2015

AATCT: Anonymously Authenticated Transmission on the Cloud with Traceability

Maged Ibrahim, *Helwan University*

AATCT: Anonymously Authenticated Transmission on the Cloud with Traceability

Maged Hamada Ibrahim
Department of Electronics, Communications and Computers Engineering
Faculty of Engineering, Helwan University
1, Sherif St., Helwan, P.O. Box 11792, Cairo, Egypt

***Abstract*—In Cloud computing, anonymous authentication is an important service that must be available to users in the Cloud. Users have the right to remain anonymous as long as they behave honestly. However, in case a malicious behavior is detected, the system – under court order – must be able to trace the user to his clear identity. Most of the proposed authentication schemes for the Cloud are either password-based authentication schemes that are vulnerable to offline dictionary attacks, or biometric-based authentication schemes that take a long time of execution specially in case of high security requirements. In this paper, we propose an efficient and secure scheme to non-interactively authenticate the users on the Cloud to the remote servers while preserving their anonymity. In case of accusations, the registration authority is able to trace any user to his clear identity. We avoid using low entropy passwords or biometric mechanisms, instead, we employ pseudonym systems in our design. The computation complexity and storage requirements are efficient and suitable to be implemented on smart cards/devices. Our proposed scheme withstands challenging adversarial attacks such as, stolen databases attacks, databases insertion attacks, impersonation attacks, replay attacks and malicious users/servers collaboration attacks.**

Keywords*—*Cloud computing; anonymous transmission; pseudonym systems; smart cards; mobile devices; authentication; IT security

I. Introduction

Cloud computing paradigm is becoming an interesting new technology in the recent years with companies of all sizes accessing the Cloud. As cost efficiency, unlimited storage, backup and recovery, automatic software integration, easy access to information stand out as advantages, security services still need attention. Efficient security services for the Cloud is a major demand for all organizations. The Cloud has many security issues as it coordinates many technologies such as networking, virtualization, memory and database management. In Cloud security, authentication is the most important factor with the need for well-defined authentication strategies. One of the first steps toward securing an IT system is to verify the authentic identity of its users [1]. Authentication is generally referred to as a mechanism that establishes the validity of the claimed identity of the individual. There are basically four approaches to achieve authentication services: *Something known* (e.g. cryptographic keys, passwords, PINs, etc.), *Something possessed* (e.g. tokens, devices or cards, etc.), *Something an individual is* (e.g. fingerprints or voice patterns, face, eye retina, etc.), *Something an individual does* (e.g. history of Internet usage). On the other hand, users' identity privacy is also expected in Cloud services. If the access to a Cloud discloses a user's real identity, the user could still be unwilling to accept this issue. Thus, the user authentication without identifying the real identity, also called anonymous authentication [2], [3] is required.

In order to preserve users privacy and allow anonymous authentication/access in a Cloud, users can anonymously authenticate themselves as part of authorized users/groups to the Cloud provider (remote server). Users can anonymously access and modify resources. The encrypted data stored by a user can be decrypted by other members of the same group. Anonymous authentication can also be used in other scenarios such as E-DRM, E-commerce, E-voting, E-library, E-cash, E-auctions as well as some medical applications, and mobile agent applications [4], [5], [6], [3], [7], [8].

The end users do not want to be classified in any manner. In these examples, people may prefer to register only once (e.g. after some payment or being a member) and would like to keep their anonymity and privacy when they use these applications. Therefore, Anonymity is one of the important services that must be available to users in the digital world as long as they behave honestly. Users communication must be kept authentic and anonymous unless malicious behaviors are detected. In this case the accused user's clear identity must be traced and revealed by the system to solve accusations. In the Cloud, anonymity and traceability are two important services, yet, achieving a satisfactory security level for both of them with acceptable complexity is not an easy task due to the contradicting requirements: anonymous transmission must not be traceable by any individual while if a transmission is traceable, then anonymity is threatened.

Many of the previous contributions in the area of authenticating remote users to remote servers in the Cloud are *password-based authentication* protocols which incorporates a user's password (text, graph or picture) in the online authentication process for login and establishing a session key for authenticated transmission between the remote user and the remote server. Such protocols are always vulnerable to offline dictionary attacks whatever the strength of the incorporated passwords since by nature, passwords have to be memorable and hence have very low entropy. On the other hand, many of them do not consider anonymity and traceability of malicious users. Moreover, none of these protocols consider the non-repudiation service where a malicious user and/or server brought to the court cannot deny the transmission. Conventional digital signature schemes with certified public/private

key pairs indeed realize the non-repudiation service, yet the transmission is not anonymous because the certificates of the public verification key incorporates the clear identity of the user. Servers and authorities have to store the users' public keys side by side with their clear identities. Many other schemes are biometric-based schemes. Biometric-based authentication requires long execution time and their security level is always constrained by time complexity. Also, the security proofs for such schemes are heuristic. Most of the previously proposed schemes are password-based, biometric based or hybrid of both. These protocols do not preserve anonymity and do not guarantee that a user (in case of a raised dispute) will not be able to deny the transmission. Conventional digital signatures will not help in this case since they are not anonymous.

When anonymous and authenticated transmission is considered, Group signatures (GS) come to play [9], [10], [11], [12]. This cryptographic tool originally introduced in [13] allows members belonging to a group to sign messages on behalf of the group such that, the signature verifier (whether a group member or a non-member) is able to check that the signature is a valid group signature but cannot trace the identity of the signer. In case of a dispute, the trusted authority (group manager) can trace the identity of the signer.

Ring signatures introduced in [14] and further studied in [15] and [16] do not require any group manager to form a group. For signature generation, every user builds a set of public keys that includes his public key and the public keys of other users. A generated signature does not reveal the public key of the signer, but a set of public keys of all possible signers. Therefore, ring signatures cannot be used for a direct communication between a verifier and a signer. Additionally, ring signatures provide unconditional anonymity, i.e., no party can reveal the signers identity. Although ring signatures have many cryptographic applications, they are not suitable for our system since traceability is impossible.

Pseudonym systems were introduced in [17] as a way of allowing a user to work effectively but anonymously with multiple organizations. The author suggests that each organization may know a user by a different pseudonym. These are unlinkable such that two organizations cannot combine their databases to build up a dossier on the user. Nonetheless a user can obtain a credential from one organization using one of his pseudonyms and demonstrate possession of the credential to another organization without revealing his pseudonym to the second organization [18]. One may view pseudonyms as a blinded version of the users clear identities.

In our construction in this paper, we use ideas from [19], [20], [21], [22], [17], [18] and proofs of knowledge primitives from [23], [24] to devise an efficient and secure message authentication scheme to allow users to communicate anonymously with the remote servers on the Cloud in an authenticated way while in case of a dispute, the user can be traced to his clear identity to solve accusations. Our scheme is of low complexity so that it is suitable to be implemented on devices with limited resources. The authentication phase in our scheme is non-interactive, i.e. in one-move, a user is able to establish a session key with the remote server.

Paper organization: This paper is organized as follows: In section II, a study of previous and related work in the field is presented. Section III describes the motivations behind the work in this paper and also our contribution. In section IV, we describe the cryptographic tools used to construct our scheme. Our assumptions and network model are given in section V. The concrete description of our AATCT scheme is presented in section VI. The security of the proposed AATCT scheme is analyzed in section VII. The efficiency of the scheme is evaluated in section VIII. Discussions and suggested improvements are given in section IX. Finally, the conclusions of our work are given in section X.

II. Related Work

Password Authentication System (PAS) [25] for Cloud Environment uses graphical passwords. Graphical-based password techniques are developed as a potential alternative to text-based techniques, supported partially by the fact that humans can remember images better than text. Psychologists have confirmed that images are more memorable and usable than text. However, graphical passwords still hard to manage and store, still of low entropy and for high security levels requires a long time for execution and a huge amount of storage. Thus, they are also constrained by time and storage complexities. Yet, graphical passwords could be fine for securing personal devices.

Multi-dimensional password generation technique for the purpose of accessing Cloud services [26] considers multiple input parameters of Cloud paradigm referred to as multidimensional passwords. The multidimensional password is generated by considering the parameters of Cloud paradigm such as: vendor details, consumer details, services, privileges and confidential inputs such as logos, images, textual information and signatures. All these dimensions combined together produce a multidimensional password. By doing so, the probability of brute force attack for breaking the password can be reduced to a large extent. It was shown that the reduction in the probability of successful hacking improves drastically with the increase in the dimension of the input. However, based on the required level of security, one can decide the dimensions for the input. Major concerns are that the processing time increases with the increase in the dimensions of input parameters.

In textual based password authentication [2] users do not need to register their passwords to a service provider. The Users are supplied with the necessary credential information from the data owner. Furthermore, to enable 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. The protocol 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 registers her/his identity (username and password) with Data Owner. Data Owner then provides public system parameters to the service provider and each user on a secure channel.

Identity-based hierarchical model (IBHM) [27] for Cloud computing is composed mainly of three levels: The top level (level- 0) represent the root private key generator (PKG). The level-1 is sub-PKGs. Each node in level-1 corresponds to a data-center (such as a Cloud Storage Service Provider) in the

Cloud computing. The bottom level (level-2) are users in the Cloud computing. In identity based hierarchal model of Cloud computing (IBHMCC), each node has a unique name, the name is the node's registered distinguished name (DN) when the node joins the Cloud storage service. The identity of a node is the DN string from the root node to the current node itself. The deployment of IBHMCC needs two modules, namely, root PKG setup and lower level setup which provides secret keys to all nodes. The IBHM does not provide anonymity service to the users.

A biometric authentication as a service on Cloud [28] uses Single Sign On/Off (SSO) property for authentication. SSO is a property of access control of multiple related, but independent software systems. The blind protocol technique reveals only the user's identity. As the protocol is based on asymmetric encryption of the biometric data, it captures the advantages of biometric authentication as well as the security of public key cryptography. During the registration process, the user enrolls with the biometric system which is provided by a Cloud, once the identity is registered his/her biometric authentication details are stored in a Cloud service provider database. The authorization details are also entered at the registration time which is then stored in encrypted format. Once authenticated, the user is redirected to the actual Cloud service for which he is authorized to use.

A 3-D password authentication system [29] combines Recognition, Recall, Tokens and Biometrics in one authentication system. The 3-D password is a multi-factor authentication scheme. It can combine all existing authentication schemes into a single 3-D virtual environment. This 3-D virtual environment contains several objects or items with which the user can interact. The type of interaction varies from one item to another. The 3-D password is constructed by observing the actions and interactions of the user and by observing the sequences of such actions. The user has the flexibility of selecting the type of authentication techniques that will be the part of their 3-D password. This is achieved through interacting with the objects that acquire information that the user is comfortable in providing and ignoring the objects that request information that the user prefers not to provide. Other schemes that are based on biometrics in establishing the authentication service are found in [30], [31], [32]. The authors in [32] proposed an authentication scheme known as Cloud cognitive authenticator (CCA). It applies one round zero knowledge protocol for authentication. CCA is an API designed for Cloud environment that integrates bio-signals, knowledge proof and Rijndaels algorithm. CCA improves security in a public Cloud by providing bi-level authentication. It also provides encryption and decryption of user identities. Electro dermal responses are used for first level authentication. The main advantage of CCA compared to other existing models is that it provides two levels of authentication combined with the encryption algorithm.

The problem with biometric-based authentication schemes is that they take a long time for execution, thus, their security level is constrained by time complexity. Also, the security level achieved by such schemes is heuristic. Finally, we recommend the reader to refer to [33] for a survey and a demonstration on the weaknesses associated with password-based authentication and why it is considered a weak link in Cloud computing technologies in general.

III. Motivations and Contribution

A. Motivations

The work in this paper is motivated by the observation that most of the previously proposed schemes for the purpose of achieving anonymous and authenticated transmission on the Cloud are password-based schemes whether these passwords are memorized or extracted from biometric patterns. Their purpose is to authenticate the users and the servers in the Cloud and establish a session key (extracted from the password) to secure the session. Passwords in general suffer from their low entropy and when they are incorporated in the communication on the link for authentication, they are always vulnerable to password guessing attacks, specially offline dictionary attacks [33]. Also there is no clear strategy how the user is traced to his clear identity in case of a dispute. Biometric-based authentication schemes require long execution time and their security level is always constrained by time complexity. Also, the security proofs for such schemes are heuristic. Most of these protocols do not preserve anonymity and do not guarantee that a user (in case of a raised accusation) will not be able to deny the session. Conventional digital signatures will not help in this case since they are not anonymous.

B. Our contribution

In this paper we devise a message authentication scheme suitable for authenticated communication on the Cloud. Our scheme avoids using passwords and biometrics in the authentication process and does not require any interaction between a user and the remote server by any means prior to the establishment of the session. The users and the remote servers interact only with the registration authority. While the communication of the user and the server is anonymous, in case of a dispute, the registration authority is able to trace the user to his clear identity and prove the transmission. In this case, a traced user cannot deny the transmission. Our scheme's computation and storage complexities are suitable for implementation on the user's smart device with limited resources and also for smart card implementation. In our scheme, the user is able to setup a session key with the remote server in a one move non-interactive way. The computations required by the user can be performed offline. Our scheme withstands challenging adversarial attacks such as, stolen databases attacks, databases insertion attacks, impersonation attacks, replay attacks and malicious users/servers collaboration attacks.

IV. Cryptographic Primitives

In this section we describe the cryptographic primitives used in building our AATCT scheme.

A. Diffie-Hellman problem (DHP)

Let p and q be two large primes such that $q|p-1$ that is there is an integer k satisfying $p = kq + 1$. Pick $a \in_R Z_p^*$ and compute $g = a^k \mod p$. If $g \neq 1$ then g is a generator of order q in Z_p. Now pick $x \in_R Z_q^*$ where $|x| = |q|$ and compute $y = g^x \mod p$. Given (q, p, g, y) it is computationally infeasible to compute $x = \log_g y$.

B. Computational Diffie-Helman problem (CDHP)

Let (p, q, g) be as above. Pick two large integers $a, b \in_R Z_q^*$ and compute $A = g^a \mod p$ and $B = g^b \mod p$. Now given (q, p, g, A, B), it is computationally intractable to compute $g^{ab} \mod p$ without knowing a and b.

C. Decisional Diffie-Helman problem (DDHP)

Let (p, q, g) be as above. Pick three large integers $a, b, r \in_R Z_q^*$ and compute $A = g^a \mod p$ and $B = g^b \mod p$. Now given (q, p, g, A, B), it is computationally intractable to distinguish $g^{ab} \mod p$ from $g^r \mod p$ without knowing a, b and r.

D. Proof of equality of two discrete logarithms

We review the protocol of [23], [24] and also in [22] that is believed to be a zero knowledge proof of equality of two discrete logarithms. In this protocol, the public parameters are two large primes p and q such that $q|p-1$, two elements $\alpha, \beta \in Z_p^*$ and the two quantities $G_1, G_2 \in Z_p^*$. The prover $(\mathcal{P})$ proves to a verifier $(\mathcal{V})$ that he knows $x \in Z_q^*$ such that $G_1 = \alpha^x \bmod p$ and $G_2 = \beta^x \mod p$. The protocol is as follows:

- $\mathcal{P} \rightarrow \mathcal{V}$: Choose $r \in_R Z_q^*$ and send $(A = \alpha^r \bmod p$, $B = \beta^r \bmod p)$.
- $\mathcal{V} \rightarrow \mathcal{P}$: Choose $c \in_R Z_q^*$ and send c.
- $\mathcal{P} \rightarrow \mathcal{V}$: Compute and send $y = r + cx \bmod q$.
- $\mathcal{V}$: Check that $\alpha^y = AG_1^c \bmod p$ and $\beta^y = BG_2^c \bmod p$.

The above protocol can be made non-interactive (we denote it, $\Pi_{LogEq} \leftarrow P_{LogEq}(\alpha, \beta, G_1, G_2, x)$) using a sufficiently strong hash function $\mathcal{H}$ and setting $c = \mathcal{H}(A, B)$. The NIZK proof of knowledge protocol Π_{LogEq} becomes as follows:

- $\mathcal{P} \rightarrow \mathcal{V}$: Choose $r \in_R Z_q^*$ and send $(A = \alpha^r \bmod p$, $B = \beta^r \bmod p$, $c = \mathcal{H}(A, B)$ and $y = r + cx \bmod q$.
- $\mathcal{V}$: Check that $\alpha^y = AG_1^c \bmod p$ and $\beta^y = BG_2^c \bmod p$.

V. Assumptions and Model

The main entities in our protocol, as illustrated in Figure 1, are: The registration authority (RA), the remote server (RS) and the Cloud user U_i. The RA is assumed fully trust to manage all secret parameters in the system while the RS as well as U_i could behave maliciously and could collaborate trying to disclose the privacy of other users in the system. We assume the existence of a PKI, such that RA and each remote server RS has its own certified public/private key pair to realize authenticated and private channels among all entities. We emphasize that the users in our scheme may use the servers' certified public keys but they are not related to this PKI.

Our scheme does not incorporate any users' passwords in the authentication process. Yet, personal PIN may be used by the user to secure his own smart card. Our scheme allows any user to anonymously establish a session secret key using the remote server's public key. The users do not interact with each other or the remote server RS, they interact only with the RA for registration and setup. Then after the registration phase is completed, any registered user can communicate anonymously with any remote server in the Cloud. Finally, in the description of our protocol we focus on the anonymity of the user. Later we show how mutual anonymity is achieved.

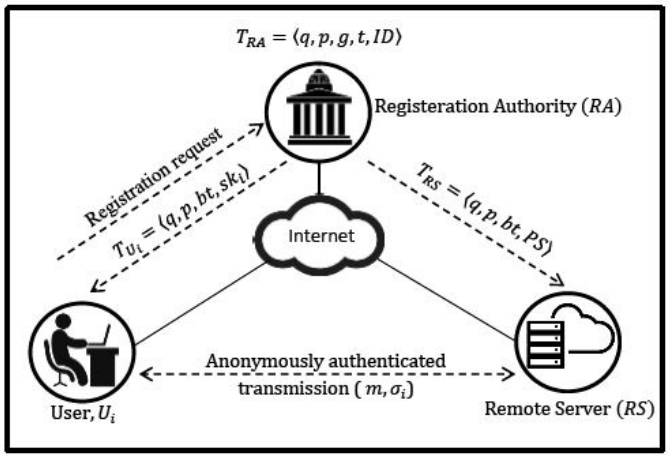

Fig. 1. *Our AATCT scheme and architecture*

VI. Concrete Description of our AATCT Scheme

In this section we give a detailed description of our AATCT. There is a registration authority RA and a remote Cloud server RS. There is a remote user U among the set of Cloud users. The RA and each remote server RS has his own certified public/private key pair to allow regular authenticated and confidential communications. The phases of our scheme are described next.

A. Initialization phase by the RA

The RA initializes the system parameters as follows:

- Picks two large primes p and q where $q|(p-1)$ and a generator g of order q in Z_p^*.
- Picks a secret tracing trapdoor parameter $t \in_R Z_q^*$ and computes its blinded version $bt = g^t \mod p$.

B. Registration phase

The RA registers a user U_i as follows:

- Picks $x_i \in_R Z_q^*$ as U_i's private key and computes $id_i = g^{x_i} \mod p$ as U_i's public identity.
- Computes U_i's pseudonym as $ps_i = (id_i)^t \mod p$.
- Parses U_i's secret key $sk_i = (x_i, ps_i)$.
- On U_i's smart card, RA installs the tuple,
$$T_{U_i} = \langle q, p, bt, sk_i \rangle$$

Let $\mathcal{ID} = \{id_1, ..., id_n\}$ be the set of the users' clear identities while $\mathcal{PS} = \{ps_1, ..., ps_n\}$ be the set of the users' pseudonyms. The RA signs and publishes to each remote server RS the tuple,

$$T_{RS} = \langle q, p, bt, \mathcal{PS} \rangle$$

The RA finalizes the registration phase by storing the tuple,

$$T_{RA} = \langle q, p, g, t, \mathcal{ID} \rangle$$

and erasing all other parameters.

Remark. In the registration phase we assumed that RA generates the private key x_i and computes the identity id_i for U_i. It is possible that U_i by himself generates his own private key, computes and sends his identity to RA. He can also compute his pseudonym $ps_i = (bt)^{x_i}$. However, in this case, U_i must provide a proof of knowledge of x_i. Notice that in this case, RA does not know the private key x_i which avoids the key escrow problem. The choice is left to the organization.

C. Authentication phase

User U_i anonymously signs a message m using his private key sk_i as follows:

- Picks a random integer r, hashes m as $H = \mathcal{H}(m, r)$ and computes $z = H^{x_i} \bmod p$.
- Generates a NIZK proof of knowledge, $\Pi_{LogEq} \leftarrow P_{LogEq}(H, bt, z, ps_i, x_i)$, which proves that $\log_H(z) = \log_{bt}(ps_i) = x_i$.
- Parses σ_i as $(r, z, ps_i, \Pi_{LogEq})$. σ_i is U_i's anonymous signature on m.

On the reception of a signature σ_i on m, RS verifies as follows:

- Parses σ_i as $(r, z, ps_i, \Pi_{LogEq})$.
- Ensures that $ps_i \in \mathcal{PS}$.
- Runs the verification algorithm, $V_{LogEq}(H, bt, z, ps_i, \Pi_{LogEq})$, if the verification failed, then reject m and abort. Else, accept σ_i as a valid signature on m.

Notice that, a remote server is able to reply by a message dedicated to a particular user U_i by simply including his pseudonym ps_i in the replied message. Moreover, ps_i is indeed a the public key of U_i that could be used to encrypt messages to U_i as will be discussed later.

D. Tracing an accused user

In case of a dispute and under court order, given a pseudonym ps_i, RA is able to trace the identity of U_i by simply computing $(ps_i)^{1/t} = id_i$ where t^{-1} is computed modulo q.

E. Establishing a session key

A user U_i anonymously establishes an authenticated session key K in a one move non-interactive way by simply setting $m = E_{pk_{RS}}(K)$ where $E_{pk_{RS}(.)}$ is an encryption using RS's public key pk_{RS}.

VII. Security Analysis

The secrecy of the tracing trapdoor parameter t is very important for retaining the anonymity service of our scheme, since it is the only parameter that can trace any pseudonym ps_i to its clear identity id_i. From the Diffie-Hellman problem (DHP), no information is revealed to a computationally bounded adversary about t from its blinded version $bt = g^t \bmod p$. The same infeasibility follows for the user's private key x_i and his clear identity $id_i = g^{x_i} \bmod p$. The remote server RS is delivered the set of pseudonyms $\mathcal{PS}$ with no information revealed about t or any of the x_i's. From the Decisional Diffie-Hellman problem (DDHP), even if a randomly permuted version of the sets $\mathcal{ID}$ and $\mathcal{PS}$ are known to an adversary, she cannot trace any $ps_i = (id_i)^t \bmod p$ to its clear identity id_i since she cannot distinguish $g^{x_i t}$ from g^r for a random r. We want to emphasize that neither $\mathcal{ID}$ nor $\mathcal{PS}$ are necessarily kept secret, only the correspondence is secret.

In the authentication phase, U_i computes $z = H^{x_i} \bmod p$ as his signature on a message m and parses this signature with his ps_i, notice that RS knows the set $\mathcal{PS}$ and hence it is easy to check whether $ps_i \in \mathcal{PS}$. Now U_i must prove to RS that his secret key x_i used to compute z is the same value in the exponent of bt to compute the ps_i and consequently U_i parses the signature with the NIZK proof of equality of discrete-log, $\Pi_{LogEq} \leftarrow P_{LogEq}(H, bt, z, ps_i, x_i)$ to prove that $\log_H(z) = \log_{bt}(ps_i) = x_i$. This proves to RS that the anonymous signer is indeed a registered user.

Since Π_{LogEq} is a zero knowledge proof of knowledge, a verifier that receives a signed message with a certain pseudonym ps_i is faced with the DHP problem to compute x_i given bt and $(bt)^{x_i}$. Given the set of identities $\mathcal{ID}$ and pseudonyms $\mathcal{PS}$ the DDHP preserves the anonymity of the signer.

From the discussion above, in case of a dispute and under court order, only RA, the holder of the tracing trapdoor parameter t, can disclose the clear identity id_i from a given pseudonym ps_i by computing $(ps_i)^{t^{-1}} = id_i$, where t^{-1} is computed modulo q.

Mutual authentication is achieved since U_i encrypts the session key K using RS's public key. Only RS with the corresponding private key is able to decrypt for K. Both entities can test the validity of K at the beginning of the session.

In the following we discuss possible adversarial attacks, the countermeasures to be taken against these attacks and how our scheme withstands them.

A. RS compromise

An adversarial compromise of RS does not threaten the security and anonymity of any user. Actually, one may have noticed that none of the parameters delivered to RS is secret.

B. RA database compromise

Beyond the tracing trapdoor parameter t, compromising the database of RA and stealing $\mathcal{ID}$ does not threaten the anonymity of any of the users without the knowledge of t.

C. Stolen RS and RA databases

Beyond the tracing trapdoor parameter t, if all other parameters are stolen by an adversary, i.e., if the adversary steals the set $\mathcal{ID}$ and the set $\mathcal{PS}$ from RA and RS, she cannot map any pseudonym ps_i to any clear identity id_j without knowing the tracing trapdoor parameter t. Given a stolen id_i and ps_i, an adversary cannot create a valid authenticated message without the knowledge of x_i.

D. Impersonation/emulation/masquerade attacks

An adversary trying to impersonate a legal user u_i by using his pseudonym ps_i will not succeed in creating the NIZK proof of knowledge without knowing his private key x_i.

E. Databases Insertion attacks

An adversary that is able to gain access to RA and RS databases is able to insert a valid pair (id_A, ps_A) as to become registered in the system. There are variety of countermeasures to withstand such attack. One solution is that, the RA signs all entries in her database using her own digital signature key. Also, each remote server RS signs each entry in the $\mathcal{PS}$ database using his own digital signature key. In this case, the adversary's insertions in the databases become invalid as these entries are not digitally signed. Another more efficient solution is to hash each entry in the database using a keyed hash (e.g. message authentication code (MAC)) and append this hash with the corresponding entry in the database. In this case an adversary – without knowing the secret key – will not be able to append a correct hash to the pair (id_A, ps_A), and hence the entry in the database is invalid.

F. Replay attacks

Like any other digital signature scheme, replay attacks are avoided by a simple association of a time-stamp mechanism. Also, one may consider random nonce and sequence numbers.

G. User compromise

The private key x_i of the user U_i is stored on his smart card which is a tamper proof device and hence an adversary will not be able to reach x_i. Any other parameter on the user's side other than x_i, if known to an adversary, does not threaten the security of this user. If a certain x_i of a user U_i is revealed to an adversary, this does not threaten other users in the system. However, for this particular user, if his x_i is revealed, he must re-register for a new private key.

H. Users-servers collaboration attacks

It is possible that several malicious minority of the users are willing to collaborate with the remote servers in order to disclose privacy of other users. Malicious users are willing to reveal their private keys x_i's, their identities $id_i = g^{x_i}$ and their pseudonyms $ps_i = g^{x_i t}$. In our scheme each user private parameters are completely independent of any other user in the system. From the CDHP/DDHP, the revealed information does not allow the collaborated entities to reveal any information about the tracing trapdoor parameter t and hence, the security of the rest of the users is preserved.

VIII. Efficiency Evaluation

To evaluate the efficiency of our scheme, we assume the standard number theoretic settings on the size of the big prime p and the small prime q where $|p|$=1024 bits = 128 bytes while $|q|$=160 bits = 20 bytes. Also, roughly we have, $|g| = |bt| = |ps_i| = |id_i| = |p|$ while $|x_i| = |t| = |q|$.

A. Complexity evaluation

A concrete evaluation of the computations and storage complexities for each party in our system is shown next. Let n be the number of registered users in the system.

1) Complexity of the RA: The RA stores the tuple, $T_{RA} = \langle q, p, g, t, \mathcal{ID}\rangle$ requiring a storage of $2|q| + 2|p| + n|p|$ which totals $(296+128n)$ bytes. On the other hand, in the registration phase, the RA computes for each user his identity id_i and his pseudonym ps_i each of which is a one modular exponentiation. In the tracing algorithm the RA performs only on modular exponentiation to reveal id_i.

2) Complexity of the RS: The RS receives the tuple $T_{RS} = \langle q, p, bt, \mathcal{PS}\rangle$ which requires a storage of $|q| + 2|p| + n|p|$ totaling $(276 + 128n)$ bytes. On the reception of a signed message from the user, the RS runs the verification algorithm by computing $V_{LogEq}(H, bt, z, ps_i, \Pi_{LogEq})$. This algorithm requires the computation of four modular exponentiations and two modular multiplications.

3) Complexity of the user: The user U_i receives and stores the tuple $T_{U_i} = \langle q, p, bt, sk_i\rangle$ where $sk_i = (x_i, ps_i)$. This tuple requires $2|q| + 3|p|$ of storage which totals 424 bytes of memory. In computing a signature $\sigma_i = (r, z, ps_i, \Pi_{LogEq})$, we ignore the hashing since it is cheap. U_i performs one modular exponentiation to compute z, two modular exponentiations, one modular multiplication and one modular addition to compute Π_{LogEq}. The storage requirements and computation complexity of our system are summarized in Table I.

4) Communication complexity: The user U_i transmission is a message m concatenated with a fixed length anonymous signature $\sigma_i = (r, z, ps_i, \Pi_{LogEq})$. We have $|\Pi_{LogEq}| = 2|p| + |q|$ in addition to $2|p|$ for z and ps_i. This totals 532 bytes of communications overheads in addition to a few bytes for r. The communication complexity of our system is summarized in Table II.

TABLE I. *Storage requirements and computations*

	Storage (in bytes)	Modulo computations		
		Exponentiations	Multiplications	Additions
RA	296+128n	3 per user	-	-
RS	276+128n	4	2	-
User	424	3	1	1

TABLE II. *Communication complexity*

	Communications (in bytes)
RA→RS	276+128n
RA→User	424
User→RS	$\|m\|$+532

B. Computation time and energy consumption

In the following, the time required by the user to prepare his authentication message is evaluated on smart cards and mobile devices. An estimation of the energy consumed by our scheme on mobile devices is also given. These are summarized in Table III and described next.

1) Computation time on smart cards: The basic method (binary method) for computing modular exponentiations is through the square-and-multiply strategy. For an k-bit exponent, this method requires $k-1$ squarings and on the average of $1.5(k-1)$ multiplications. From Table I, the user requires three modular exponentiations, one modular multiplication and one modular addition. Benchmarks on Smart-card devices [34] shows that on an Oberthur ld-one v7.0-a, one modular exponentiation of 160 bits exponent and 1024 bits modulus takes 190 ms, one modular multiplication on two 1024 bits numbers and 1024 bits modulus takes 200 ms. The modular addition and hash invocations are a negligible fraction of milliseconds. Hence, it takes the user 3(190)+200 plus few fractions of milliseconds resulting in about 800 ms to generate an authentication message. Computation time on other smart cards could be deduced from [34]. We remind the reader that these computations could be performed offline.

2) Computation time on mobile devices: An implementation of modular arithmetic on an ASUS-TF300T tablet shows that a modular exponentiation of 160 bits exponent and 1024 bits modulus takes 4 ms, one modular multiplication on two 1024 bits numbers and 1024 bits modulus takes 0.1 ms. Hence, it takes only about 13 milliseconds to generate an authentication message on a mobile device. The computation time of modular arithmetic operations on other smart phones could be found in [34].

3) Energy consumption: In this part, the energy consumed by cryptographic operations is used to evaluate the schemes. This time, we use a low-processor and 64 MB memory running Windows Mobile 5.0 for pocket pc[1]. According to PXA270, the typical power consumption of PXA270 in active is 500 mW. Therefore, using the computation time in the previous calculations, we can calculate the corresponding energy consumption. For example, if it takes 13 ms to generate the authentication message, the energy consumption is approximately 13(500/1000) = 6.5 mJ.

TABLE III. *Computation time and energy consumption*

Device	Computation time	Energy Consumption
Oberthur ld-one v7.0-a smartcard	800 ms	
ASUS-TF300T tablet	13 ms	
Pocket PC with Intel PXA270		6.5 mJ

C. Simple key management

In the proposed scheme, the key management is very simple since only the tracing trapdoor parameter t is required to be kept secret by RA. On the user's side, only his private key x_i is required to be kept secret. On the server's side, no parameter is required to be kept secret. The private keys for the PKI are already there and are not due to our scheme. Notice that, the users are not part of this PKI.

[1]http://pdf.dzsc.com/cxx/nhpxa270cxxx.pdf

IX. Discussions and Improvements

A. Users join and leave

The RA easily manage the joining of a new user to the system by running the setup phase for him, adding his new identity to the set $\mathcal{ID}$ and notify the RS with the new pseudonym. Also leaving the group (revoking a user) is as simple as erasing the user from $\mathcal{ID}$ and $\mathcal{PS}$.

B. User's embedded El-gamal public/private key pair

Recall that $bt = g^t \mod p$, where g is a generator of order q on the form $g = a^k \mod p$ for $a \in_R Z_q^*$ and $k = (p-1)/q$. We have $bt = (a^t)^k \mod p$ and hence, bt is indeed a generator of order q. Thus, the pair (x_i, ps_i) where $ps_i = (bt)^{x_i}$ could be used as the user's U_i El-gamal public/private key pair for an El-gamal cryptosystem [35]. A remote server RS may encrypt a message m for U_i as follows: Picks $\mu \in_R Z_q^*$, computes the El-gamal ciphertext $C = (A, B)$ where $A = (bt)^{\mu} \mod p$ and $B = m(ps_i)^{\mu} \mod p$. Only U_i, the holder of the corresponding private key x_i, is able to decrypt C for m where $m = B/A^{x_i} \mod p$.

C. Mutual anonymity

We focused in our AATCT scheme on the anonymity of the user since it is the most important. Although servers' anonymity to the users is much less important (sometimes is undesired), it could be achieved by treating the servers in the Cloud as a group in the same way the users were treated and assign an identity and a pseudonym sets for them. However, the storage requirements on the user's side will grow linearly with the number of communicating servers in the Cloud.

D. Further Improvements

The cryptographic number theoretic tools used in devising our scheme could be replaced with Elliptic Curve (EC) tools where in this case the storage and computation time are improved by more than 20% [36]. The security of the tracing trapdoor parameter t could be further improved by applying threshold cryptographic techniques [37], [38], [39] to distribute the trust among several entities. Although in our AATCT, the user U_i stores a tuple $T_{U_i} = \langle q, p, bt, sk_i \rangle$ where $sk_i = (x_i, ps_i)$, the user may not store ps_i preserving 128 bytes of storage. However, this requires the user to compute ps_i each time a signature is performed. On the other hand, the RA may store the set of secret keys $\{x_1, ..., x_n\}$ instead of the set of identities $\mathcal{ID}$ preserving a significant amount of space ($20n$ bytes instead of $128n$ bytes). However, although it allows key recovery, this requires countermeasures for managing the privacy of the secret keys.

E. Future work

1) Deniable transmission: Another important service that must be available to users on the Cloud is deniable transmission, which allows a user on the Cloud to escape a coercion attempted by a coercive adversary. Such an adversary approaches the coerced user after transmission forcing him to reveal all his random inputs used during encryption or decryption. Since traditional encryption schemes commits the user to his random inputs, the user is forced to reveal the true values of all his

random inputs (including the encrypted/decrypted messages and the encryption/decryption keys) which are verifiable by this coercer using the intercepted ciphertext. In this scenario, a coercer may force the user to perform actions against his own beliefs. For more information about this notion, please refer to [40], [41], [42]. A deniable encryption helps to protect the users in many applications such as E-voting, E-elections and E-auctions where coercive actions come to play as a potential threat.

2) Forward security: It would be nice if one is able to efficiently realize forward security in our scheme where the blinded identity and the user's private key is updated at regular intervals so as to provide a forward security property: compromise of the current private key does not enable an adversary to forge signatures pertaining to the past. This can be useful to mitigate the damage caused by key exposure.

3) Reducing transmission overheads and complexity: One may also work on a way to reduce the bit-length of the user's signature to reduce transmission overheads. For example, finding a way to minimize the burden of the proofs of knowledge and the modular exponentiations. Working on elliptic curves would greatly improves the bit-length of all parameters as well as the computation time.

X. Conclusions

In this paper, we proposed an efficient scheme to realize anonymous authentication for Cloud computing networks based on pseudonym systems which allows a user to non-interactively establish an authenticated channel with the remote server while a registration authority is always able to trace the user to his clear identity in case of a dispute. We avoided using passwords and biometrics in the design of our scheme. Our scheme could be regarded as an anonymous signature scheme for the Cloud where a user cannot later repudiate the transmission. We designed our system in a way that the storage requirements and computation complexity for the user is suitable for mobile devices and also for smart card implementation. A complete security analysis and efficiency evaluation was presented. Our scheme requires few milliseconds to prepare the authentication message on mobile devices and few hundred milliseconds to prepare the authentication message on a smart card. Moreover, since the scheme is non-interactive, the user can prepare this message offline. Our scheme allows a user to non-interactively establish a session key with any of the remote servers in a fully private, authenticated and anonymous way. Our scheme withstands challenging attacks such as, stolen databases attacks, databases insertion attacks, impersonation attacks, replay attacks, and malicious users/servers collaboration attacks. Finally, we discussed further possible improvements to the scheme and suggested several future work for other researchers in the field to add other services and improve the proposed scheme.

Acknowledgment

The author of this paper would like to thank the anonymous reviewers of the IJACSA for their valuable and helpful comments.

References

[1] B. Guttman and E. A. Roback, *An introduction to computer security: the NIST handbook*. DIANE Publishing, 1995.

[2] A. A. Yassin, H. Jin, A. Ibrahim, W. Qiang, and D. Zou, "Efficient password-based two factors authentication in cloud computing," *International Journal of Security and Its Applications*, vol. 6, no. 2, pp. 143–148, 2012.

[3] M. H. Ibrahim, "Noninteractive, anonymously authenticated, and traceable message transmission for vanets," *International Journal of Vehicular Technology*, vol. 2009, 2010.

[4] M. H. Ibrahim, "Secure and robust enterprise digital rights management protocol with efficient storage," *International journal on information (information-Tokyo)*, vol. 18, no. 2, pp. 625–640, 2015.

[5] M. H. Ibrahim, "Efficient robust and secure E-DRM with encrypted content search," *International journal on information (information-Tokyo)*, vol. 18, no. 6(A), pp. 2531–2546, 2015.

[6] M. H. Ibrahim, "A novel approach to fully private and secure auction: A sealed-bid knapsack auction," *International Journal of Research and Reviews in Applied Science (IJRRAS)*, vol. 9, no. 2, 2011.

[7] A. H. Soliman, M. H. Ibrahim, and A. E. El-Hennawy, "Improving security and efficiency of enterprise digital rights management," in *proceedings of the 6th IEEE International Conference on Computing, Communications and Networking Technologies (ICCCNT 2015)*. IEEE, July 2015.

[8] M. H. Ibrahim, "Secure anonymously authenticated and traceable enterprise DRM system," *International Journal of Computer Applications*, vol. 126, no. 3, pp. 1–9, September 2015.

[9] A. Kiayias and M. Yung, "Group signatures with efficient concurrent join," *Advances in Cryptology–EUROCRYPT 2005*, pp. 198–214, 2005.

[10] M. Bellare, H. Shi, and C. Zhang, "Foundations of group signatures: The case of dynamic groups," *Topics in Cryptology–CT-RSA 2005*, pp. 136–153, 2005.

[11] J. Camenisch and J. Groth, "Group signatures: Better efficiency and new theoretical aspects," *Security in Communication Networks*, pp. 120–133, 2005.

[12] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," *Advances in Cryptology–Eurocrypt 2003*, pp. 614–629, 2003.

[13] D. Chaum and E. Van Heyst, "Group signatures," in *Advances in Cryptology–EUROCRYPT'91*. Springer, 1991, pp. 257–265.

[14] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in CryptologyASIACRYPT 2001*. Springer, 2001, pp. 552–565.

[15] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," in *Information Security and Privacy*. Springer, 2004, pp. 325–335.

[16] A. Bender, J. Katz, and R. Morselli, "Ring signatures: Stronger definitions, and constructions without random oracles," in *Theory of Cryptography*. Springer, 2006, pp. 60–79.

[17] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[18] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Selected Areas in Cryptography*. Springer, 2000, pp. 184–199.

[19] M. Manulis, A. R. Sadeghi, and J. Schwenk, "Linkable democratic group signatures," *Information Security Practice and Experience*, pp. 187–201, 2006.

[20] M. Manulis, "Democratic group signatures: on an example of joint ventures," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. ACM, 2006, pp. 365–365.

[21] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," in *Information Security and Privacy*. Springer, 2004, pp. 325–335.

[22] M. H. Ibrahim, "Resisting traitors in linkable democratic group signatures." *IJ Network Security*, vol. 9, no. 1, pp. 51–60, 2009.

[23] C.-P. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[24] D. Chaum and T. P. Pedersen, "Wallet databases with observers," *Advances in Cryptology–CRYPTO92*, pp. 89–105, 1993.

[25] A. Bhavana, V. Alekhya, K. Deepak, and V. Sreenivas, "Password authentication system (PAS) for cloud environment," *International Journal of Advanced Computer Science and Information Technology*, vol. 2, no. 1, p. 29, 2013.

[26] V. Agrawa *et al.*, "Multi-dimensional password generation technique for accessing cloud services," *arXiv preprint arXiv:1207.3636*, 2012.

[27] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing," in *Cloud computing*. Springer, 2009, pp. 157–166.

[28] H. Vallabhu and R. Satyanarayana, "Biometric authentication as a service on cloud: Novel solution," *International Journal of Soft Computing and Engineering*, vol. 2, no. 4, p. 163, 2012.

[29] D. Pooja, G. Shilpi, S. Sujata, and G. Vinita, "Secured authentication: 3d password," *International Journal of Engineering and Management Sciences*, vol. 3, no. 2, pp. 242–245, 2012.

[30] A. A. Pawle and V. P. Pawar, "Face recognition system (FRS) on cloud computing for user authentication," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, 2013.

[31] P. Wang, C.-C. Ku, and T. C. Wang, "A new fingerprint authentication scheme based on secret-splitting for enhanced cloud security," 2011.

[32] L. Jivanadham, Y. Katayama, S. Komaki, S. Baharun *et al.*, "Cloud cognitive authenticator (cca): A public cloud computing authentication mechanism," in *Informatics, Electronics & Vision (ICIEV), 2013 International Conference on*. IEEE, 2013, pp. 1–6.

[33] W. Page, "http://searchcloudsecurity.techtarget.com/tip/password-based-authentication-a-weak-link-in-cloud-authentication."

[34] J. Hajny, L. Malina, Z. Martinasek, and O. Tethal, "Performance evaluation of primitives for privacy-enhancing cryptography on current smart-cards and smart-phones," in *Data Privacy Management and Autonomous Spontaneous Security*. Springer, 2014, pp. 17–33.

[35] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[36] D. Hankerson and A. Menezes, "Nist elliptic curves," in *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 843–844.

[37] M. H. Ibrahim, "Eliminating quadratic slowdown in two-prime RSA function sharing," *I. J. Network Security*, vol. 7, no. 1, pp. 106–113, 2008.

[38] M. H. Ibrahim, I. I. Ibrahim, and A. H. El-Sawy, "Fast three-party shared generation of rsa keys without distributed primality tests," in *in Proceedings of the Information Systems: New Generations (ISNG'04)*, 2004.

[39] M. H. Ibrahim, I. A. Ali, I. I. Ibrahim, and A. El-sawi, "A robust threshold elliptic curve digital signature providing a new verifiable secret sharing scheme," in *Circuits and Systems, 2003 IEEE 46th Midwest Symposium on*, vol. 1. IEEE, 2003, pp. 276–280.

[40] M. H. Ibrahim, "Efficient coercion resistant public key encryption," *International Journal of Computer Science and Security (IJCSS)*, vol. 8, no. 1, pp. 1–24, 2014.

[41] M. H. Ibrahim, "Realizing sender's deniability in public key encryption via random coins isolation," *European Journal of Scientific Research*, vol. 119, no. 2, pp. 177–187, 2014.

[42] B. Meng, "A critical review of receipt-freeness and coercion-resistance," *Information Technology Journal*, vol. 8, no. 7, pp. 934–964, 2009.