Multi Factor Authentication For Web Based Cloud Computing Services

[1]S.Lingapriya, [2]U.S.Sandhya, [3]T.Rajeswari, [4]P.Subbalakshmi
[1] [2] [3]UG Student,[4] Assistant Professor
[1][4]S.R.Rengasamy College of Engineering for Women, Sivakasi
lingapriya13@gmail.com, **subbalakshmirpk2006@gmail.com**

Abstract:

In this paper, we introduce a multi factor authentication for web based cloud computing services. Specifically, in our proposed MFA control system,an Attribute based access control mechanism is applied. This mechanical fundamentals of both a user secret key and a lightweight security device. As a user cannot entry the system if they do not envelop both, then the procedure can upgrade the security of the system, typically in those schemes where many users distribute the same computer for web based cloud services. In addition, the key generation mechanism is implemented. The secret key sends in encryption methods.

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 softwareand 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 tomarket.

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, 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 atnight.
- In a university, computers in the undergraduate lab are usually shared by differentstudents.

Manuscript received March 10, 2015; revised July 31, 2015 and September 20, 2015; accepted September 29, 2015. Date of publication October 26, 2015; date of current version December 24, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61472083, Grant U1405255, and Grant 61402110, in part by the Fok Ying Tung Education Foundation under Grant 141065, in part by the Program for New Century Excellent Talents in Fujian University under Grant JA14067, in part by the Distinguished Young Scholars Fund of Fujian Province, China, in part by the State Key Laboratory of Cryptology Research Fund, China, and in part by the Natural Science Foundation of Guangdong Province for Distinguished Young Scholars under Grant 2014A030306020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. ShouhuaiXu. *(Corresponding author: Man Ho Au.)*

J. K. Liu is with the Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia (e-mail: joseph.liu@monash.edu).

M. H. Au is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csallen@comp.polyu.edu.hk).

X. Huang iswiththe School of Mathematics andComputer Science, Fujian Normal University, Fuzhou 350108, China, and also with the State Key Laboratory of Cryptology, Beijing 100878, China (e-mail: xyhuang81@gmail.com).

R. Lu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: rxlu@ntu.edu.sg).

J. Li is with the Department of Computer Science, Guangzhou University, Guangzhou 510006, China (e-mail: jinli71@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2015.2493983

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.

A more secure way is to use Multi factor authentication (MFA). 2FA is very common among web-based e-banking services. In addition to introduce 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 MFA, 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 MFA system for users in the web-based cloud services in order to increase the security level in thesystem. [6] discussed about a method, Wireless sensor networks utilize large numbers of wireless sensor nodes to collect information from their sensing terrain.

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 storedinside.

With this device, our protocol provides a MFA 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 theaccess.

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 theuser.

To show the practicality of our system, we simulate the prototype of the protocol.

In the next section, we will review some related works that are related to our concept.

II. RELATEDWORKS

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

A. Attribute-BasedCryptosystem

Attribute-based encryption (ABE) [20], [39] is the cornerstone of attribute-based cryptosystem. ABE enables fine-grained access control overencrypteddata using access policies and associatesattributeswith private keys and ciphertexts. 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 signature (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 ofABS.

B. Access Control With SecurityDevice

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 modi-fied as security mediated certificateless (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 preserved.

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 timeperiod.

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 securitydevice every time the user tries to access the system. Furthermore, there is no key updating required in oursystem.

III. PRELIMINARIES

In this section, we introduce the notations deployed in our scheme.

A. Pairings

Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of prime order p. A map $\hat{e}:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ is bilinear if for any generators $g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $\hat{e}(g^a, g^b) = \hat{e}(g,g)^{ab}$. Let $\mathcal{G}$ be a pairing generation algorithm which takes as input a security parameter 1^λ and outputs $(p, \mathbb{G}, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$. The generators of the groups may also be given. All group operations as well as the bilinear map $\hat{e}$ are efficiently computable.

B. Monotone SpanProgram

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

$$\Upsilon(x_1,\ldots,x_n)=1 \Leftrightarrow \exists v \in \mathbb{F}^{1\times\ell} : v\mathbf{M} = [1, 0, 0, \ldots, 0] \text{ and } (\forall i : x_{\rho(i)} = 0 \Rightarrow v_i = 0).$$

In other words, $\Upsilon(x_1, \ldots, x_n) = 1$ if and only if the rows of $\mathbf{M}$ indexed by $\{i \mid x_{\rho(i)} = 1\}$ span the vector $[1, 0, 0, \ldots, 0]$. We call the length ℓ and m the width of the span program, and $\ell + m$ the size of the span program. Every monotone boolean function can be represented by some monotone span program, and a large class does have compact monotone span programs. Given a monotone boolean function Υ, one can use the method given in [20] to obtain the matrix $\mathbf{M}$.

C. BBS+Signatures

We briefly review a signature scheme called BBS+. It belongs to a class of signature schemes, commonly known as CL-signatures [11]. 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 [12] and of Boneh et al. [7]. It is also referred to as credential signatures [2] as it is normally used to certify a set of credentials [1], [10], [34].

Let $(p, \mathbb{G}, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$ be the public parameters as discussed. In addition, let $\hat{g}, h, h_0, h_1, \ldots, h_n \in \mathbb{G}$ be publicly known generators of $\mathbb{G}$.

The signer's secret key is $\gamma \in_R \mathbb{Z}_p$ and the publickey is $w = h^\gamma \in \mathbb{Z}_p$.

To sign a message block $(x_0, x_1, \ldots, x_n) \in \mathbb{Z}_p^{n+1}$, the signer randomly picks $e, s \in_R \mathbb{Z}_p$ computes $A = (hh_0^{x_0}h_1^{x_1}\cdots h_n^{x_n}\hat{g}^s)^{\frac{1}{\gamma+e}}$. The signer outputs (A, e, s) as the signature on the block of messages $(x_0, \ldots, x_n)$.

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

$$\hat{e}(A, wh^e) = \hat{e}(hh_0^{x_0}h_1^{x_1}\cdots h_n^{x_n}\hat{g}^s, h)$$

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

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 MFA, 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 bepowerful.

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 $\mathbb{G}_T$[2]. All the heavy computations such as pairing are done on the computer.

The idea of our system is illustrated in Figure 1.

[2] The exponentiation done in group $\mathbb{G}_T$ is much lighter than those in group $\mathbb{G}$. It can be seen from the simulation data in the benckmark.

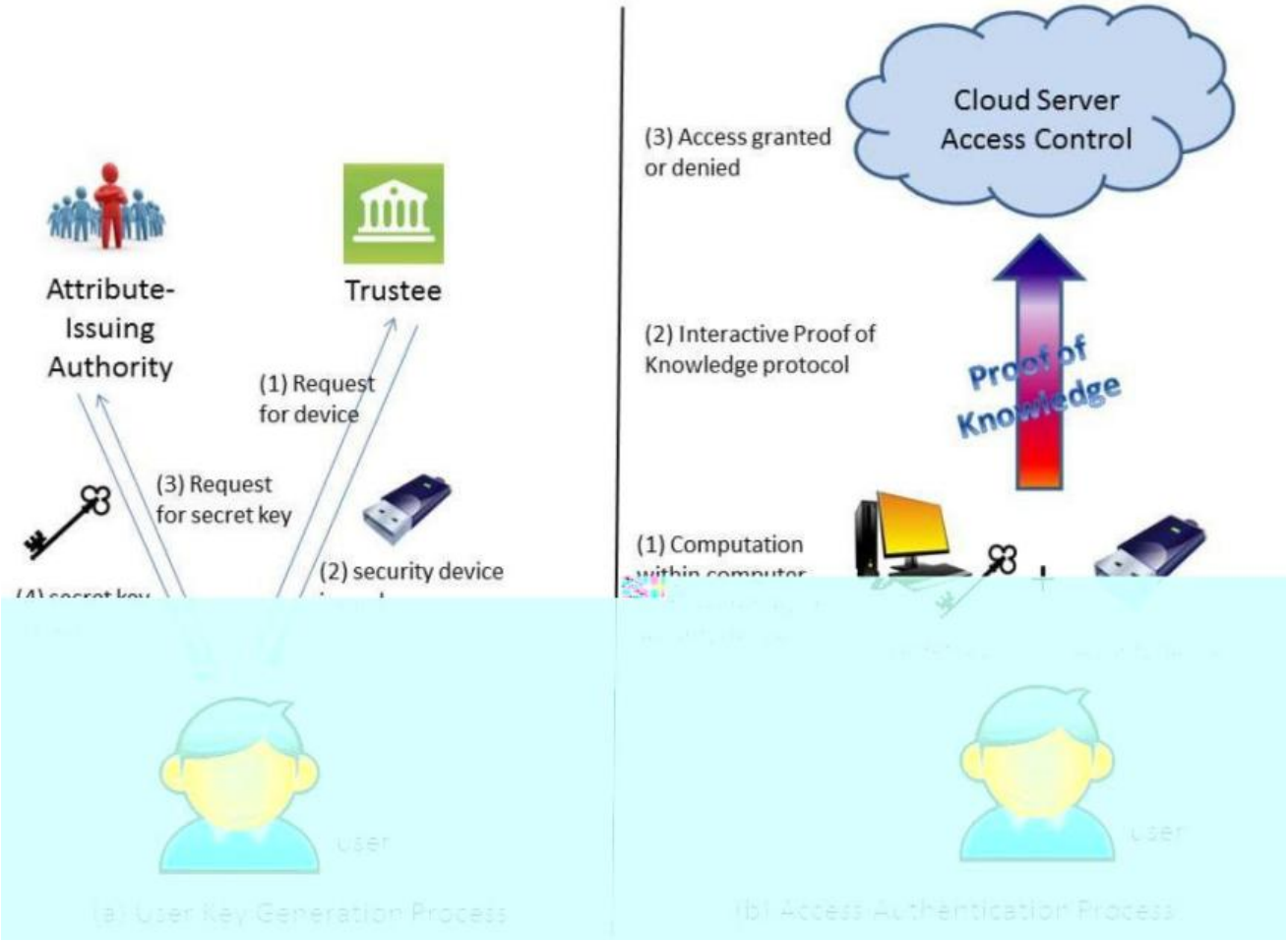


Fig. 1.Overview idea of our system.

B. Entities

Our system consists of the following entities:

- Trustee: It is responsible for generating all system parameters and initialise the securitydevice.
- Attribute-issuing Authority: It is responsible to gener-ate user secret key for each user according to theirattributes.
- 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 thetrustee.
- Cloud Service Provider: It provides services to anony-mousauthorised users. It interacts with the user during the authenticationprocess.

C. Assumptions

The focus of this paper is on preventing private informa-tion 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 [26], [37] 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.

D. ThreatModel

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 otherusers.
2) Access without Security Device: The adversarytries 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 securitydevice.
4) Privacy: The adversary acts as the role of the cloud server and tries to find out the identity of the user it is interactingwith.

TABLE I
FREQUENTLY USED NOTATIONS

TPK	public parameters of the trustee
APK	public key of the attribute-issuing authority
ASK	secret key of the attribute-issuing authority
UPK	public key of the user
USK	partial secret key of the user
$sk_{A,UPK}$	attribute secret key for the user with attribute A and public key UPK
Υ	claim-predicate for the access control

E. Notation

Frequently used notations in our system are summarized in Table I.

V. 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 $\mathcal{A}$ be the desired universe of attributes. For simplicity, we assume $\mathcal{A} = [1, n]$ for some natural number n. We will use a vector $x \in \{0,1\}^n$ to represent the user's attribute set. Let $x = (x_1, \ldots, x_n) \in \{0,1\}^n$. 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 TSetup 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 $\mathcal{G}(1^\lambda)$ (described in Section III-A) to generate $\mathsf{param} = (\mathbb{G}, G_T, p, \hat{e})$ and randomly picks generators $g, g^*, h, h_0, h_1, \ldots, h_n \in \mathbb{G}$. It also picks a collision resistant hash function $H: \{0,1\}^* \to \mathbb{Z}_p$. Further, let $\mathsf{tpk} = \hat{e}(g, h_0)^{\mathsf{tsk}}$ for a randomly generated $\mathsf{tsk} \in_R \mathbb{Z}_p$.

It publishes $\mathsf{TPK} = (\mathsf{param}, g, g^*, h, h_0, h_1, \ldots, h_n, H, \mathsf{tpk})$.

ASetup: The attribute-issuing authority randomly picks $\gamma \in \mathbb{Z}_p$ and computes $w = h^\gamma$. It publishes $\mathsf{APK} = (w)$ and sets $\mathsf{ASK} = (\gamma)$.

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 initialized by the trustee in Device Initialization. Finally the attribute-issuing authority generates the user attribute secret key according to the user's attribute in AttrGen.

USetup: The user randomly picks $y \in \mathbb{Z}_p$. It publishes $\mathsf{UPK} = Y = h_0^y$ and sets $\mathsf{USK} = y$.

Device Initialization: The trustee initializes the security device for user (whose public key is UPK) with values $\mathsf{TY} = \hat{e}(g, Y)$, $\mathsf{TG} = \hat{e}(g, h_0)^{\mathsf{tsk}}$ and tsk.

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

The user runs a zero-knowledge proof of knowledge protocol PK_0 with the attribute-issuing authority to prove the knowledge of his partial secret key y:

$$PK_0\{y : Y = h_0^y\}.$$

This proof of knowledge of discrete logarithm is straightforward and is shown in the next subsection. If the proof is correct, the attribute-issuing authority chooses random $s \in \mathbb{Z}_p$ and uses his secret key ASK to create the user attribute secret key $\mathsf{sk}_{\mathcal{A},Y} := (A, e, s)$ as

$$A = (hY h_1^{x_1} \cdots h_n^{x_n} g^s)^{\frac{1}{\gamma + e}}$$

3) *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 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 $\mathsf{sk}_{\mathcal{A},Y}$ for attribute $\mathcal{A}$, USK $= y$ and the security device. Assume $\Upsilon(\mathcal{A}) = 1$. Parse $\mathsf{sk}_{\mathcal{A},Y}$ as (A, e, s, x).

1) The authentication server picks at random a challenge $R \in \mathbb{Z}_p$ and sends R to the user.
2) The user computes $C = e(g,h_0)^{\frac{1}{y+R}}$ and submits (C, y, R) to his/her security device.
3) The security device validates $C^{(y+R)} = \mathsf{TG}$ and $\mathsf{TG} = \mathsf{TY}$.
4) Upon successful validation, the security device picks a random $r \in_R \mathbb{Z}_p$, computes $c_R = H(\mathsf{TG} \,||\, R || C)$ and $z_R = r - c_R \cdot \mathsf{tsk}$. It returns (c_R, z_R) to the user.
5) The user converts Υ to its corresponding monotone span program $\mathbf{M} = (M_{i,j}) \in (\mathbb{Z}_p)^{l \times m}$, with row labeling $\rho: [l] \to \mathcal{A}$. Also compute the vector $v = (v_1, \ldots, v_m) \in \mathbb{Z}_p^m$ that corresponds to the satisfying assignment $\mathcal{A}$. That is $v\mathbf{M} = (1, 0, \ldots, 0)$. Note that if $x_{\rho(i)} = 0$ (i.e., the user does not possess the attribute $\rho(i)$), v_i must be 0).
6) For $i = 1$ to l, the user randomly picks $x_i, r_i \in_R \mathbb{Z}_p$ and computes $C_i = g^{x_i} h^{r_i}$, $D_i = g^{r_i}$. The user also computes $b_i = t_i - a_i v_i$.
7) For $j = 1$ to m, the user computes $f_j = \sum_{i=1}^{t} M_{i,j}$. Then the user sends $(C, c_R, z_R, C_1, \ldots, C_l, D_1, \ldots, D_l)$ to the authentication server.

[3] We assume the user stores both the partial secret key and attribute secret key in his/her computer.

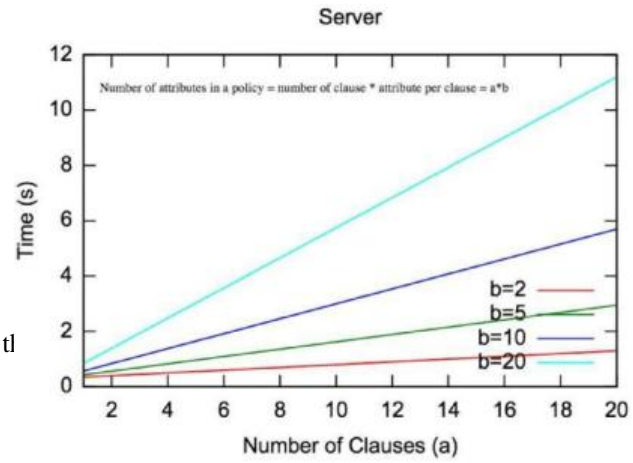
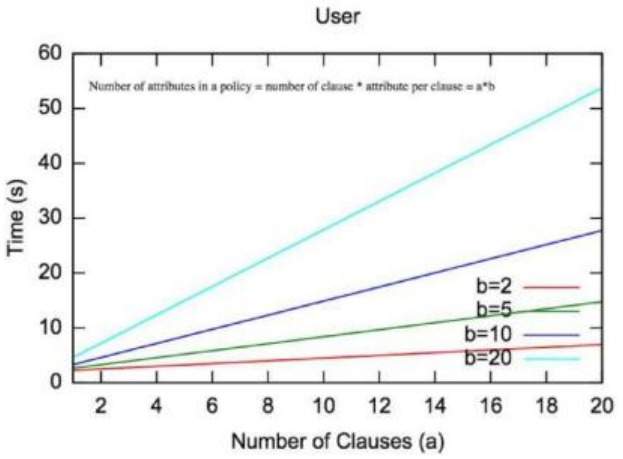


Fig. 2. Running time of the Auth protocol (Server side) (s).



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 optimised. Two possible approaches could be adopted. Firstly, notice that many of the exponentiations are

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 byhalf.

x
of tl

Fig. 3. Running time of the Authprotocol (User side) (s).

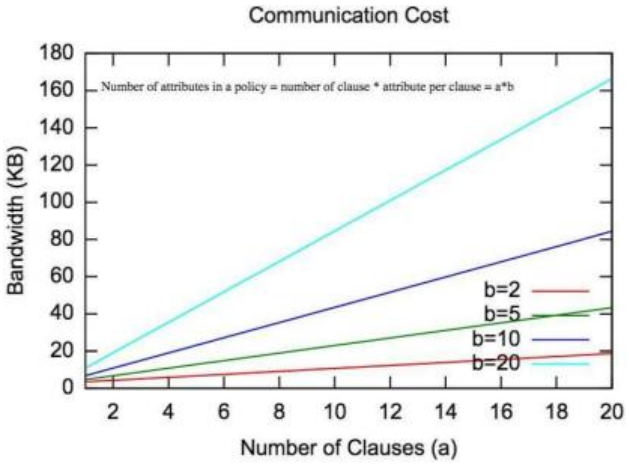


Fig. 4. Communication cost of the Authprotocol (KB).

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

VI. CONCLUSION

In this paper, we have presented a new MFA (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 MFA 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 thesystem.

VII 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, pp. 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] Christo Ananth, T.Rashmi Anns, R.K.Shunmuga Priya, K.Mala, "Delay-Aware Data Collection Network Structure For WSN", International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Special Issue 2 - November 2015, pp.17-21

[7] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2004, pp. 41–55.

[8].J. K. Liu, M. H. Au, W. Susilo, K. Liang, R. Lu, and B. Srinivasan, "Secure sharing and searching for real-time video data in mobile cloud," *IEEE Netw.*, vol. 29, no. 2, pp. 46–50, Mar./Apr. 2015.