UDC 003.26

ORGANIZATION OF PROTECTION MECHANISMS FOR CLOUD STORAGE SERVICES

T. GALIBUS[1], V. KRASNOPROSHIN[1], THIAGO P.B. VIEIRA[2], RAFAEL TIMÓTEO DE SOUSA JÚNIOR[2], JOÃO PAULO C.L. COSTA[2], EDISON PIGNATON DE FREITAS[3], ANTON ZALESKI[4], H.E.R.M. VISSIA[4]

[1]*Belarusian State University, Minsk, 4, Nezavisimosti, 220030, Belarus*

[2]*University of Brasilia, UnB - FT – ENE – CP: 4386, 70910-900, Brasília – DF – Brazil*

[3]*Federal University of Rio Grande do Sul, UFRGS – INF – CP: 15064, 91501-970 Porto Alegre – RS – Brazil*

[4]*Byelex Multimedia Products BV Argon 1, 4751 XC Oud Gastel, The Netherlands*

Submitted 2 December 2015

Providing effective security mechanisms is an important requirement to any secure cloud solution. In this sense, it's proposed to use a hybrid approach to cloud security that includes both warning and detection mechanisms in order to minimize the possibility of a successful attack. According to that proposal, the customized attribute-based encryption (ABE) is a comprehensive access control solution for cloud storage services including user accountability and key revocation. The authors apply state-of-the-art signal processing techniques in order to detect the malicious activities and Man-in-the-Cloud (MITC) attacks in the cloud environment.

Keywords: cloud storage, ABE access control, signal processing, security.

Introduction

Many organizations and enterprises adopted a cloud computing infrastructure, contributing to a global transition to a distributed system paradigm. Cloud services make the data accessible for multiple users and the concept of access control in the cloud data storage should be carefully considered both by providers and end-users/organizations [1]. There are numerous attacks against cloud systems and the one most significant in the protected cloud environment is called “Man in the Cloud” (MITC) [2]. In the typical scenario of such attack, the attacker steals the user credentials (a token or password), and uses this data in order to substitute or to steal the protected data.

There are two basic approaches to address this attack: the cryptographic/key-based mechanisms that work as a precaution against it by means of encryption and secret sharing, and the data collection/traffic analysis mechanisms that allow to detect the attack and to prevent it from being successful as fast as possible [3, 4]. The main difficulty of relying solely on the cryptographic methods is that the attacks are often based on the reverse or social engineering and, therefore, the majority of the attack scenarios cannot be handled completely.

A novel approach to cloud security based on a hybrid protection system is proposed. First it's applied proactive attribute-based-encryption (ABE) in order to protect the access to the protected cloud. The signal processing techniques similarly to a honeypot detection system in order to provide an immediate alarm system for the rapid identification of the MITC attacks was adopted. The system monitors the activity of device users, including the mobile ones, and warns the domain administrator in the case of suspicious actions of the user/device so that the administrator can take the appropriate decision to restrict the access of the dangerous/untrusted device or remove it from the domain.

The components of the cloud storage security infrastructure

The main purpose of the proposed security infrastructure is the setup of efficient access control (AC) and attack detection (AD) in a cloud-based protected environment. The proposed infrastructure for the protected cloud includes the following components illustrated in Fig. 1:

1) encryption server. The encryption server manages all AC and encryption operations and grants the user access to the data storage. This server can store the encryption keys and/or connect to a separate Key Storage;

2) file storage. The file storage is secure in the sense that some of the files specified by the domain administrator are store, encrypted and have restricted access as it's show in Fig. 1. Due to the fact that the file storage data is partially stored in the cloud i.e. externally it is recommended to encrypt this external part of file storage completely;

3) client UI. The client can connect to the encryption server and ask for the permission to access the file storage in order to view/edit/upload specific files or folders;

4) key storage. The key storage is accessible only for the domain administrator;

5) attack detection system. The attack detection system monitors the mobile device activity and detects the threats and suspicious actions.

The Fig. 1 demonstrates the components of the proposed security system.

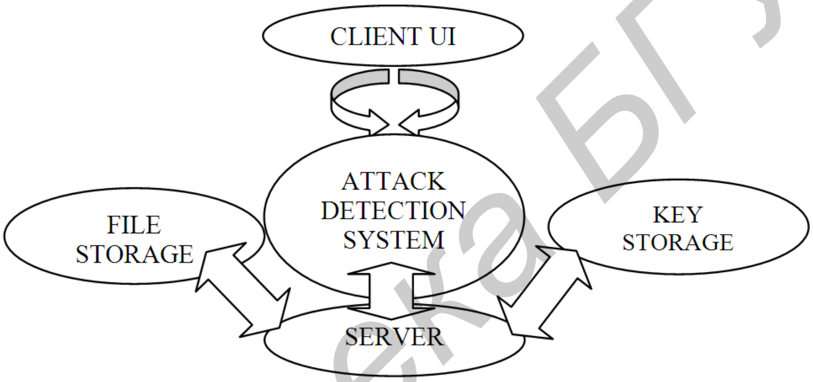


Fig. 1. Components of the proposed security infrastructure and their interactions

In order to increase the speed of data encryption on-the-fly, a hybrid encryption system is set up which combines both symmetric and attribute-based (asymmetric) encryption. More importantly, the basic ABE [1, 5] approach is modified both to increase the speed of encryption and to implement the configuration parameters necessary to set up the user key expiration dates and more sophisticated attribute sets corresponding both to the file shares and to the user groups.

The basic functionality of each security component is briefly described in the following.

1. File storage: bulk data in the protected file storage is encrypted with the appropriate block cypher (AES, Blowfish, IDEA, Serpent). The key to the encrypted data is stored in the key storage component and has an expiration period in order to increase the protection.

2. Key storage: the symmetric keys for accessing the data in the file storage are kept in a separate storage. The protection of the key storage is implemented via some strong authentication method such as two-factor authentication. Additionally, the administrator can set up the key expiration period, use different keys for the different files and/or apply a secret sharing mechanism in order to store the keys for the most sensitive data.

3. Encryption server: the most important cryptographic services are run on the encryption server. This server generates the user keys and receives connections from each client user interface (UI), i.e., it instantiates for each separate user of the system and decides whether the access to the specific dataset should be granted to this user. In addition, the server runs the key renewal routines, stores the user public keys and attributes, as well as the auditing data.

4. Client UI: the client UI connects to the encryption server and checks the expiration period of the user keys (in the case it has been configured for this task) and permits the device user to view/edit/upload data. The Client UI stores the user keys for the ABE and the unique symmetric session keys which serve for restricting the access to the downloaded files. The symmetric keys are encrypted

with the ABE keys. The client UI allows the whole system to work in a heterogeneous environment as it supports different platforms and operating systems.

5. Attack detection system. The client requests are processed via the detection system, which collects data regarding requests (time/frequency/amount data) and analyzes it in order to detect malicious user activity.

The described modular infrastructure allows setting up the different components separately and configuring the security system according to specific needs of the enterprise/organization.

Access control functions

A central process supported by the proposed security infrastructure is the setup of access control for the cloud storage, including the mechanisms to perform the following tasks:

– authentication of users in the cloud system: The initial and simplest authentication is performed by means of the user password and email id. For highly sensitive data it is necessary to implement a more sophisticated two-factor authentication by which, besides the password and access to e-mail, the possession of a specific device is verified too. This can be used in situations such as the government services or otherwise services with highly sensitive data;

– provision of access control functions and protection of data from unauthorized access: The AC services are run by the encryption server, which generates and distributes the user keys and keeps the group attributes along with the file sharing ids. Access control allows to securely distribute and present to the user (or accept from user) only the data that this user is permitted to view/edit. In order to achieve this protection, a special version of ABE is used with the implementation of two possible policies: key policy – KP, and cyphertext policy – CP, in order to support the attributes of the groups of users as well as the attributes of the file shares. This algorithm is developed specifically for the access structure of the proposed cloud protection architecture;

– protection of the user data privacy: Once the user wishes to access a separate file downloaded on his/her user device, the client, after performing authentication, uses his/her ABE key in order to decrypt the symmetric session key and open the file.

The access control for the protected cloud storage is based on the selective ABE encryption. This encryption allows to set up user attributes t_i corresponding to the set of the access identifiers of the user groups, designated Group1, Group2, ... Group*n*:

$t_1, t_2, \ldots, t_n \in Z_q$ where q is prime, Group1 $\rightarrow t_1$, Group2 $\rightarrow t_2$, …, Group$n \rightarrow t_n$.

The access control verifies the hash-value of some open text which could be accessed by a user U if this user has the necessary credentials. This hash value is specified by M. Additionally, there are user attributes $\{t_i\}_U$ and the set of attributes of the encrypted text $\{t_i\}_M$. If at least one attribute in the set $\{t_i\}_U$ is equal to one attribute in the set $\{t_i\}_M$, the corresponding user U can decrypt the text corresponding to M.

The key generation and encryption is implemented on the server side, while the decryption is performed on the client side. The additional parameters of the ABE encryption [5] allow implementing the key revocation and renewal procedures.

The client workflow

The additional alarm protection system based on the signal processing techniques is correlated with the client workflow. Below the typical activities of the client are described. The most important feature of the client-side encryption is the uniqueness and unchangeability of the session key to access the separate shares (files) which allows accessing a share in the off-line mode. But, this feature can compromise security. To address this problem, it's provided a session key encryption by means of the modified ABE with the key expiration period. Thus, we implement a hybrid encryption system:

– file encryption: a unique symmetric key is generated for each file. Then, the file is encrypted with AES-128 using this unique key. The resulting encrypted file is sent to the client (e.g. Iphone, Ipad or Android device). The AES key is encrypted with the modified ABE with respect to the file sharing participants list and sent to the client to be stored locally;

– file decryption: the client checks if the private key of a user is still valid. If so, the file key is decrypted with his appropriate ABE key. If the file key is decrypted successfully then the file is decrypted with AES-128. ABE serves to preserve the access control policy and does not allow the access to the files by unauthorized users.

The typical actions of the user are as follows.

1. The user starts-up the cloud storage client application, enters the PIN code and opens the domain, i.e., the interaction with the protected cloud. Meanwhile, the login procedure receives the actual user ABE private key.

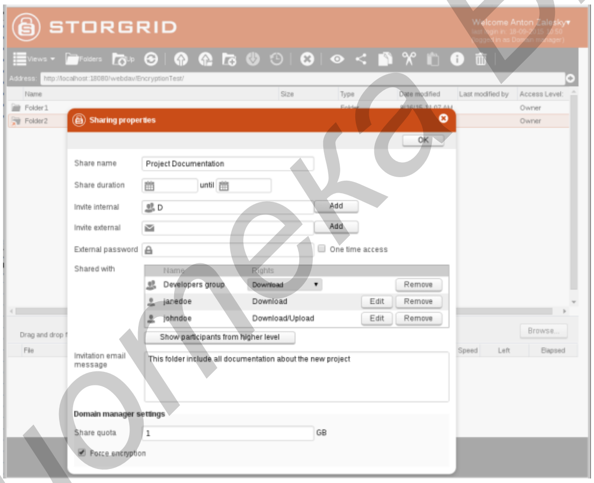
2. The user selects the files for synchronization with the server. This means that encrypted files are saved to the user device storage. A user can see them in his file browser, but these files are encrypted and cannot be used directly.

3. The user clicks via the context menu/button "Decrypt". The file is then decrypted, and directly opened with the application that is associated with the appropriate extension. This is possible since in the background the application checks the private key validity, decrypts the file and saves the non-encrypted file in the local file system.

4. The user modifies the decrypted files and saves them locally. When a file is modified it will be uploaded to the server and will be encrypted again. It is the responsibility of the user to save the file back to the globally controlled cloud storage and not to save it anywhere else. When the user saves the file, it is sent to the encryption server and hence encrypted.

5. The user synchronizes with the global storage in order to save the modifications if allowed.

The Fig. 2 demonstrates the UI of the protected cloud storage.



Share name: Project Documentation

Share duration: until

Invite internal: D

Invite external:

External password: □ One time access

Shared with:

Name	Rights
Developers group	Download
janedoe	Download
johndoe	Download/Upload

Invitation email message: This folder include all documentation about the new project

Domain manager settings

Share quota: 1 GB

☑ Force encryption

Fig. 2. Protected storage UI

Analysis of attack scenarios

Regarding the private keys of the ABE encryption, they are protected by the following mechanisms: centralized generation, uniqueness, and expiration/revocation. Therefore, the attacker cannot do much even if he steals the whole set of keys. He needs to gain access to the device itself, and then is able to explore this access only within the key expiration period. The attacker cannot substitute these keys as performed in the classical MITC attack [2].

The most insecure scenario for the proposed system is when the attacker steals the permanent user credentials, namely, password and login. He can do it by means of social engineering, for example. In order to protect the user from this threat we propose to use such cautionary methods as hashing/secret sharing or PBE/ PAKE protocols [6], which can be integrated to our proposal.

Also, to increase the security an attack detection system, which allows to quickly discovering the malicious activity based on the user actions analysis, is proposed. This system functionality is based on the typical behavior of the attacker after stealing an authorized user credentials:

– downloading user files randomly;

– modifying and uploading files randomly or massively;

– repeatedly downloading the files that are already on the device.

All these types of activities can be detected with signal processing techniques. It's proposed to analyze the following info:

– number of files uploaded/downloaded;

– time delay between downloading the files (if it is too short – this indicates a bot action, not a real user one);

– number of files re-downloaded/uploaded – normally the user keeps the files on his device and doesn’t need to refresh it for some time.

It's proposed to extract the info from the log in the following form:

– apply feature selection techniques to identify valuable features;

– using Model Order Selection schemes for abnormalities detection;

– enriching the attack detection through techniques for obtaining detailed information about attacks.

The log analysis algorithm

Most of the attacks incurs into significant variation (anomaly) from the standard behavior of information systems or present well-known signatures that can be easily detected by a monitoring system. Intrusion detection and intrusion prevention systems are security systems used respectively to detect (passively) and prevent (proactively) threats to computer systems and computer networks. Given the cited attack characteristics, such systems can work in the following modes: signature-based, anomaly-based or hybrid [7, 8].

Signal processing techniques have been successfully applied to network anomaly detection [3, 4] and have been a research problem whose solutions are important in order to achieve improvements in detection accuracy and reducing the computational cost of detection.

In the context of anomaly-based schemes, in this paper the proposed log analysis algorithm applies signal processing techniques, such as Principal Component Analyis and Model Order Selection schemes, for automatic identification of attacks or malicious behaviors. Additionally, other techniques can be used to obtain detailed information about the malicious behavior, making possible to identify patterns and obtain the necessary information for performing reactive and proactive actions against the attacks and possible threats.

Therefore, the desired information for detection purposes is extracted from the collected log, in order to obtain useful features that shall be modeled as matrices that represent a signal superposition containing noise, legitimate and malicious traffic. The table shows the example of the collected log information.

Information extracted from the cloud storage log utility

User login	Average	Min	Max	KB/sec	Avg. Bytes
5 shares available	982	225	1990	146.36	7459
10 shares available	1140	340	4176	223.88	13218
25 shares available	1267	887	2303	294.55	19466

From the extracted features the behavioral evaluation shall be performed for identification of anomalies over time, such as outstanding anomalies or less expressive variations on the observed behavior. For this analysis the authors adopt the eigenvalue analysis based on covariance and correlation, thus highlighting behavior changing that shall be used as input for attack detection through Model Order Selection schemes.

The selected Model Order Selection scheme detects the attack occurrences, and needs to be complemented by techniques to extract detailed information of detected attacks. For detailed information extraction and attack identification the authors apply eigen analysis and similarity analysis for obtaining detailed information about the accurate attack time and the attacker identification.

Conclusion

A novel set of access control and attack detection mechanisms to be used for the organization of cloud storage protection is proposed. This set of mechanisms is rather flexible and allows solving various complex security problems including the attack on the user login/password and unauthorized access to the data. Currently, the system is implemented as an application software and is being tested.

References

1. *Galibus T., Vissia H.* Cloud Storage Security // Proceedings of NSCE'2014. Hong Cong, 24–25 December 2014. P. 123–127.
2. Imperva: Hacker Intelligence Initiative Report – Man in the Cloud (MITC) Attacks. [Electronic source]. Access mode: https://www.imperva.com/docs/HII_Man_In_The_Cloud_Attacks.pdf. – Access date: 28.11.2015.
3. *Lu W., Ghorbani A.A.* Network anomaly detection based on wavelet analysis // EURASIP J. Adv. Signal Process. 2009, Vol. 4(1416). P. 1–17.
4. *Tenorio D.F., da Costa J.P.C., de Sousa Jr R.* // Proceedings of the International Conference on Forensic Computer Science (ICoFCS). Brasilia, 14–16 August 2013. P. 46–51.
5. *Galibus T.V., Vissija H.E.R.M.* // Informatizacija obrazovanija. 2014. № 4. P. 43–48.
6. *Katz J., Ostrovsky R., Yung M.* Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords // LNCS. 2001. Vol. 2045. P. 475–494.
7. *Huang C.-T., Chang R.K.C., Huang P.* Editorial: Signal processing applications in network intrusion detection systems // EURASIP J. Adv. Signal Process. 2009. Vol. 9 (192). P. 1–4.
8. *Mudzingwa D., Agrawal R.* A study of methodologies used in intrusion detection and prevention systems (idps) // Proceedings of IEEE Southeastcon, 2012. Orlando, 15–18 March 2012. P. 1–6.