Secure Password by Using Two Factor Authentication in Cloud Computing

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

Services Computing Technology and System Lab
Cluster and Grid Computing Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, 430074, China
aliadel2005alamre@yahoo.co, 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 make 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

In this section, we present a new password authentication scheme and privacy-preservation for cloud environments. Our proposed scheme is involved with 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 (Ui) registers her/his identity (username Un and password Pw) into DW who saves Un and Pw, 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 $(Mi, M_i^{-1}, gi, ki \in \mathbb{Z}_n^*)$. DW uses a cryptographic hash function $H(\cdot)$, asymmetric scalar-product-preserving encryption $E_T(\cdot)$, symmetric key encryption $Enc(\cdot)$ and $T$ transport function. DW
computes important information \((f_i, y_i, x_i, s_i)\), where \(x_i = H(Pw_i), y_i = g_i^{s_i} \mod n, f_i = g_i^{U_i y_i}, s_i = Pw_i^T M_i M_i^{-1} Pw_i, Pw_i = M_i^{-1} Pw_i\). The public system parameters contain \(ZKP = (g_i, k_i, n, H(\cdot), Enc(\cdot))\). Briefly, \(DW\) supplies \(U_i\) and \(SP\) by important information as follows. 1) \(DW \rightarrow U_i : ZKP, M_i, f_i, y_i, x_i\) 2) \(DW \rightarrow SP : U_i n_i, H(y_i, k_i), Pw_i^t, s_i, ZKP\).

\(U_i\) encrypts his important information \((ZKP, M_i, f_i, y_i, x_i)\) by using private key \(pk_i\), i.e., \(E_{Enc_{pk_i}}(ZKP, M_i, f_i, y_i, x_i)\), 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 authentication stage 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 \(Dec_{pk_i}(ZKP, M_i, f_i, y_i, x_i)\) to decrypt his credential file by \(pk_i\) and sends \((U_n_i, (y_i, K_i))\) to SP as a first factor. 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. \(U_i\) computes \(E_{1i} = E_T(Pw_i^T, M_i) = Pw_i^T M_i\), generates a random \(r_{xi_i} \in Z_n^*\), and then calculates \(t_i = g_i^{r_{xi_i}}\).
3. \(U_i\) calculates \(E_{12} = Enc_k(y_i, c_i = H(y_i, t_i, f_i, \alpha)\), \(z_{xi_i} = r_{xi_i} - c_i x_i\) and \(w_{xi_i} = x_i - c_i x_i\). Finally, \(U_i\) submits \((E_{13}, E_{12}, c_i, z_{xi_i}, w_{xi_i})\) to SP as a login request (second factor).
4. Service Provider: Upon receiving the information in Step 3, \(SP\) performs the following steps:
   - \(SP\) computes \(s_i' = E_{1i} Pw_i = Pw_i^T M_i M_i^{-1} Pw_i\) to check whether \(s_i'\) equals the stored \(s_i\). If so, \(SP\) computes as follows.
     a) \(y_i' = Dec_{pk_i}(E_{12}), t_i' = (y_i')^{c_i} g_i^{z_{xi_i}},\) and \(f_i' = (y_i')^{c_i} g_i^{U_i n_i} g_i^{w_{xi_i}}\).
     b) \(c_i' = H(y_i', t_i', f_i', \alpha).\) The mathematical proofs (1, 2) demonstrate how \(SP\) obtains the secret parameters \((t_i', f_i')\) from \(U_i\).
   - If \(c_i'\) equals \(c_i\) that means \(U_i\) is an authorized user, \(SP\) computes \(E_{1i}' = Enc_k(t_i' + f_i')\) and then sends it to \(U_i\).
5. \(U_i\) will ensure the validity of \(SP\) by computing \(E_{13} = Enc_k(t_i + f_i).\) After that, he checks whether \(E_{13} = E_{1i}'\) 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_{xi_i}} \\
= (g_i^{r_{xi_i}})^{c_i} g_i^{(r_{xi_i} - c_i x_i)} \\
= g_i^{c_i x_i + r_{xi_i} - c_i x_i} \\
= g_i^{r_{xi_i}} = t_i
\]

### Prof (2)

\[

t_i' = (y_i')^{c_i} g_i^{s_i} w_{xi_i} \\
= (g_i^{s_i})^{c_i} g_i^{s_i} g_i^{x_i - x_i c_i} \\
= g_i^{s_i} g_i^{x_i - x_i c_i + x_i c_i} \\
= f_i
\]

### 2 Compared with Previous Works and Performance Investigation

We compare security properties of our proposed scheme with ones of four authentication schemes, including Das et al. [1], Chien et al. [2], and Pathan et al. [3]. Table 1
Table 1. Comparison of authentication schemes

<table>
<thead>
<tr>
<th></th>
<th>Our Scheme</th>
<th>Das et al. [8]</th>
<th>Chien et al. [9]</th>
<th>Pathan et al. [10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>C2</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>C3</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>C4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>C5</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

C1: Freely chosen password; C2: User anonymity; C3: Secure password change; C4: session key agreement; C5: Mutual authentication.

describes comparison of security properties. We conduct several experiments for gauging the efficiency and the effectiveness of our work. We test the effectiveness in terms of authentication accuracy. 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.

3 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.

References