Fine-grained Two-factor Access Control for Web-based Cloud Computing Services

Mr. M. Santhosh kumar
MTECH, Department of Computer science Engineering, Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, India.

Mr. G. Sravan Kumar
Professor, Department of Computer science Engineering, Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, India.

Abstract:-

we introduce a new fine-grained two-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 user secret key and a lightweight security device. As a user cannot access the system if s/he does 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, Hash Functions, Two-Factor, Access Control, Web Service

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 data storage, big data management, medical information system 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 pro-vides 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. How-ever, 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 may be 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 organizations. 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 unauthorized party. Even though the computer may be locked by a password, it can still be possibly guessed or stolen by undetected malwares.

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 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.

RELATED WORKS

We review some related works including attribute-based cryptosystems and access control with security device in this section.

Attribute-Based Cryptosystem

Attribute-based encryption (ABE) is the corner-stone 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 encrypt or defines the access policy that the decrypt or (and his/her attributes set) needs to satisfy to decrypt the cipher text. 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 unauthorized data access. Besides dealing with authenticated access on encrypted data in cloud storage service, 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 cipher text (which means the user's attributes set satisfies the prescribed policy), then it is allowed to access the cloud computing service.

Access Control with Security Device

Security Mediated Cryptosystem

Mediated cryptography was first introduced in 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 The notion of SEM cryptography was further modified as security mediated certificate less (SMC) cryptography. 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.

Key-Insulated Cryptosystem

The paradigm of key-insulated cryptography was introduced in 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.

PRELIMINARIES

Pairings

Let G and G_T be cyclic groups of prime order p. A map e^ : G G ! G_T is bilinear if for any generators g 2 G and a; b 2 Z_p, e^(g^a; g ^{b}) = e^(g; g)^{ab}. Let G be a pairing generation algorithm which takes as input a security parameter 1 and outputs (p; G; G; G_T ; e^) G(1). The generators of the groups may also be given. All group operations as well as the bilinear map e^ are efficiently computable.

BBS+ Signatures

We briefly review a signature scheme called BBS+. It belongs to a class of signature schemes, commonly known as CL-signatures. CL-signatures are useful in certifying credentials since their structures allows (1) a signer to create a signature on committed values; and (2) a signer holder to prove to any third party that he/ she is in possession of a signature from the signer in zero knowledge. BBS+ is proposed by Au et al. [3], which is based on the schemes of Camenisch and Lysyanskaya and of Boneh et al. It is also referred to as credential signatures as it is normally used to certify a set of credentials .

Let (p; G; G; G_T ; e^) G(1) be the public parameters as discussed. In addition, let g;^ h; h_0; h_1; : : : ; h_n 2 G be publicly known generators of G.

and the public
signer’s secret key is key
The 2_R Z_p
is w =
h . $(x_0; x_1; : : : ; x_n)$ 2
To sign a message block Z_p^{n+1},
randomly picks e;
s , computes A =
the signer 1 2_R Z_p
$(hh^{x}{}_{0}{}^{0}\ h^{x}{}_{1}{}^{1}\ h^{x}{}_{n}n\ \underline{g^{\wedge s})}$ +e . The signer outputs (A; e; s) as the signature on the block of messages (x_0; : : : ; x_n).

To verify a BBS+ signature, one can test if the following equation holds.

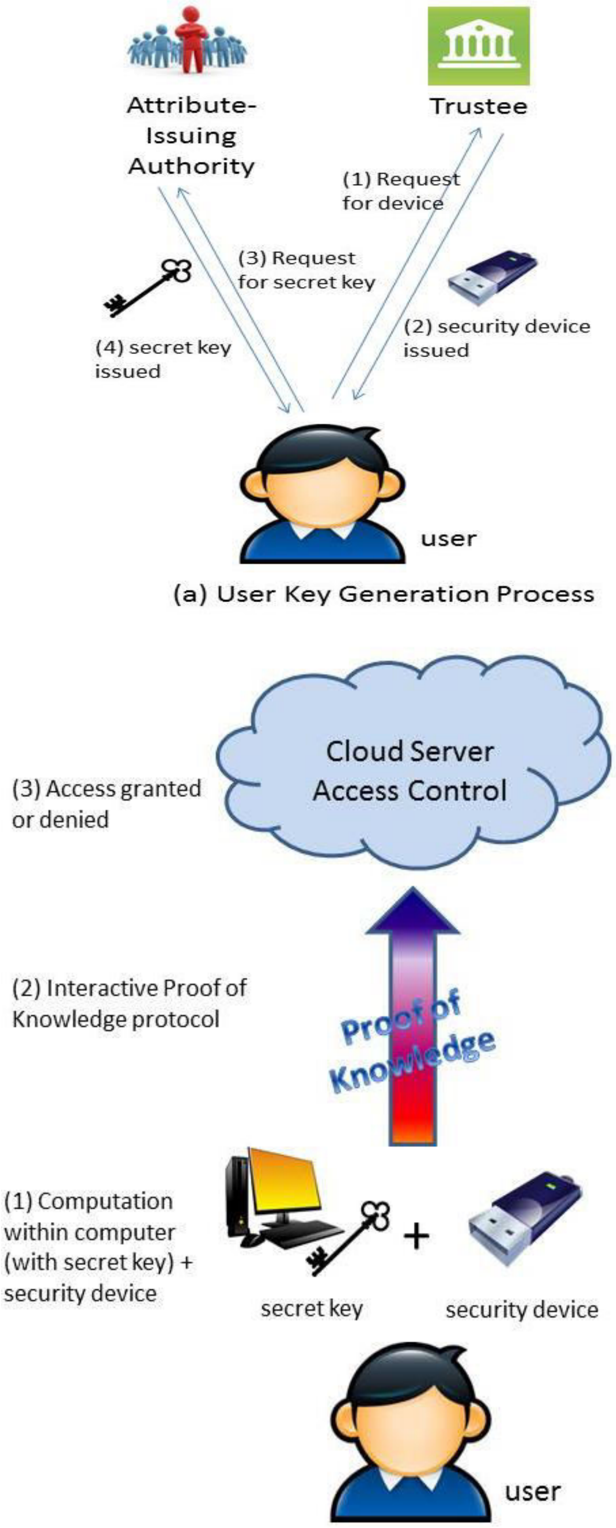
$$e^{\wedge}(A;\ wh^{e}) = e^{\wedge}(hh^{x}{}_{0}{}^{0}\ h^{x}{}_{1}{}^{1}\ \ h^{x}{}_{n}n\ g^{\wedge s};\ h)$$

BBS+ is existentially unforgivable against adaptive chosen message attack under the q-SDH assumption.

OVERVIEW

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


Attribute-Issuing Authority
Trustee
(1) Request for device
(3) Request for secret key
(2) security device issued
(4) secret key issued
user
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

(a) User Key Generation Process

(b) Access Authentication Process

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 information 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's device for the authentication. The communication overhead is minimal and the computation required in the device is just some lightweight algorithms such as hashing or exponentiation over group G_T [2]. All the heavy computations such as pairing are done on the computer.

Entities

Our system consists of the following entities:

Trustee: It is responsible for generating all system parameters and initializes the security device.

Attribute-issuing Authority: It is responsible to generate user secret key for each user according to their attributes.

User: It is the player that makes authentication with the cloud server. Each user has a secret key issued by the attribute-issuing authority and a security device initialized by the trustee.

Cloud Service Provider: It provides services to anonymous authorized users. It interacts with the user during the authentication process.

Assumptions

The focus of this paper is on preventing private information leakage at the phase of access authentication. Thus we make some assumptions on system setup and communication channels. We assume each user communicates with the cloud service provider through an anonymous channel or uses IP-hiding technology. We also assume that trustee generates the security parameters according to the algorithm prescribed. Other potential attacks, such as IP hijacking, distributed denial-of-service attack, man-in-the-middle attack, etc., are out of the scope of this paper.

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.

The following graphs show the bandwidth requirement, computational cost at server and user of our system for policy of various size.

Fig. 2 shows the time cost of the server to authenticate a single user. For a relatively simple policy, say, consisting of 2 clauses with 2 attributes per clause for a total of 4 attributes, the time is less than 0.3 seconds. For a policy of 10 clauses with 10 attributes per clause, the time is around 3 seconds. While the asymptotic complexity at the user is similar to that of the server, the time cost for a user is about five times slower due to the use of a less powerful computing device (a smartphone). One should note that the security device is not the bottleneck as it only accounts for a constant time cost of 0.6 seconds. Please refer to Fig. 3 for the time complexity at the user side. The total authentication time for a policy with 100 attributes, arranged as 10 clauses with 10 attributes each is about 18 seconds. The communication cost of our protocol is depicted in Fig. 4. In particular, for a policy of 100 attributes, the total bandwidth requirement is around 45 KB, which is acceptable for today's network. One could conclude that our protocol is plausible for very simple policy and is still not practical yet for policy of medium size.

Having said that, we would like to remark that the protocol might be optimized. Two possible approaches could be adopted. Firstly, notice that many of the exponentiations are of the form $g^x h^y$ for some fixed bases g and h. This kind of operation is known as multi-base exponentiation and can be computed at about the cost of 110% of a single base exponentiation. It is also worth noting that for fixed base, there are a number of pre-processing techniques available. It is quite likely to reduce the time by half.

(s)

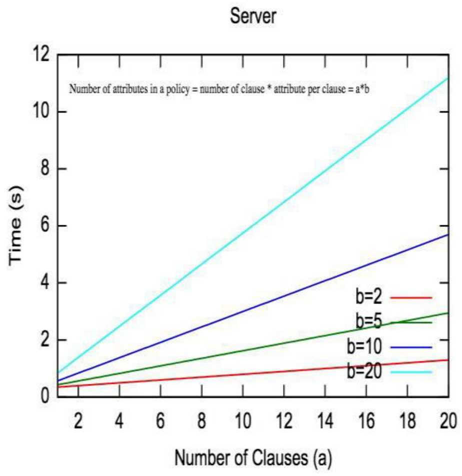


Fig. 2: Running time of the Auth protocol (Server side)

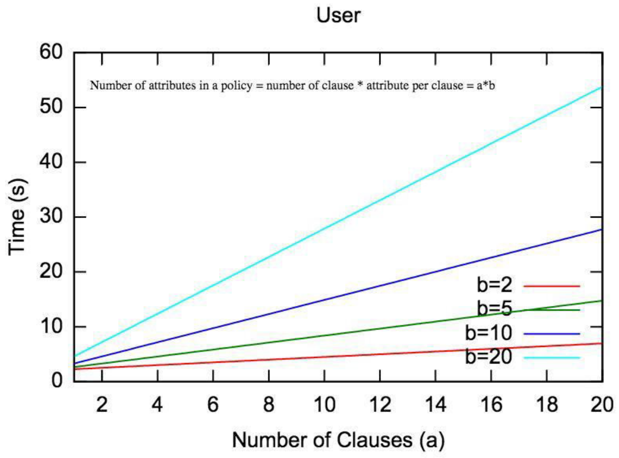


Fig. 3: Running time of the Auth protocol (User side) (s)

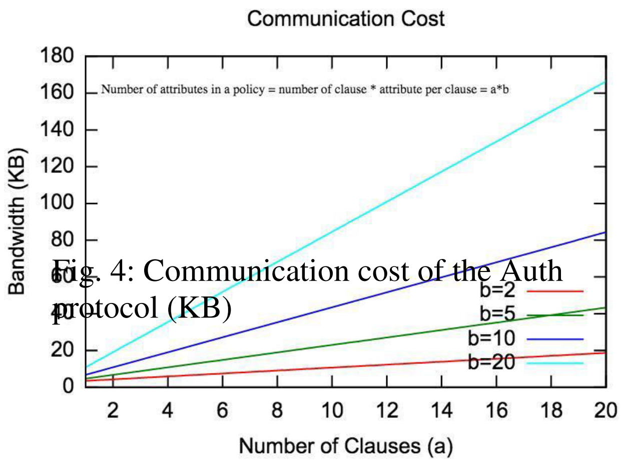


Fig. 4: Communication cost of the Auth protocol (KB)

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 T. Yu, G. Danezis, and V. D. Gligor, editors, the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012, pages 929– 940. ACM, 2012.

[2] M. H. Au, A. Kapadia, and W. Susilo. Blacr: Ttp-free blacklistable anonymous credential with reputation. In NDSS. The Internet Society, 2012.

[3] M. H. Au, W. Susilo, and Y. Mu. Constant-Size Dynamic k-TAA. In SCN, volume 4116 of Lecture Notes in Computer Science, pages 111–125. Springer, 2006.

[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 T. Cloud Computing, 3(2):233–244, 2015.

[5] M. Bellare and O. Goldreich. On defining proofs of knowledge. In CRYPTO, volume 740 of Lecture Notes in Computer Science, pages 390–420. Springer, 1992.

[6] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In IEEE Symposium on Security and Privacy, pages 321–334. IEEE Computer Society, 2007.

[7] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In Franklin [19], pages 41–55.

[8] D. Boneh, X. Ding, and G. Tsudik. Fine-grained control of security capabilities. ACM Trans. Internet Techn., 4(1):60–82, 2004.

[9] J. Camenisch. Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem. PhD thesis, ETH Zurich, 1998. Reprint as vol. 2 of ETH Series in Information Security and Cryptography, ISBN 3-89649-286-1, Hartung-Gorre Verlag, Konstanz, 1998.

[10] J. Camenisch, M. Dubovitskaya, and G. Neven. Oblivious transfer with access control. In E. Al-Shaer, S. Jha, and A. D. Keromytis, editors, Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, November 9-13, 2009, pages 131–140. ACM, 2009.

[11] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In S. Cimato, C. Galdi, and G. Persiano, editors,

Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers, volume 2576 of Lecture Notes in Computer Science, pages 268–289. Springer, 2002

[12] J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In Franklin [19], pages 56–72.

[13] Y. Chen, Z. L. Jiang, S. Yiu, J. K. Liu, M. H. Au, and X. Wang. Fully secure cipher text-policy attribute based encryption with security mediator. In ICICS '14, volume 8958 of Lecture Notes in Computer Science, pages 274–289. Springer, 2014.