Efficient Password-based Two Factors Authentication in Cloud Computing

Ali A. Yassin, Hai Jin, Ayad Ibrahim, Weizhong Qiang and Deqing Zou

Cluster and Grid Computing Lab
Services Computing Technology and System Lab
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, 430074, China
Aliadel2005alamre@yahoo.com, hjin@hust.edu.cn

Abstract

Security threats are considered the main barrier that precluded potential users from reaping the compelling benefits of the cloud computing model. Unfortunately, traditional password authentication jeopardizes user privacy. Anonymous password authentication (APA) represents a promising method to maintain users' privacy. However, the major handicap that faces the deployment of APA is the high computation cost and inherent shortcomings of conventional password schemes. In our proposed scheme, we present a new setting where users do not need to register their passwords to service provider. They are supplied with the necessary credential information from the data owner. Furthermore, for enabling the service provider to know the authorized users, data owner provides the service provider with some secret identity information that is derived from the pair (username/password) of each user. Our approach shows good results in terms of high scalability which makes our scheme more suitable to the cloud environment, strong authentication that withstands different known attacks.

Keywords: *cloud authentication; zero-knowledge proof; service provider; password authentication; privacy-preserving; asymmetric scalar-product-preserving encryption (ASPE)*

1. Introduction

Cloud computing considers a modern computing model which aims to supply customers by many advantages such as: reduced costs, flexibility, re-provisioning of resources and more mobility, and thereby increased profits. It has attracted significant interest from users whom they can be accessed to the stored data anytime, anywhere when the internet is available. However, security issue imposes strong hindrance for users' adoption of cloud services. As the center security element, password authentication plays a vital role in a new computing model. So, identity theft stills is one of the most predominant issues in the cloud systems. In the past, the important data such as username/password, saved in a PC, are difficult to access by other users except the valid user. In this case, the sensitive data is protected physically. Unfortunately, cloud computing platform has not provided such physical protection, and all protections rely remarkably on the mechanism of user authenticating [1].

In fact, most customers tend to pick something such as phone number, favorite games and name as their passwords. These things are easily to memorize. Consequently, adversaries can build a table of significant words to transgress the system, which is named dictionary attack.

Furthermore, using the password based authentication still suffers from the traditional security attacks, namely on-off-line, *Man-in-the-Middle* (MITM) attacks. In on-line attack, adversaries reckon all the possible users' passwords, iteratively select one guess password to adopt online transactions with the service provider and then confirm their possible using the reply from the service provider. In off-line attack, adversaries iteratively choose one possible password and confirm their guess by using information eavesdropped during communication channels. Since the service provider is not involved in the attack sessions, adversaries can again and again launch their attacks without evocation any attention of the service provider until they obtain a hit. The on-line guessing attack can be circumvented simply by either using time restrictions or limiting the number of successive login tries of the suspect user. Whereas, off-line guessing attacks are much harder to be thwarted, and they require more effort to be obviated. In the MITM attack, the adversary attempts to intercept the password of valid user and then impersonates of victim user to login when the user signs out of the server [2, 3].

Due to the bridled references in password authentication, the security and performance overhead must be equilibrated in practical implementation.

Viet et al. [4] proposed the first anonymous password authentication that aggregates a password scheme with the *Private Information Retrieval* (PIR) scheme. There are some problems regarding this scheme. First, PIR requires from the server to be passed a whole database to detect user. Second, it cannot resist on-line guessing attacks.

The smart card-based authentication methods [5] implement two factors of the authentication research. In the first factor, users' investigation credentials are saved in the smart card while in the second factor, the smart card has been preserved by password. These two-factors do not need the server to store a password file. The negative side of smart card is that it is not a simple device, and the card reader considers an extra expense. It also requires additional middleware application to obtain a match between smart card and communication standards.

The biometric data has been used [6] to confirm the digital identity of the user by using his biometric features such as iris, speech. However, it suffers from reply attack and is limited because these methods require extra devices and additional time cost for extracting and processing it. Moreover, it becomes unacceptably slow when a large of users logged to the system at the same time.

In this paper, we present an efficient and secure password-based mutual authentication and key exchange scheme using ASPE for the cloud environment with high performance. We analyze the security to explain that our proposed scheme is well resisted for malicious attacks.

Our contributions in this work can be summarized as follows:

1. The main feature (and novelty) of our proposed scheme derives from the use of ASPE to achieve key confirmation and solves password’s breakthrough risks in new technique.
2. The service provider can management login of the user to prevent the abuse of login.
3. Our work does not need to use a PIR and the service provider computation is not linked to the number of customers in the system.
4. Our scheme enjoys many features as follows (a) users can freely select and change their passwords; (b) it supports mutual authentication between a service provider and a user; (c) it includes user anonymity. Moreover, our scheme is secure against on-line, off-line, dictionary, MITM and replay attacks.

This paper presents the necessary primitives and requirements of our work in section 2. An overview of related work is discussed in section 3. The proposed scheme is addressed in section 4. The experimental results are covered in section 5. Conclusions are presented in section 6.

2. Cryptographic Primitives

For more visibility, we review the major cryptographic primitives to be applied in our proposed scheme.

2.1 Zero Knowledge Proof

We can explain briefly the overall work of *ZN* between two parties [7]: *Prover P* and *Verifier V*. Given g as a generator of group *Gp* and let *p*, *q* are two prime numbers, such that $p=2q+1$. *V* wants to convince P that he possesses the value *x* with respect to *y* such that $y=g^x$ without revealing x value to *V*. First, *P* starts the game by selecting a random number $r_x \in Z_P^*$, assigns a commitment $t = g^{r_x}$ and sends it to the verifier. Upon receiving *t*, *V* accepts the challenge and feeds back $\alpha \in Z_P^*$ to *P*. Then, *P* responds by performing some operations and returns response $z_x = r_x - c \times x$ to *V*. Finally, *V* agrees while $g^{z_x} y^c = t$ holds. Zero knowledge proofs are used broadly to enable the work of two main problems, namely, discrete logarithm and square-root problem that tightly depends on the prime numbers' theory. The advantages of *ZN* are as follows:

- When the method completes, the verifier does not have any sensitive information related to the privacy of the prover.
- ZN method assists the cloud authentication and makes it more flexible, easier, accurate, and faster.

2.2 Asymmetric Scalar-Product-Preserving Encryption

ASPE is one of the best encryption method [8]. The source of data leakage, the adversary can access to point's distances of encrypted data. ASPE overcomes this limitation by presenting encryption function that prevents to disclosure actual distance information.

Suppose *Enc* as encryption function and Enc (p, K) represents an encrypted value of a point p by key K. The *Enc* refers an ASPE if and only if *Enc* exists in scalar product of the query point with an encrypted database point [8].

1. $p_i.q = Enc(p_i,K).Enc(q,K)$; so p_i is any point in database and q denotes any query point and,
2. $p_i.p_j \neq Enc(p_i,K).Enc(p_j,K)$ for any point in database such that p_i and p_j.

In above definition, we notice that q as encrypted query should not be matched to that of any point p_j in encrypted database even $p_i = q$. These conditions make all points in database and query points must be encrypted in a different manner. The encryption function $Enc_T()$ is different from $Enc_Q()$ in the ASPE scheme. The scalar product of point p in the database and query point of q (displayed by column vectors) can be formed as $p^T I q$, where T is the transpose function, and the identity matrix represents as $I = (d \times d)$. We refer I as MM^{-1}; so M^{-1} is invertible matrix of M, i.e., $p^T q = p^T M M^{-1} q$. Now, the encrypted value of point p in the database is $p' = Enc_T(p,K) = M^T p$. At the same time, the value of encrypting query q is $q' = Enc_Q(q,K) = M^{-1} q$. Nobody can detect the value of p from p' without owning M.

3. Proposed Scheme

In this section, we present a new password authentication scheme and privacy-preservation for cloud environments. Our proposed scheme involves three components, *data owner* (*DW*), a user set, a server such as a *service provider* (*SP*). Our work consists of three stages—setup, registration, and authentication. Setup and registration stages are executed only once, and the authentication stage is executed whenever a user wishes to login. In the setup and registration stages, the user U_i registers her/his identity (username Un_i and password pw_i) into DW who saves Un_i and pw_i , and then provides public system parameters (*ZPK*) to service provider and each user in secure channel. We can describe this step as follows.

DW sets up $n=pq$; where p and q are two large primes. He selects $(M_i,M_i^{-1},g_i,k_i \in Z_n^*)$. *DW* uses a cryptographic hash function *H*(.), symmetric key encryption $Enc(.)$, and asymmetric scalar-product-preserving encryption $E_T(.)$. *DW* computes important information (f_i,y_i,x_i,s_i); where $x_i=H(pw_i)$, $y_i=g_i^{x_i} mod\ n$, $f_i=g_i^{un_i}y_i$, $s_i=pw_i^T M_i M_i^{-1} pw_i$, $pw_i^{'}=M_i^{-1}pw_i$. The public system parameters contain $ZKP=(g_i,k_i,n,H(.),Enc())$. Briefly, *DW* supplies U_i and *SP* by important information as follows.

1) DW → $U_i : ZKP,M_i,\mathrm{f_i},y_i,x_i$; *2) DW* →*SP*: $Un_i,H(y_i,k_i),pw_i^{'},s_i,ZKP$

U_i encrypts his important information (y_i,M_i,f_i,x_i,ZKP) by using private key pk_i , i.e., $Enc_{pk_i}(y_i,M_i,f_i,x_i,ZKP)$, he computes his private key by composing between pw_i and M_i , private key is $pk_i=pw_i \,||\, M_i$, where || means concatenation function. Then, U_i saves his credential file to his preferred storage such as USB.

After that, the user may use the authentication stage to login. *2FA* authentication session is qualified as follows.

1. U_i uses decryption function $Dec_{pk_i}(y_i,M_i,f_i,x_i,ZKP)$ to decrypt his credential file by pk_i and sends $(Un_i,H(y_i,k_i))$ to *SP* as a first factor.

   U_i → USB $: pk_i$
   USB →$U_i : ZKP,M_i,\mathrm{f_i},y_i,x_i$
   U_i → SP : $(Un_i,H(y_i,k_i))$
   SP → U_i : α

   When *SP* detects the identity of U_i , it will provide U_i by $\alpha \in Z_n^*$ which generates randomly for each login attempt of U_i .

2. U_i compute $E_{i1}=E_T(pw_i^T,M_i)=pw_i^T M_i$, generates a random $r_{x_i} \in Z_n^*$, and then calculates $t_i=g_i^{r_{x_i}}$.
3. U_i calculates $E_{i2}=Enc_{k_i}(y_i)$, $c_i=H(y_i,t_i,f_i,\alpha)$, $z_{x_i}=r_{x_i}-c_i x_i$ and $w_{x_i}=x_i-c_i x_i$.
4. Finally, U_i submits $(E_{i1},E_{i2},c_i,z_{x_i},w_{x_i})$ to *SP* as a login request (second factor).

   U_i → SP : $(E_{i1},E_{i2},c_i,z_{x_i},w_{x_i})$

5. Service Provider: Upon receiving the information in Step 4, *SP* performs the following steps:
   - *SP* computes $s_i^{'}=E_{i1}pw_i^{'}=pw_i^T M_i M_i^{-1} pw_i$ to check whether $s_i^{'}$ equals to the stored s_i . If so, *SP* computes as follows.
     1. $y_i^{'}=Dec_{k_i}(E_{i2})$, $t_i^{'}=(y_i^{'})^{c_i} g_i^{z_{xi}}$, and $f_i^{'}=(y_i^{'})^{c_i} g_i^{Un_i} g_i^{w_{xi}}$.

2. $c'_i = H(y'_i, t'_i, f'_i, \alpha)$.

The mathematical proofs (1, 2) demonstrate how *SP* obtains the secret parameters $(t_i^{'}, f_i^{'})$ from U_i.

- If $c_i^{'}$ equals to c_i that means U_i is an authorized user, so *SP* computes $E_i^{''} = Enc_{k_i}(t_i^{'} + f_i^{'})$ and then sends it to U_i. $SP \rightarrow U_i : E_i^{''}$

6. U_i will ensure the validity of *SP* by computing $E_{i3} = Enc_{k_i}(t_i + f_i)$. After that, he checks whether $E_{i3} = E_i^{''}$ or not. If the result of the comparison is true, *SP* is a valid server otherwise it is an impersonator party.

Prof (1):

$$t'_i = (y'_i)^{c_i} g_i^{z_{x_i}}$$
$$= (g_i^{x_i})^{c_i} g_i^{(r_{x_i} - c_i x_i)}$$
$$= g_i^{c_i x_i + r_{x_i} - c_i x_i} = g_i^{r_{x_i}} = t_i$$

Prof (2):

$$f'_i = (y'_i)^{c_i} g_i^{Un_i} g_i^{w_{x_i}}$$
$$= (g_i^{x_i})^{c_i} g_i^{Un_i} g_i^{x_i - x_i c_i}$$
$$= g_i^{Un_i} g_i^{x_i - x_i c_i + x_i c_i} = f_i$$

3.1 Security Analysis

In this section, we provide the security analysis of our proposed scheme. We will view that our scheme is secure against impersonation attack, off-line guessing attack, MITM attack, and supports mutual authentication, and unlinkability.

Mutual Authentication. This feature means that an adversary cannot impersonate the legal U_i to *SP*, and vice versa. In our work, authentication of U_i to *SP* represents by two factors $(Un_i, H(y_i, k_i))$ and $(c_i^{'}, t_i^{'}, f_i^{'}, s_i^{'})$. An adversary is not able to generate $(E_{i1}, E_{i2}, c_i, s_i, z_{x_i}, w_{x_i})$. These are unforgability of $(c_i, s_i, z_{x_i}, w_{x_i})$. Additionally, an adversary cannot conclude $E_i^{''}$ which should be matched with E_{i3}. Then, an adversary cannot impersonate *SP* to U_i.

Unlinkability. Our work enjoys this feature which ensures that *SP* cannot determine whether U_i has logged previously or not. We embed this feature in our work by changing the values $\left(\alpha, r_{x_i}, c_i, z_{x_i}, w_{x_i}, t_i\right)$ each time U_i tries to login. Therefore, *SP* cannot link different logins with the same user.

Off-line guessing. An adversary cannot use this type of attack in our work, so he is not able to decrypt both E_{i1} and E_{i2} since he does not have the necessary keys for decrypting this secret information. Moreover, an adversary does not have the ability to get the values of each (x_i, f_i, t_i, s_i). Subsequently, *SP* does not respond unless he ensures from the identity of the user. Therefore, our proposed scheme prevents disclosing any information through the communication protocol between U_i and *SP*.

MITM attack. It means that an adversary has the ability to intercept the messages between users and the server. Then, he uses this message when the user signs out the server.

In our proposed scheme, the credential is securely encrypted and sent to the service provider. Generation of the random value r_{x_i} is through the creation of sensitive data $\left(c_i, z_{x_i}, w_{x_i}, t_i\right)$ by U_i as a login request (second factor) to *SP*. These sensitive data became useless when U_i signs off the server. As a result, an adversary spotting communication between U_i and SP can learn r_{x_i} which is used only once; he is unable to compute $g_i^{r_{x_i}}$. Nevertheless, when U_i signs out of the server, an adversary cannot compute $\left(t_i, z_{x_i}, w_{x_i}\right)$ to guess new c_i for impersonating the user.

3.2 Performance Investigation

In this section, we conduct several experiments for gauging the efficiency and the effectiveness of our work. We test the effectiveness in terms of authentication accuracy. The efficiency of our work has been tested in term of measuring the response time of SP. We have registered during our experiments 2000 users and suppose that each user needs maximum 2 seconds for logging the system.

Unsurprisingly, the average time for the authentication stage of our work is equal to 0.0257 seconds for each user which indicates the high speed of our solution. We gain this average time from 100 runs of our proposed scheme.

4. Conclusion

This paper investigates the feasibility of adopting 2FA and anonymous password for user authentication in cloud computing environment. Security analysis explains that our proposed scheme can resist various possible attacks and gratify all the security requirements. The main comparison reinforces the good qualities of our work in contrast to the preceding anonymous password authentication schemes for cloud environment. Especially, our scheme can support the privacy preservation of password. Thus, our scheme proposes anonymity and security of the login users. In the performance appraisal, our presented scheme has been evidenced to achieve sturdy security with lower cost than its previous schemes. Our on-going research is to apply the current work to multi-cloud, hacker blocking.

Acknowledgements

The work is supported by National High-tech R&D Program (863 Program) under grant No. 2012A-A012600, National 973 Basic Research Program of China under grant No. 2007CB310900, Wuhan City Programs for Science and Technology Development under grant No. 201010621211, and Key Lab of Information Network Security, Ministry of Public Security under grant No. C11602.

References

[1] The top 10 cloud computing trends, http://www.cloudtweaks.com.

[2] Tech-cloud-security, http://fcw.com.

[3] M. Zhou, Z. Rong, W. Xie, W. Qian and A. Zhou, "Security and Privacy in Cloud Computing: A Survey", Proc. of the 6th International Conference Semantics Knowledge and Grid, **(2010)** pp. 105-112, Beijing, China.

[4] D. Q. Viet, A. Yamamura and T. Hidema, "Anonymous Password-Based Authenticated Key Exchange. Proceedings of the 6th International Conference on Cryptology in India, **(2005)** pp. 233-257; Bangalore, India.

[5] S. Jeon, H. S. Kim and M. S. Kim, "Enhanced Biometrics-based Remote User Authentication Scheme Using Smart Cards", J. of Security Engineering. 8, pp. 237-254 **(2011)**.

[6] J. Wayman, A. Jain, D. Maltoni and D. Maio, "An Introduction to Biometric Authentication Systems: Biometric systems Technology", Design and Performance Evaluation. Springer **(2005)**.

[7] S. Goldwasser, S. Micali and C. Rackoff, "The knowledge complexity of interactive proof-systems", Proceedings of the 17th Annual ACM Symposium on Theory of Computing, **(1985)**, pp. 291-304; New York , USA.

[8] E. Cho, G. Ghinita and E. Bertino, "Privacy-Preserving Similarity Measurement for Access Control Policies", Proceedings of the 6th ACM Workshop on Digital Identity Management, **(2010),** pp. 3-11; Chicago, USA.