Efficient and Secure Mutual Authentication Scheme in Cloud Computing

Ali A.Yassin, Hikmat Z. Neima, Zaid Ameen Abduljabbar, Haider Sh.Hashim

Abstract - Nowadays, cloud computing considers an important topic for small, medium and large e-commences alike. The security is a constitutive trouble that hinders its widespread adoption. The password authentication is a first level of security in cloud computing, aiming to guarantee that only legitimate users are allowed to be used data that stored in the cloud server. The demeanor of multi-factor authentication schemes such as token, biometric provided a promising password authentication solution. There are many challenging matters that raise fears of using multi-factor are the high cost, not easy to carry, does not provide the functionalities of revocation, and fails to resist well-known attacks such as off-line guessing password, Man-in-the-Middle (MITM) Seed-tracing. In this paper, we propose a scheme of Two-Factor Authentication (2FA) that overcomes aforementioned issues and reduces the cost. We use Zero-Knowledge and One-Time Password (OTP) to implement a Cloud-based two-factor authentication as a design paradigm. Our proposed scheme includes many security characteristics like mutual authentication, user anonymity, session key agreement, freely chosen password, no time synchronization, and has a good performance of password authentication.

Keywords: **Cloud computing, Mutual Authentication, Zero-knowledge proof, Service provider, One-time password.**

I. INTRODUCTION

Cloud computing, as an originating computing model, allows customers to be remote saved their data with a cloud server, to gain services anytime and anywhere. There are many advantages from migrating user's data to the cloud side, since he can use data from the cloud side on-demand, using any device, without any attention at the cost of software and hardware infrastructures. The users can avoid extra cost and achieve the flexibility to scale exploitations on-demand [1, 2, 3]. The authentication schemes are based on password which depends on something the user knows. Usually, the users are selected simple passwords (e.g. names, phone number) which are comfortable to remember, but are suffered from malicious attacks. The exact opposite, complex and randomly passwords are more secure, but are hard to remember. Besides, when users selected simple passwords, attackers may generate a table of meaningful words to damage the system which is named dictionary attack.

Manuscript received on October, 2013.

Ali A.Yassin,Computer Science Dept., Education College, Basrah University, Basrah,61004, Iraq.

Hikmat Z. Neima, Computer Science Dept., Science College, Basrah University, Basrah,61004, Iraq.

Zaid Ameen Abduljabbar, Computer Science Dept., Education College, Basrah University, Basrah,61004, Iraq.

Haider Sh.Hashim, Computer Science Dept., Education College, Basrah University, Basrah,61004, Iraq.

The adversary can apply the dictionary in two essential manners [4, 5]. The first one is called the off-line guessing attack, where an adversary tries to get the communication data of genuine users' login requests and then checks through his dictionary to launch attacks. An adversary chooses one password after another from his dictionary, calculates the communication values with the selected password and then compares with the recorded data. If so, then the selected password is a valid password. The on-line guessing attack represents the second dictionary attack. An adversary picks one password after another from his dictionary and then employs these passwords to impersonate as the rightful user to ship a service request. An adversary chooses another password and tries again when his request failed. The standard methods of avoiding this attack are to either restrict the number times of failed runs before the password becomes invalid [4, 5].

2-Factor Authentication (2FA) is more suitable with principles of cloud authentication [6]. A user sends his username and password to the cloud server for authentication. The cloud server asks the user to send his second factor when it ensures from matching of user's username/password with a cloud server's database. The user gains permit to reach a cloud server's resources when his second factor has validity in the cloud server. The second factor can be one of Token, Smart Card, finger print, voice, etc. Only the genuine user has registered his second factor to the server in advance. However, the token cannot resist the MITM Seed-tracing, requires high cost, and when it is lost or stolen, the service provider security may compromise. Furthermore, how to arrange tokens $_{\text{issued}}$ by several cloud servers is a big trouble for user and server as well. The drawbacks of personal physiological, when a large number of users tried to authenticate to the system at the same time, the mechanism of the scheme becomes unacceptably slow. Moreover, the biometric factor requires extra hardware and software.

In this paper, we address the problem of the authentication process in the context of cloud service provider setting. Instead of using the traditional identity of second factor such as biometric and token techniques which require extra devices and cost, we designed a new scheme that does not require saving a password file on the server and depends on cryptography tools.

Our contributions in this work can be summarized as follows:

1. The proposed scheme contains important merits as follows: (1) it provides mutual authentication between user and service provider; (2) it accomplishes user anonymity; (3) The service provider and a user can achieve authenticated session's keys; (4) it allows users to freely choose their password; (5) it provides revocation phase when the user lost his authentication keys; (6) it describes by low cost, simple integration with available infrastructure, and essay to deploy and manage.
2. Our work presents a new setting of 2FA that depends on

two cryptography tools. The first factor is produced by zero knowledge and second factor views in a new manner of One-Time Password (OTP) mechanism.

3. Our work reduces the overload on the cloud server, where users do not save their password file in the service provider. Additionally, Our proposed scheme does not need to use synchronized clocks because we employ random numbers instead of timestamps.
4. Our proposed scheme can resist off-line guessing attacks, replay attacks, forgery attacks, MITM Seed-tracing attacks, and insider attacks.

This paper presents the necessary primitives and requirements of our work in section II. An overview of related work is displayed in section III. The proposed scheme is addressed in section IV. Conclusions are presented in section V.

II. DESIGN ISSUES

A. Problem Definitions

Fig. 1 illustrates the basic architecture of our proposed scheme, which composes of three components: Data owner (DW), Service Provider (SP), and users. DW is the owner of the data to be stored in the cloud such as databases, images, videos.

The essential role of SP is to ensure that just the authorized users can get accessed to the secret data. The overall work can be divided into three phases: Setup, Registration, and Authentication. In the setup and registration phases, the user U_i sends his username/ password to DW who sets up keys (important information) to use in the next phases. During the registration phase, DW issues to each user and SP important information (credential, public parameters) to be used for authentication. The user's credential file contains user's two-factor authentication. The valid user derives a key from his password, which is used for encrypting his credential file. Furthermore, the user will save his encrypted credential file in any extra device who prefers it. Each time, the user this sends his first factor to SP . After that , SP wants to login into the SP ; he decrypts his credential using key, and verifies user's first factor and sends back challenge value to the user for ensuring from validity of SP . Finally, the user sends his second factor to SP who checks again the validity of the user's second factor.

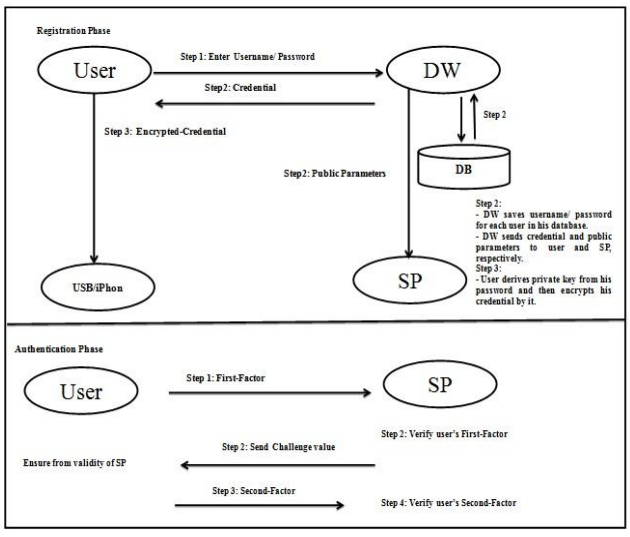

Figure.1. Basic architecture of our proposed authentication.

B. Zero Knowledge Proof

We can explain briefly the overall work of Zero knowledge (ZN) between two parties [7]: Prover P and Verifier V . Given g as a generator of group G_p and let both 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 $z_x = r_x - cx$ to V . Finally, V agree 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.

III. RELATED WORK

ZN authentication is applied in P2P networks [8]. However, they presented a modified method ZKPI (Zero-Knowledge Public Infrastructure) that required a key exchange mechanism to exchange the keys between the peers over the network. A new method for improving the ability of the prover work has been supported in where the prover can prove his identity with simple computing requirements [9]. They achieve their goals by using special devices such as smart cards and exploring the bilinear pairing in the verifier side. They also use zero-knowledge to increase security guarantee. The work of [10] proposes to use the isomorphic graphs for performing and evaluating *Zero-Knowledge Proof Authentication* (ZKPA).

Our proposed scheme avoids using the time-consuming bilinear pairing technique in the verifier side to meet with the cloud model that focuses on the pay as you go foundation. We use Zero-Knowledge and one password for each user's login request to achieve the authentication goal anonymous (username and password). The purpose of the proposed scheme is to attain a high level of security, low cost, and height performance.

Viet et al. [11] 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. Firstly, PIR requires from

the server to be passed a whole database to detect user. Secondly, it cannot resist on-line guessing attacks.

We presented scheme to overcome these problems; that suits of cloud environment. DW issues to each user and service provider important information (credential, secret parameters) to be used for authentication. Moreover, 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.

The smart card-based authentication schemes [12, 13] implemented 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.

Our proposed scheme does not need middleware to save software applications or files. Additionally, our work enjoys the feature of a password-file-free service provider and uses zero-knowledge proof with OTP to built a Cloud-based two-factor authentication scheme.

The biometric data has been used scheme in [14, 15] to confirm the digital identity of the user by using his biometric features such as face, iris, speech. However, their schemes suffer from reply attack and is limited because these schemes 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.

We presented scheme does not require any biometric technique. Our scheme needs simple interaction between the user and system. Add to that, our work does not require any cost compared with biometric systems.

Lastly, OTP token considers one of the most security products during the last five years. OTP token has used many fields such as PC, PDA, and Cloud based token [16, 17]. It is suffered from many drawbacks such as high token cost and malicious attacks (e.g. MITM Seed-tracing attack).

We present a secure encryption scheme and use it to implement a cloud-based password by using simple cryptographic primitives. Our proposed scheme is armed by high-security level, can withstand the above-mentioned malicious attacks as well, and does not require any cost compare with OTP token.

We compare security properties of our proposed scheme with ones of four authentication schemes, including Yan et al. [18], Das et al. [19], Chien et al. [12], and Pathan et al. [20]. Table 1 is described comparison of security properties.

IV. OUR PROPOSED SCHEME

In this section, we present a new password authentication scheme and privacy-preservation for cloud environments. The following notations in Table 2 will be used throughout our scheme.

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 phases—setup, registration, and authentication.

Setup and registration phases are executed only once, and the authentication phase is performed whenever a user wishes to login. In the setup and registration phases, 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 the service provider and each user in the secure channel. We can describe this step as follows.

Table. 1 Comparison of authentication schemes.

Feature	Our Scheme	Yan et al. [18]	Das et al. [19]	Chien et al. [12]	Pathan et al. [20]
C1	Yes	Yes	Yes	Yes	Yes
C2	Yes	Yes	Yes	No	No
C3	Yes	Yes	No	No	No
C4	Yes	Yes	No	Yes	Yes
C5	Yes	No	No	No	No
C6	Yes	No	No	No	No

C1: Freely chosen password; C2: User anonymity; C3: Session key agreement; C4: Mutual authentication; C5: Revocation; C6: Without synchronized clocks**.**

Table. 2 Notations of our proposed scheme.

Symbol	Definition
Un_i	The username of a user U_i
Pw_i	The password of a user U_i which saved inside DW
ZPK	Public system parameter
g_i, k_i	Shared keys which will be available to both U_i and SP
$R_i, R_i^{'}$	The agreement keys between U_i and SP for each user's login request where R_i equals $R_i^{'}$.
x_i	The hash of the password Pw_i
f_i, y_i	These are used for the verifier in the computing of the proof of knowledge
$H(.)$	Hash function
$Enc(.)$	symmetric key encryption
$\|\|$	concatenation function
C	Counter is incremented for each login user's session
$Pw_i^{'}$	One-Time password
$Pw_i^{'}, Pw_i^{''}, C', v_i,$ $E_{i1}, E_{i2}, E_{i3}, E_i^{'}$	Other miscellaneous values which are used in the verification
α	The random token generated for each login attempt

DW sets up $n = pq$; Where both p and q are two large primes and selects $(g_i, k_i \in Z_n^*)$. DW uses a cryptographic hash function $H()$, and symmetric key encryption $Enc(.)$. Then, DW computes important information (f_i, y_i, x_i) ; Where $x_i = H(pw_i)$, $y_i = g_i^{x_i} \bmod n$, $f_i = g_i^{un_i} y_i$. The secret system parameters contain $ZKP = (g_i, k_i, n, H\ .\ , Enc())$. Briefly, DW supplies U_i and SP by important information as follows.

1) $DW \rightarrow U_i : ZKP, f_i, y_i, x_i$

2) $DW \rightarrow SP : Un_i, H\ y_i, k_i\ , ZKP$

U_i encrypts his credential (ZKP, y_i, f_i, x_i) by using private key pk_i , i.e., $Enc_{pk_i}\ y_i, f_i, x_i, ZKP$, he is computed his private key by composing between pk_i and x_i , private key is $pk_i = pw_i \,||\, x_i$, where $||$ means concatenation function. Then, U_i saves his credential file to his preferred storage such as USB, iPhon, iPod. After that, the user may use the authentication phase to login. The 2FA authentication session is qualified as follows (see Fig. 2).

1. $U_i \rightarrow SP : Pw_i^{'}, E_{i1}$. User uses decryption function $Dec_{pk_i}\ y_i, f_i, x_i, ZKP$ to decrypt his credential file by pk_i , generates a random number $r_i \in Z_n^*$, and computes the pair $(Pw_i^{'}, R_i)$ as follows.

- The one-time password is established as follows $Pw_i^{'} = H(H\ y_i, k_i\ \,||\, r_i)$.

- The authenticated session key is computed as follows $R_i = Pw_i^{'} \,||\, k_i$.

After that, he computes $E_{i1} = Enc_{R_i}(r_i, Un_i)$ and then sends $FirstFactor = (Pw_i^{'}, E_{i1})$ to SP .

2. $SP \rightarrow U_i : Enc_{R_i}(\alpha)$. Upon receiving the login request Firs Factor, SP performs the following computations:

- Retrieve (r_i, Un_i) by decrypting $Dec_{R_i}(E_{i1})$, and compute $R_i^{'} = Pw_i^{'} \,||\, k_i$, $Pw_i^{''} = H(H(y_i,\ k_i) \,||\, r_i)$.

- Check whether $Pw_i^{''} = Pw_i^{'}$. Aborts if not valid. When SP detects the identity of U_i , will provide U_i by $\alpha \in Z_n^*$ which generates randomly for each login attempt of U_i .

3. $U_i \rightarrow SP : Enc_{R_i}(y_i, t_i, v_i, z_{x_i}, w_{x_i})$. U_i decrypts $Dec_{R_i}(\alpha)$ and computes $t_i = g_i^{r_i}$, $v_i = H(y_i, t_i,\ f_i, \alpha)$, $z_{x_i} = r_i + v_i x_i$ and $w_{x_i} = x_i - v_i x_i$. Then, he sends $SeconFactor = Enc_{R_i}(y_i, t_i, v_i, z_{x_i}, w_{x_i})$ to the user.

4. $SP \rightarrow U_i : E_{i2}$. SP computes as follows.

$Dec_{R_i^{'}}(y_i, t_i, v_i, z_{x_i}, w_{x_i}), t_i^{'} = (y_i)^{v_i} g_i^{z_{xi}}$, and $f_i^{'} = (y_i)^{v_i} g_i^{Un_i} g_i^{w_{x_i}}$.

- $v_i^{'} = H(y_i, t_i^{'}, f_i^{'}, \alpha)$.

Finally, SP checks, if $v_i^{'}$ equals v_i , U_i is an authorized user, so SP computes $E_{i2} = Enc_{R_i^{'}}(t_i^{'} + f_i^{'})$ and then sends it to U_i . The mathematical proofs (1, 2) demonstrate how SP obtains the secret parameters $(t_i^{'}, f_i^{'})$ of U_i .

5. U_i will ensure the validity of SP by computing $E_{i3} = Enc_{R_i}(t_i + f_i)$. After that, he checks whether $E_{i3} = E_{i2}$ or not. If the result of the comparison is true, SP is a valid server otherwise it is an impersonator party.

A. Security Analysis

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

Proposition 1. Our scheme can prevent a replay attack.

Proof. An adversary performs a replay attack by eavesdropping the login message which sent by a rightful user to the server. Then an adversary reuses this message to impersonate the user when logging into the system in a next session. In our proposed scheme, each new login request should be identical with SP's keys; $(Pw_i^{''}, v_i^{'})$ therefore, an adversary cannot pass any replayed message to the SP's verification. Moreover, our work can resist this attack without synchronization clocks. So, our scheme depends on random nonce r_i instead of timestamp. Therefore, an adversary fails to apply this type of attack.

Proposition 2. Our scheme can resist the forgery attack.

Proof. If an adversary tries to impersonate U_i , he should be Proof. If any adversary tries to impersonate U_i , he should be accessed a valid login message by using two-factor $(Un_i, Pw_i^{'}, E_{i1}, Enc_{R_i}(y_i, t_i, v_i, z_{x_i}, w_{x_i}))$. An adversary does not possess any idea about (y_i, k_i) to compute first factor and he also cannot get $(g_i, x_i, y_i, r_i, R_i)$ to obtain second factor. Lastly, an adversary will fail to forge a valid login message and therefore, cannot use a forgery attack.

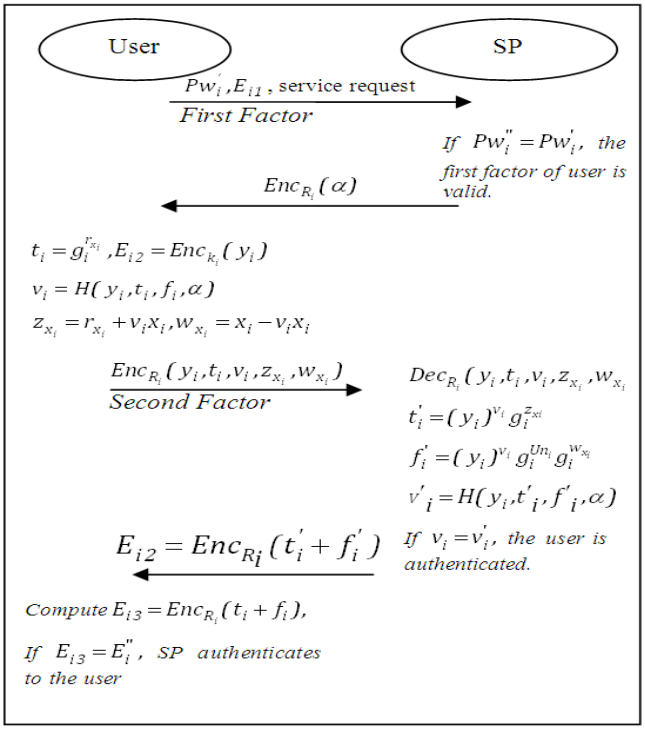

Figure 1. Mutual Authentication of our proposed scheme

Proposition 3. Our scheme can prevent an insider attack.

Proof. In our work, when U_i wishes to register with a SP for remote-access services, has to submit $(Un_i, H\ y_i, k_i\ , ZKP)$

instead of (Un_i, Pw_i). Due to the utilization of zero-knowledge (ZKP, y_i), Salt key (k_i), authenticated sessions key (R_i) and one-way hash function H, they are considered practically impossible for SP to gain the user's password Pw_i from the hashed value $H\ y_i, k_i$. Furthermore, the values of a pair ($Pw_i^{'}, R_i$) are generated once time for each user's login request. Therefore, even the service provider does not know the user's password. Obviously, our scheme can preclude the insider attack.

Proposition 4. Our scheme can resist the parallel-session attack.

Proof. In our proposed scheme, an adversary fails to impersonate a legal user by generating a correct login message in another on-going executed from the reliable execute since SP's replay message $E_i^{'} = Enc_{R_i^{'}}(t_i^{'} + f_i^{'})$ is encrypted with $R_i^{'}$, which depends on secret key k_i. This key just exists in SP and U_i. The adversary cannot compute $Pw_i^{'} = H(H\ y_i, k_i\ //\ r_i)$, and $R_i = Pw_i^{'}\ //\ k_i$ that generated once time for each user's login request. Therefore, our work can resist the parallel-session attack.

Proposition 5. Our scheme can resist a reflection attack.

Proof. This type of attack is happen, when a valid user submits his login message to the service provider, the adversary tries to catch user's message and sends it (or an updated version of the message) back to the same user. In our proposed scheme, the adversary fails to cheat the service provider since he can decrypt the pair $(E_{i1}, Enc_{R_i}(y_i, t_i, v_i, z_{x_i}, w_{x_i}))$ that sent from the user via two-factor authentication. The service provider has a secret key k_i to compute $R_i^{'}$. Thus, our proposed scheme prevents the reflection attack.

Proposition 6. Our scheme can resist the off-line guessing attack.

Proof. An adversary tries to record the transmission data of honest users' requests and then compares with set words to launch attacks. He picks one password after another from these words, calculates the communication values with the selected password and then looks for a similarity in the registered data. If a similarity is found, then the selected password is the legal password. In our proposed scheme, the service provider will not replay unless he is checked of the honest of the user. An attacker is not able to obtain both $Pw_i^{''}$ and $v_i^{'}$ since he does not have the necessary keys for decrypting this secret information. Our scheme is immune against this type of attack, even if an adversary eavesdrops two-factors on the communication between U_i and SP. He cannot gain any advantages because fails to access $(g, k_i, H(Pw_i, k_i), r_i)$ does not possess R_i to decrypt $Enc_{R_i}(y_i, t_i, v_i, z_{x_i}, w_{x_i})$, and $Pw_i^{'}$ generates once time for each user's login request. Therefore, an adversary cannot launch an off-line guessing attack.

Proposition 7. Our scheme can resist the MITM Seed-tracing attack.

Proof. Often, OTP token schemes are suffered from the fixed seed-key. In the moment, scheme generated a fixed pseudo random series style. An adversary may possess the opportunity to hunt the seed-key if he obtains enough sequences of OTP values from the same token. This type attack is called MITM Seed-tracing attack. Furthermore, the user may be threatening by Shoulder-surfing attack. This attack may be happened while the user is entering the password via the login phase or using the OTP token. The adversary can get user's secretive information during the secret attack without user knowing anything. Then, he can trace out Seed-key, if he gets enough sequences of OTP codes. Our proposed scheme resists this type of attacks. The adversary does not gain any advantages from his attempts to detecting seed password; It cannot obtain the values of (ZKP, g_i, y_i, k_i) to perform its malicious attack. The adversary must perform the following operations to get the seed of password.

- He must guess the values of $h(y_i, k_i)$, $y_i = g_i^{x_i} mod\ n$ and $x_i = H(pw_i)$.
- The adversary cannot access to credential file that saves in extra-device by user or even the adversary comprises the service provider; He cannot get to the password.

Obviously, our proposed scheme can withstand the MITM Seed-tracing attack.

Proposition 8. Our scheme can resist the stolen-verifier problem.

Proof. An adversary who plunders the user's verifier may employ the stolen verifier to apply an impersonation attack or a denial-of-service attack. In our proposed scheme, an adversary cannot launch these attacks for the following reasons.

The values of first factor ($Un_i, Pw_i^{'}, E_{i1}$) and the verifier $Pw_i^{''}$ must be known before an adversary has the ability to verify a user's request, which is not probable to occur. Also, the values of second factor ($Enc_{R_i}(y_i, t_i, v_i, z_{x_i}, w_{x_i})$) and verifier values (t'_i, v'_i, f'_i) should be available before an adversary can verify a request in our proposed scheme.

The values ($Un_i, Pw_i^{'}, E_{i1}$) must be available at the same time to generate first factor and the same case of second factor to create a correct request, which is, infeasible for the adversary. Therefore, our proposed scheme can prevent the stolen-verifier problem.

Proposition 9. Our proposed scheme can support perfect forward secrecy.

Proof. This security characteristic means that the exposure secret keys such as secret key R_i does not lead to detect the secrecy of the agreed keys in setup phase. In our work, perfect forward secrecy is ensured since the secret key is used once time to establish the authenticated session R_i. Even if the adversary knows the secret key R_i, he cannot use it to the next login request session. So, this key becomes invalid when a user leaved the system. Additionally, R_i alone is not enough to reveal of the long-term secret key material (e.g. secret key k_i and user's password Pw_i). Hence, the proposed scheme can provide perfect forward secrecy.

Proposition 10. The proposed scheme can provide mutual authentication.

Proof. This feature means that an adversary cannot impersonate a legal user to SP, and vice versa. In our work,

mutual authentication of U_i to SP is by two factors: first factor ($Un_i, Pw_i^{'}, E_{i1}$) and second factor ($Enc_{R_i}(y_i, t_i, v_i, z_{x_i}, w_{x_i})$). Without the expertise of a correct ($y_i, k_i, g_i, x_i, f_i, r_i, R_i$), an adversary cannot generate two factors that make SP accept his request. Authentication of SP to U_i is by $E_i^{'}$. An adversary distinctly is not able to get correct $t_i^{'}, f_i^{'}, k_i$ and in turn $Enc_{R_i^{'}}(t_i^{'} + f_i^{'})$ that will be approved by U_i. Hence, our work can achieve mutual authentication.

Proposition 11. The proposed scheme can provide Unlinkability.

Proof. It means that SP is not able to link variant logins by the same user (U_i). We embed this feature in our work by changing the values ($Pw_i^{'}, E_{i1}, t_i, v_i, z_{x_i}, w_{x_i}, R_i$) each time U_i tries to login the system. Therefore, the proposed scheme can support unlinkability.

Proposition 12. The proposed scheme can provide user anonymity.

Proof. If an adversary eavesdrops on the user's login request via first and second factors, he fails to get the user's identity from both factors ($E_{i1}, Enc_{R_i}(y_i, t_i, v_i, z_{x_i}, w_{x_i})$) which encrypted by R_i. Hence, it is hard for the adversary to detect a user's identity. Explicitly, our proposed scheme can supply user anonymity.

Proposition 13. Our proposed scheme can provide revocation.

Proof. In case of lost or stolen of user's prefer storage such as USB, U_i will present request to the SP for its revocation by pushing his first factor ($Pw_i^{'}, E_{i1}$). SP decrypts E_{i1} and ensures whether $Pw_i^{'} = Pw_i^{''}$. If the result is valid, SP deletes registration credential ($Un_i, H\ y_i, k_i\ , ZKP$) of the user registration table. Lastly, the user can change his username and password by re-performing the registration phase. Additionally, an adversary cannot get any benefits from stealing user's extra storage because he cannot be tolerated to decrypt credential file which requires from an adversary to compute $pk_i = pw_i \,||\, x_i$. Obviously, our proposed scheme can provide revocation.

Table. 3 Estimation Parameters.

Symbc	Definition
T_H	Time processing of a hash function
T_{EXP}	Time processing of an exponentiation operation
T_{ENC}	Time processing of symmetric encryption operation
T_{DEC}	Time processing of symmetric decryption operation
T_{Opr}	Time processing of mathematical operations such as multiplication, addition and subtraction
$T_{\|\|}$	Time processing of concatenation function

B. Performance Investigation

In this section, we conduct several experiments for gauging the efficiency and the effectiveness of our work. The Unsurprisingly, Fig. 3 shows that the response time is increased linearly with the user's number. Furthermore, the average time for the login and authentication phase of our work is equal to 0.0385 seconds for each user who indicates the high speed of our solution. estimation parameters are declared in Table 3. The time requirement of our scheme is brief in Table 4.

We test the effectiveness in terms of authentication accuracy. The efficiency of our work has been tested in terms 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.

Table. 4 Performance of our proposed scheme.

Phase	DW	User	SP
Setup & Registration	$2T_{Exp} + 2T_H$	$T_{Enc} + T_{\|\|}$	--
Login	--	$T_{Dec} + T_{Enc} + T_H + T_{\|\|}$	--
Mutual Authentication	--	$T_{Dec} + T_{Exp} + T_H + 4T_{Opr} + 2T_{Enc}$	$2T_{Dec} + 4T_{Exp} + 2T_H + 4T_{Opr} + 2T_{Enc} + 2T_{\|\|}$
Total	$2T_{Exp} + 2T_H$	$2T_{Dec} + T_{Exp} + 2T_H + 4T_{Opr} + 2T_{\|\|} + 4T_{Enc}$	$2T_{Dec} + 4T_{Exp} + 2T_H + 4T_{Opr} + 2T_{Enc} + 2T_{\|\|}$

V. CONCLUSION

In this paper, we have presented an efficient 2FA-based mutual authentication, unlinkablity and one-time password with a zero-knowledge scheme for a cloud environment.

Our proposed scheme assumes a new setting where users keep their passwords far away from the service provider in the cloud. This feature has been gained a good chance to service provider to increase time processing. Furthermore, our proposed scheme resists insider attacks, MITM attacks, forgery attacks, replay attacks, off-line attacks, and parallel session attacks. Also, our work has many virtues, including freely chosen password, user anonymity, mutual authentication, session key agreement and does not require the synchronized clock. In performance evaluation, our scheme has been proven to obtain strong security with lower communion cost than previous works.

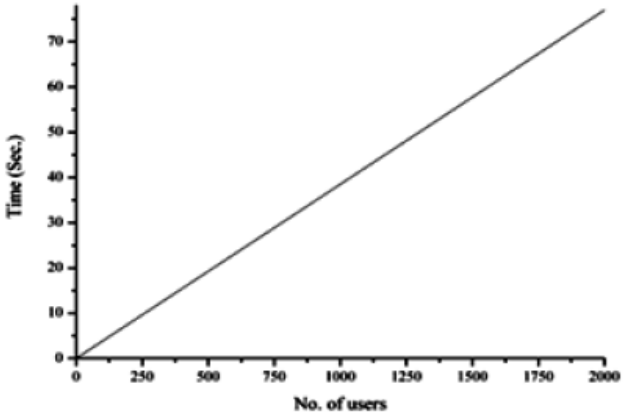

Figure. 3. Average time of login and authenticating phases for our proposed scheme

REFERENCES

[1] S. Subashini, V. Kavitha, A survey on security issues in service delivery models of cloud computing, Journal of Network and Computer Applications 34(1) (2011)1-11.
[2] D. Zissis, D. Lekkas, Addressing cloud computing security issues, Future Generation Computer Systems 28 (2012) 583-592.
[3] Md. T. Khorshed, A.B.M. S. Ali, S. A. Wasimi, A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing, Future Generation Computer Systems 28 (2012) 833-851.
[4] H-Y Chien, and J-k Jan, Robust and Simple Authentication Protocol, The Computer Journal 46(2) (2003) 193-201.
[5] S. Shin, K. Kobara, H. Imai, A Secure Construction for Threshold Anonymous Password-Authenticated Key Exchange, IEICE Transactions on Fundamentals E91-A(11) (2008) 3312-3323.
[6] A. A. Yassin, H. Jin., A. Ibrahim, W. Qiang, D. Zou, A Practical Privacy-preserving Password authentication Scheme for Cloud Computing, Paper presented at the Proceedings of IEEE IPDPSW, May 21-25, Shanghai, China,2012.
[7] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof-systems, Paper presented at the Proceedings of ACM STOC, May 6-8, New York, USA,1985.
[8] L. Lu, J. Han, Y. Liu, L. Hu, J. Huai, L. M. Ni, J. Ma, Pseudo Trust: Zero-Knowledge Authentication in Anonymous P2Ps, IEEE Transactions on Parallel and Distributed System 19 (2008) 1325-1337.
[9] Q. Jean-Jacques, M. Muril, M. G. Louis, M. Annick, G. Anna, G. Soazig, T. Berson, How to Explain Zero-Knowledge Protocols to Your Children, in: Advances in Cryptology, in: Lecture Notes in Computer Science, vol. 435, 1989, pp. 628-631.
[10] D. Anshul, S. Roy, ZKP-based Identification Scheme for Base Nodes in Wireless Sensor Networks, Paper presented at the Proceedings of ACM SOAC, Mar 13-17 , Santa Fe, New Mexico, USA, 2005.
[11] D. Q. Viet, A. Y., H. Tanaka, Anonymous Password-Based Authenticated Key Exchange, in: International Conference on Cryptology in India, in: Lecture Notes in Computer Science, vol. 3797, 2005, pp. 233-257.
[12] S. Jeon, H-S Kim, M-S Kim, Enhanced Biometrics-based Remote User Authentication Scheme Using Smart Cards, Security Engineering 8(2) (2011) 237-254.
[13] H-Y Chien, J-K Jan, Y-M Tseng, An Efficient and Practical Solution to Remote Authentication, Computers and Security 21(4) (2002) 372-375.
[14] C-T Li, M-S Hwang, An efficient biometrics-based remote user authentication scheme using smart cards, Journal of Network and Computer Applications 33(1) (2010) 1-5.
[15] C-T Chu, C-H Chen, The Application of Face Authentication System for Internet Security Using Object-Oriented Technology, Journal of Internet Technology 6(4) (2005) 419-425.
[16] S. Mizuno, K. Yamada, K. Takahashi, Authentication Using Multiple Communication Channels, Paper presented at the Proceedings of ACM DIM, Nov. 11 , Fairfax, VA, USA, 2005.
[17] O. Arasatnam, S. Boardman, 2010, Security for the cloud and SOA retrieved 8 May 2011, from http://www.opengroup.org/soa/projects/security.htm.
[18] J. Yang, Z. Zhang, A New Anonymous Password-Based Authenticated Key Exchange Protocol, in: 9th International Conference on Cryptology in India, in: Lecture Notes in Computer Science, vol. 5365, 2008, pp. 200-212.
[19] M. L. Das, A.Saxena, V. P. Gulati, A dynamic ID-based remote user authentication scheme, IEEE Transactions on Consumer Electronics 50(2) (2004) 629-631.
[20] A-S K. Pathan, C-S Hong, T. Suda, A novel and efficient bilateral remote user authentication scheme using smart cards, Paper presented at the Proceedings of IEEE ICCE, Jan.10-14, Las Vegas, NV, USA, 2007.