7)	(4)	$(27+t+2r)$ XOR
	Password Change Phase	(10)	$(15+2r)$	
Proposed Scheme	Registration Phase	$(3+t)$	$(1+t)$	$(17+t)$ H
	Verification Request Phase	(3)	(1)	
	Mutual Authentication Phase	(7)	(1)	$(5+t)$ XOR
	Password Change Phase	(4)	(2)	

References

[1] Li, Z., Chen, C. and Wang, K. (2011). Cloud computing for agent-based urban transportation systems. *IEEE Intelligent Systems*, 26(1), pp. 73–79.

[2] Zhou, M., Zhang, R., Xie, W., Qian, W. and Zhou, A. (2010). Security and privacy in cloud computing: A survey. *In Proceedings of* 6^{th} *International Conference on Semantics, Knowledge and Grid*, Shanghai, China, pp. 105–112.

[3] Subashini, S. and Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1), pp. 1–11.

[4] Pearson, S. and Benameur, A. (2010). Privacy, security and trust issues arising from cloud computing. *In Proceedings of* 2^{nd} *IEEE International Conference on Cloud Computing Technology and Science*, Bristol, U.K., pp. 693–702.

[5] Jensen, M., Schwenk, J., Gruschka, N. and Iacono, L. (2009). On technical security issues in cloud computing. *In Proceedings of IEEE International Conference on Cloud Computing*, Bangalore, India, pp. 109–116.

[6] Kandukuri, B.R., Ramakrishna, P.V. and Rakshit, A. (2009). Cloud security issues. *In Proceedings of IEEE International Conference on Services Computing*, Bangalore, India, pp. 517–520.

[7] Takabi, H., Joshi, J.B.D. and Ahn, G.J. (2010). SecureCloud: Towards a comprehensive security framework for cloud computing environments. *In Proceedings of* 34^{th} *Annual IEEE Computer Software and Applications Conference Workshops*, P.A., U.S.A., pp. 393–398.

[8] Wang, C. and Yan, H. (2010). Study of cloud computing security based on private face recognition. *In Proceedings of International Conference on Computational Intelligence and Software Engineering*, Beijing, China, pp. 1–5.

[9] Shen, Z. and Tong, Q. (2010). The security of cloud computing system enabled by trusted computing technology. *In Proceedings of* 2^{nd} *International Conference on Signal Processing Systems*, Wuhan, China, pp. 11–14.

[10] Zech, P. (2011). Risk-based security testing in cloud computing environments. *In Proceedings of* 4^{th} *IEEE International Conference on Software Testing, Verification and Validation*, Innsbruck, Austria, pp. 411–414.

[11] Hwang, M.S., Lee, C.C. and Tang, Y.L. (2001). An improvement of SPLICE/AS in WIDE against guessing attack. *Informatica*, 12(2), pp. 297–302.

[12] Yang, C.C., Chang, T.Y. and Hwang, M.S. (2003). Security of improvement on methods for protecting password transmission. *Informatica*, 14(4), pp. 551–558.

[13] Yoon, E.J., Ryu, E.K. and Yoo, K.Y. (2005). Attacks and solutions of Yang *et al.*'s protected password changing scheme. *Informatica*, 16(2), pp. 285–294.

[14] Ku, W.C. and Tsai, H.C. (2005). Weaknesses and improvements of Yang-Chang-Hwang's password authentication scheme. *Informatica*, 16(2), pp. 203–212.

[15] http://en.wikipedia.org/wiki/Smart_card.

[16] Chang, C.C. and Wu, T.C. (1991). Remote password authentication with smart cards. *IEE Proceedings E: Computers and Digital Techniques*, 138, pp. 165–168.

[17] Chen, T.H., Horng, G. and Wu, K.C. (2007). A secure YS-like user authentication scheme. *Informatica*, 18(1), pp. 27–36.

[18] Liao, C.H., Chen, H.C. and Wang, C.T. (2009). An exquisite mutual authentication scheme with key agreement using smart card. *Informatica*, 33(2), pp. 125–132.

[19] Pippal, R.S., Jaidhar, C.D. and Tapaswi, S. (2012). Highly secured remote user authentication scheme using smart cards. *In Proceedings of* 7^{th} *IEEE International Conference on Industrial Electronics and Applications*, Singapore, pp. 988–992.

[20] Hu, J., Chen, H.H. and Hou, T.W. (2010). A hybrid public key infrastructure solution (HPKI) for HIPAA privacy/security regulations. *Computer Standards and Interfaces*, 32(5-6), pp. 274–280.

[21] Yoon, E.J. and Yoo, K.Y. (2009). Robust key exchange protocol between set-top box and smart card in DTV broadcasting. *Informatica*, 20(1), pp. 139–150.

[22] Pippal, R.S., Tapaswi, S. and Jaidhar, C.D. (2012). Secure key exchange scheme for IPTV broadcasting. *Informatica*, 36(1), pp. 47–52.

[23] He, D., Ma, M., Zhang, Y., Chen, C. and Bu, J. (2011). A strong user authentication scheme with smart cards for wireless communications. *Computer Communications*, 34(3), pp. 367–374.

[24] Pippal, R.S., Jaidhar, C.D. and Tapaswi, S. (2013). Robust Smart Card Authentication Scheme for Multiserver Architecture. *Wireless Personal Communications*. DOI: 10.1007/s11277-013-1039-6.

[25] Fan, R., He, D., Pan, X. and Ping, L. (2011). An efficient and DoS-resistant user authentication scheme for two-tiered wireless sensor networks. *Journal of Zhejiang University-SCIENCE C (Computers and Electronics)*, 12(7), pp. 550–560.

[26] Hao, Z., Zhong, S. and Yu, N. (2011). A time-bound ticket-based mutual authentication scheme for cloud computing. *International Journal of Computers, Communications and Control*, 6(2), pp. 227–235.

[27] Yoon, E.J., Ryu, E.K. and Yoo, K.Y. (2005). An improvement of Hwang-Lee-Tang's simple remote user authentication scheme. *Computers and Security*, 24(1), pp. 50–56.