Web-Based Cloud Computing Services Using Fine-Grained Two-Factor Access Control

Mrs Farhana[1], Aaryan[2], Chinmayi Arun Bandargal[3], Joseph James[4]
Atria institute of technology

Abstract- *In this paper, we introduce a new fine-grainedtwo-factor authentication (2FA) access control system for web-based cloud computing services. Specifically, in our proposed 2FA access control system, an attribute-based access control mechanism is implemented with the necessity of both a user secret key and a lightweight security device. As a user cannot access the system if they do not hold both, the mechanism can enhance the security of the system, especially in those scenarios where many users share the same computer for web-based cloud services. In addition, attribute-based control in the system also enables the cloud server to restrict the access to those users with the same set of attributes while preserving user privacy, i.e., the cloud server only knows that the user fulfills the required predicate, but has no idea on the exact identity of the user. Finally, we also carry out a simulation to demonstrate the practicability of our proposed 2FA system.*

Keywords- Fine-grained, two-factor, access control,Web services.

I. INTRODUCTION

CLOUD COMPUTING is a virtual host computer system that enables enterprises to buy, lease, sell, or distribute software and other digital resources over the internet as an on-demand service. It no longer depends on a server or a number of machines that physically exist, as it is a virtual system. There are many applications of cloud computing, such as data sharing [22], [30], [31], [33], data storage [15], [25], [32], [45], big data management [4], medical information system [44] etc. End users access cloud-based applications through a web browser, thin client or mobile app while the business software and user's data are stored on servers at a remote location. The benefits of web-based cloud computing services are huge, which include the ease of accessibility, reduced costs and capital expenditures, increased operational efficiencies, scalability, flexibility and immediate time to market

Though the new paradigm of cloud computing provides great advantages, there are meanwhile also concerns about security and privacy especially for web-based cloud services. As sensitive data may be stored in the cloud for sharing purpose or convenient access; and eligible users may also access the cloud system for various applications and services, user authentication has become a critical component for any cloud system. A user is required to login before using the cloud services or accessing the sensitive data stored in the cloud. There are two problems for the traditional account/password-based system. First, the traditional account/password-based authentication is not privacy-preserving. However, it is well acknowledged that privacy is an essential feature that must be considered in cloud computing systems. Second, it is common to share a computer among different people. It maybe easy for hackers to install some spyware to learn the login password from the web-browser. A recently proposed access control model called *attribute-based access control* is a good candidate to tackle the first problem. It not only provides anonymous authentication but also further defines access control policies based on different attributes of the requester, environment, or the data object. In an attribute-based access control system,[1] each user has a user secret key issued by the authority. In practice, the user secret key is stored inside the personal computer. When we consider the above mentioned second problem on web-based services, it is common that computers may be shared by many users especially in some large enterprises or organisations. For example, let us consider the following two scenarios:

- In a hospital, computers are shared by different staff. Dr. Alice uses the computer in room A when she is on duty in the daytime, while Dr. Bob uses the same computer in the same room when he is on duty at night.

- In a university, computers in the undergraduate lab are usually shared by different students.

In these cases, user secret keys could be easily stolen or used by an unauthorised party. Even though the computer may be locked by a password, it can still be possibly guessed or stolen by undetected malwares.

A.more secure way is to use two-factor authentication (2FA). 2FA is very common among web-based e-banking services. In addition to a username/password, the user is also required to have a device to display a one-time password. Some systems may require the user to have a mobile phone while the one-time password will be sent to the mobile phone through SMS during the login process. By using 2FA, users will have more confidence to use shared computers to login for web-based e-banking services. For the same reason, it will be better to have a 2FA system for users in the web-based cloud services in order to increase the security level in the system.

A.Our Contribution

In this paper, we propose a fine-grained two-factor access control protocol for web-based cloud computing services, using a lightweight security device. The device has the following properties: (1) it can compute some lightweight algorithms, e.g. hashing and exponentiation; and (2) it is tamper resistant, i.e., it is assumed that no one can break into it to get the secret information stored inside.

With this device, our protocol provides a 2FA security. First the user secret key (which is usually stored inside the computer) is required. In addition, the security device should be also connected to the computer (e.g. through USB) in order to authenticate the user for accessing the cloud. The user can be granted access only if he has both items. Furthermore, the user cannot use his secret key with another device belonging to others for the access.

Our protocol supports fine-grained attribute-based access which provides a great flexibility for the system to set different access policies according to different scenarios. At the same time, the privacy of the user is also preserved. The cloud system only knows that the user possesses some required attribute, but not the real identity of the user.In the next section, we will review some related works that are related to our concept.

II.RELATED WORKS

A. Attribute-Based Cryptosystem

Attribute-based encryption (ABE) [20], [39] is the cornerstone of attribute-based cryptosystem. ABE enables fine-grained access control over encrypted data using access policies and associates attributes with private keys and cipher texts. Within this context, cipher text-policy ABE (CP-ABE) [6] allows a scalable way of data encryption such that the encryptor defines the access policy that the decryptor (and his/her attributes set) needs to satisfy to decrypt the ciphertext. Thus, different users are allowed to decrypt different pieces of data with respect to the pre-defined policy. This can eliminate the trust on the storage server to prevent unauthorised data access.

Besides dealing with authenticated access on encrypted data in cloud storage service [21], [23], [24], [27]–[29], [36], [42], [43], ABE can also be used for access control to cloud computing service, in a similar way as an encryption scheme can be used for authentication purpose: The cloud server may encrypt a random message using the access policy and ask the user to decrypt. If the user can successfully decrypt the ciphertext (which means the user’s attributes set satisfies the prescribed policy), then it is allowed to access the cloud computing service.

In addition to ABE, another cryptographic primitive in attribute-based cryptosystem is attribute-based signa-ture (ABS) [35], [38], [41]. An ABS scheme enables a user to sign a message with fine-grained control over identifying information. Specifically, in an ABS scheme, users obtain their attribute private keys from an attribute authority. Then they can later sign messages for any predicate satisfied by their attributes. A verifier will be convinced of the fact that the signer’s attributes satisfy the signing predicate if the signature is valid. At the same time, the identity of signer remains hidden. Thus it can achieve anonymous attribute-based access control efficiently. Recently, Yuen *et al.* [47] proposed an attribute-based access control mechanism which can be regarded as the interactive form of ABS.

B. Access Control With Security Device

*1)*Security Mediated Cryptosystem: Mediated cryptography was first introduced in [8] as a method to allow immediate revocation of public keys. The basic idea of mediated cryptography is to use an on-line mediator for every transaction. This on-line mediator is referred to a SEM (security Mediator) since it provides a control of security capabilities. If the SEM does not cooperate then no transactions with the public key are possible any longer. Recently, an attribute-based version of SEM was proposed in [13].

The notion of SEM cryptography was further modified as security mediated certificate less (SMC) cryptography [14], [46]. In a SMC system, a user has a secret key, public key and an identity. In the signing or decryption algorithm, it requires the secret key and the SEM together. In the signature verification or encryption algorithm, it requires the user public key and the corresponding identity. Since the SEM is controlled by an authority which is used to handle user revocation, the authority refuses to provide any cooperation for any revoked user. Thus revoked users cannot generate signature or decrypt ciphertext.

Note that SMC is different from our concept. The main purpose of SMC is to solve the revocation problem. Thus the SME is controlled by the authority. In other words, the authority needs to be *online* for every signature signing and ciphertext decryption. The user is not anonymous in SMC. While in our system, the security device is controlled by the user. Anonymity is also preserve

2)Key-Insulated Cryptosystem: The paradigm of key-insulated cryptography was introduced in [17]. The general idea of key-insulated security was to store long-term keys in a physically-secure but computationally-limited device. Short-term secret keys are kept by users on a powerful but insecure device where cryptographic computations take place. Short term secrets are then refreshed at discrete time periods via interaction between the user and the base while the public key remains unchanged throughout the lifetime of the system. At the beginning of each time period, the user obtains a partial secret key from the device. By combining this partial secret key with the secret key for the previous period, the user renews the secret key for the current time period.

Different from our concept, key-insulated cryptosystem requires all users to update their keys in every time period. The key update process requires the security device. Once the key has been updated, the signing or decryption algorithm does *not* require the device anymore within the same time period. While our concept *does* require the security device every time the user tries to access the system. Furthermore, there is no key updating required in our system.

III. PRELIMINARIES

Pairings : Let G and G_T be cyclic groups of prime order *p*. A map $\hat{e}$:G×G→$G_T$is bilinear if for any generators g ∈Gand *a*, *b* ∈Z_p, $\hat{e}(g^a , g^b) = \hat{e}(g, g)^{ab}$. Let *G* be a pairing generation algorithm which takes as input a security parameter 1^λ and outputs $(p$, G, G, $G_T,\hat{e})$ ← $G(1^{\lambda})$. The generators of the groups may also be given. All group operations as well as the bilinear map $\hat{e}$ are efficiently computable.

*Monotone Span Program :*Our access control mechanism depends on expressing the attribute predicate as a monotone span program. We review some notation about monotone span program using the notation in [35]. Let Υ: $\{0, 1\}^n \rightarrow \{0, 1\}$ be a monotone boolean function. A monotone span program for Υ over a field F is an × *m* matrix **M** with entries in F, along with a labeling function ρ : [1,] → [1,*n*] that associates each row of **M** with an input variable of Υ, that, for every $(x_1, . . . ,x_n)$∈ $\{0, 1\}^n$, satisfies the following:

$$\Upsilon \quad x_1, . . . , x_n) =1 \Leftarrow\Rightarrow \exists v \in F^{1\times}: v\, \mathbf{M}$$

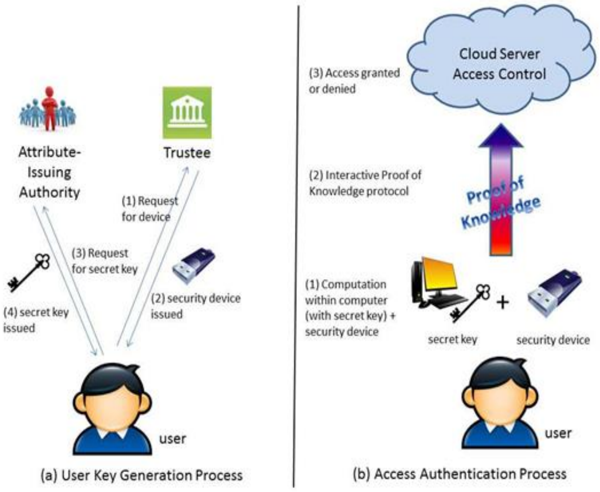
IV. OVERVIEW

A. Intuition

A naive thinking to achieve our goal is to use a normal ABS and simply split the user secret key into two parts. One part is kept by the user (stored in the computer) while another part is initialized into the security device. Special care must be taken in the process since normal ABS does not guarantee that the leakage of part of the secret key does not affect the security of the scheme while in two 2FA, the attacker could have compromised one of the factors. Besides, the splitting should be done in such a way that most of the computation load should be with the user's computer since the security device is not supposed to be powerful.

We specifically design our system in another manner. We do not split the secret key into two parts. Instead, we introduce some additional unique information stored in the security device. The authentication process requires this piece of infor-mation together with the user secret key. It is guaranteed that missing either part cannot let the authentication pass. There is also a linking relationship between the user's device and the secret key so that the user cannot use another user'

Assumptions paper.


Attribute-Issuing Authority
Trustee
(1) Request for device
(3) Request for secret key
(2) security device issued
(4) secret key issued
user
(a) User Key Generation Process
Cloud Server Access Control
(3) Access granted or denied
(2) Interactive Proof of Knowledge protocol
Proof of Knowledge
(1) Computation within computer (with secret key) + security device
+
secret key
security device
user
(b) Access Authentication Process

D. Threat Model

In this paper, we consider the following threats:

1) Authentication: The adversary tries to access the system beyond its privileges. For example, a user with attributes

{Student, Physics} may try to access the system with policy “Staff” AND “Physics”. To do so, he may collude with other users.

2) Access without Security Device: The adversary tries to access the system (within its privileges) without the security device, or using another security device belonging to others.

3) Access without Secret Key: The adversary tries to access the system (within its privileges) without any secret key. It can have its own security device.

4) Privacy: The adversary acts as the role of the cloud server and tries to find out the identity of the user it is interacting with.

OUR PROPOSED SYSTEM

A. Specification of the Security Device

We assume the security device employed in our system satisfies the following requirements.

1) *Tamper-resistance.* The content stored inside the security device is not accessible nor modifiable once it is initialized. In addition, it will always follow the algorithm specification.

2) *Capability.* It is capable of evaluation of a hash function. In addition, it can generate random numbers and compute exponentiations of a cyclic group defined over a finite field.

B. Construction

Let A be the desired universe of attributes. For simplicity, we assume A = $[1,n]$ for some natural number n. We will use a vector $x \in \{0, 1\}^n$ to represent the user’s attribute set. Let $x = (x_1, . . . ,x_n) \in \{0, 1\}^n$. If the user is in possession of attribute i , $x_i = 1$. Otherwise, $x_i = 0$.

1) System Setup: The system setup process consists of two parts. The first part T Setup is run by a trustee to generate public parameters. The second part ASetup is run by the attribute-issuing authority to generate its master secret key and public key.

TSetup: Let λ be a security parameter. The trustee runs $G(1^{\lambda})$ (described in Section III-A) to generate param = $(G, G_T, p, \hat{e})$ and randomly picks generators $g, \hat{g}, h, h_0, h_1, . . . , h_n \in G$. It also picks a collision resistant hash function $H : \{0, 1\}^* \rightarrow Z_p$. Further, let tpk = $\hat{e}(g,h_0)^{tsk}$ for a randomly generated tsk $\in_R Z_p$.

It publishes TPK = $($param$,g,\hat{g},h,h_0,h_1, . . . ,h_n,H,tpk)$.
ASetup: The attribute-issuing authority randomly picks$\in Z_p$ and computes $w = h^{\gamma}$. It publishes APK = (w) and sets ASK = (γ).

2) User Key Generation: The user key generation process consists of three parts. First, the user generates his secret and public key in USetup. Then the security device is initialised by the trustee in Device Initialisation. Finally the attribute-issuing authority generates the user attribute secret key according to the user’s attribute in AttrGen.

USetup: The user randomly picks $y \in Z_p$. It publishes UPK = $Y = h_0^y$ and sets USK = y.

Device Initialisation: The trustee initialises the security device for user (whose public key is with values TY = $\hat{e}(g,Y)$, TG = $\hat{e}(g,h_0)$ and

AttrGen: The key generation algorithm takes as input TPK, APK, UPK = Y and an attribute set A represented as a by a vector $(x_1, . . . ,x_n) \in \{0, 1\}^n$.

Access Authentication: The access authentication process is an interactive protocol between the user and the cloud service provider. It requires the user to have his partial secret key, attribute secret key[3] and the security device.

Auth: The interactive authentication protocol takes as input TPK, APK and a claim-predicate Υ . The user has some additional inputs including an attribute secret key $sk_{A,Y}$ for attribute A, USK = y and the security device.

Assume

$\Upsilon(A)$ =1. Parse$sk_{A,Y}$as (A, e, s, x).

1) The authentication server picks at random a challenge $R \in Z_p$and sends R to the user.
1
2) The user computes $C = \hat{e}(g,h_0)y^{+}R$ and submits (C, y, R) to his/her security device.

3) The security device validates $C^{(y+R)}$=TG and TG^y = TY.
4) Upon successful validation, the security device picks a random $r \in_R Z_p$, computes $c_R = H(TG^r \parallel R \| C)$ and$z_R = r - c_R$tsk. It returns (c_R , z_R) to the user.

5) The user converts Υ to its corresponding monotone span program **M** = $(M_{i,j}) \in (\mathrm{Z}_{p})^{\times m}$, with row labeling ρ : [1,] → A. Also compute the vector
6) For i = 1 to , the user randomly picks $a_i, t_i \in_R \mathrm{Z}_p$ and computes $C_i = g^{v}ih^{t}i$, $D_i = g^{x\rho\,(}i^{)}h^{a}i$. The user also

V. CONCLUSION

In this paper, we have presented a new 2FA (including both user secret key and a lightweight security device) access control system for web-based cloud computing services. Based on the attribute-based access control mechanism, the proposed 2FA access control system has been identified to not only enable the cloud server to restrict the access to those users with the same set of attributes but also preserve user privacy. Detailed security analysis shows that the proposed 2FA access control system achieves the desired security requirements. Through performance evaluation, we demonstrated that the construction is "feasible". We leave as future work to further improve the efficiency while keeping all nice features of the system.

REFERENCES

[1] M. H. Au and A. Kapadia, "PERM: Practical reputation-based blacklisting without TTPS," in *Proc. ACM Conf. Comput. Commun.Secur. (CCS)*, Raleigh, NC, USA, Oct. 2012, pp. 929–940.

[2] M. H. Au, A. Kapadia, and W. Susilo, "BLACR: TTP-free blacklistable anonymous credentials with reputation," in *Proc. 19th NDSS*, 2012,1–17.

[3] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic *k*-TAA," in *Proc. 5th Int. Conf. SCN*, 2006, pp. 111–125.

[4] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Trans. Cloud Comput.*, vol. 3, no. 2, pp. 233–244, Apr./Jun. 2015.

[5] M. Bellare and O. Goldreich, "On defining proofs of knowledge," in *Proc. 12th Annu. Int. CRYPTO*, 1992, pp. 390–420.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007,321–334.

[7] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2004,