Implementation of a Group Encryption System in a Cloud-based Environment

Tomasz Hyla
West Pomeranian University of Technology
Szczecin, Poland
e-mail: thyla@zut.edu.pl

***Abstract*— Nowadays, mobile devices offer almost the same level of functionality as standard personal computers. Cloud solution and faster Internet connections allow developers to build applications that do most of data processing in the cloud. On the other hand, cyber-crimes are growing problem and complex information system like cloud solutions are vulnerable to more threats. One of the most dangerous threats, i.e., data loss or leakage, requires countermeasures that will protect against dishonest cloud provider. Group encryption mechanisms are one of the key elements to ensure data privacy. This paper presents a new architecture for group encryption system that uses bilinear mappings. The architecture uses cloud solutions and supports mobile devices. Pros and cons of moving cryptographic operations to a cloud and resulting from the analysis of the demonstration system are discussed.**

Keywords-group encryption; cloud; mobile device; architecture; bilinear mapping.

I. INTRODUCTION

Nowadays, mobile devices offer almost the same level of functionality as standard personal computers. Of course some engineering applications requiring high-end processors are not available for mobile devices and someone might not use them due to lower screen sizes. However, cloud solution and faster Internet connections allow developers to build applications that do most of data processing in a cloud. The use of mobile devices and cloud computing increases, because of its desirable properties, like rapid elasticity or broad network access [1].

The cybercrimes are growing problem. The protection against them requires to constantly develop new security measures that will secure more and more complex systems that are using mobile devices and the cloud. The new threats related to the cloud together with proposed countermeasures are listed in [2]. One of the most dangerous threats, i.e., data loss or leakage, requires countermeasures that will protect against dishonest cloud providers.

One of the main business applications of mobile devices is reading and writing different types of documents. Those documents usually contain some kind of information that cannot be disclosed. When many entities are involved in documents' exchange, group encryption schemes can be used to ensure data privacy. The scheme must have properties that will allow to encrypt a document in such a way, that entities from authorised group can decrypt it when they will meet certain conditions.

The conditions required to access an asset can be described using access structures [3]. An access structure is a rule that defines how to share a secret, or more widely, who has an access to particular assets in information system. Access structures can be classified as structures with or without threshold. Although threshold access structures are frequently used, the non-threshold structures (i.e., general access structures) are more versatile.

In this paper, a new architecture for group encryption system, that uses advanced cryptographic operations is presented. The architecture uses cloud solutions and support mobile devices. Pros and cons of moving cryptographic operations to a cloud and these resulting from an analysis of demonstration system are discussed.

The reminder of this paper is as follows. Section 2 contains description of group encryption schemes with an emphasis on Certificate and ID-Based group-oriented Encryption scheme with General Access Structure (CIBE-GAS) scheme and library which contains its implementation. Section 3 introduces a cloud-based architecture for a group-encryption scheme together with a presentation of encryption and decryption processes in the demonstration system. The paper ends with conclusions.

II. GROUP ENCRYPTION

In the group encryption schemes a group of users must act together to decrypt or encrypt a file. This can be achieved in two ways. In the first one, group members consecutively encrypt the file using private keys. In the second one, the encryption key is calculated using private keys from each group member (the simplest solution is to use xor operation) by a designated user from the group. The designated user encrypts the file using the group key and deletes the key. In both cases, it is assumed that intermediate, temporary files (e.g., partial keys, partially decrypted files) that are created during encryption or decryption are deleted after the process is finished.

The private keys of each group member should be kept in secret. Several techniques exists: a key is created on the fly from a password that is entered by a user; a key is stored in an encrypted form and a user password is used to decrypt a key; or a key is stored inside a secure device (e.g., a smart card, a trusted platform module) and can be accessed after a user authenticates to the device.

Currently, many group encryption algorithm exists. Further in this section is described the group encryption algo-

 ISBN: 978-1-61208-399-5

rithm, which is using pairing-based cryptography and has properties interesting from the perspective of cloud implementation.

A. Cryptograhic Scheme

The CIBE-GAS [4] is a group encryption scheme that is more suitable, comparing to threshold secret sharing methods, when the same access rights to decrypt data should be selectively assigned to all participants belonging to the same well defined group of users. The original version of CIBE-GAS scheme works with limited length messages only, while its modification Certificate and ID-Based group-oriented Encryption scheme with General Access Structure Hybrid (CIBE-GAS-H) [5] works with arbitrary length messages.

In CIB-GAS scheme a designated user (i.e., a dealer) is responsible for encrypting documents. The encryption algorithm requires as an input: dealer identity; public and private keys; public system parameters; information about privilege set of users who will be able to decrypt a document; and public share information belonging to users from the privileged set. During encryption no communication between the dealer and users from the privilege set is required. Public share information enables the dealer to encrypt a file in such a way, that only users who have a private keys associate with public share information will be able to partially decrypt a file. Decryption has two phases. In the first phase, each of users from the designated set using the ciphertext partially decrypts the text. In the second one, combiner (a user whom other users have transfer rights to decrypt the document) decrypts the ciphertext using the values obtained from partial decryption from all required users from the designated set.

The scheme combines three different ideas [4]: the secret sharing scheme [6], publicly available evidence of being a member of a particular group [7] and Sakai-Kasahara Identity Based Encryption (SK-IBE) scheme [8] with technique introduced by Fujisaki and Okamoto [9]. As a result the following properties where achieved:

a) the dispatcher (i.e., an entity which encrypts the document) is not required to know the structure of qualified subsets, which members are authorised to decrypt the information;
b) there is no need to designate a specific recipient of encrypted information - each member within a qualified subset can decrypt it; moreover, a dispatcher can temporarily remove some subgroups from having access rights to encrypted information (i.e., a dispatcher can arbitrarily select the recipients by overlaying the appropriate filter on the access structure);
c) the CIBE-GAS scheme is the certificate and identity based encryption scheme; in the scheme partial key created by trusted authority is published as a certificate and it allows simplifying the user's identity verification.

More about the CIBE-GAS and CIBE-GAS-H schemes can be found in [4] [5].

B. Code Libraries

The CIBE-GAS and CIBE-GAS-H schemes were implemented and are a part of the mobile Pairing-Based Cryptography (mPBC) library which is a part of MobInfoSec project [10][11]. The schemes are built on bilinear mappings [12]. Bilinear mappings requires complex mathematical operations, so the schemes were implemented using Pairing-Based Cryptography (PBC) library written by Ben Lynn [13]. PBC is one of the first libraries that allowed to write a code using bilinear mappings. Developer only needs to know mappings properties and internals are hidden.

PBC is written in C language and uses GNU Multiple Precision Arithmetic (GMP) library [14]. Similarly to PBC, which hides bilinear mapping internals, the mPBC hides the CIBE-GAS schemes internals from users and provides high level Application Programming Interface (API). Hence, mPBC user does not need any knowledge about pairing-based cryptography. Also, mPBC contains data structure definitions, import and export functions and tests that demonstrate basic functionality. The mPBC purpose is to provide implementation of cryptographic schemes that can be used directly or indirectly on mobile devices.

Except CIBE-GAS and CIBE-GAS-H schemes mPBC library also contains other schemes that support digital signature and public key encryption built on top of bilinear mappings.

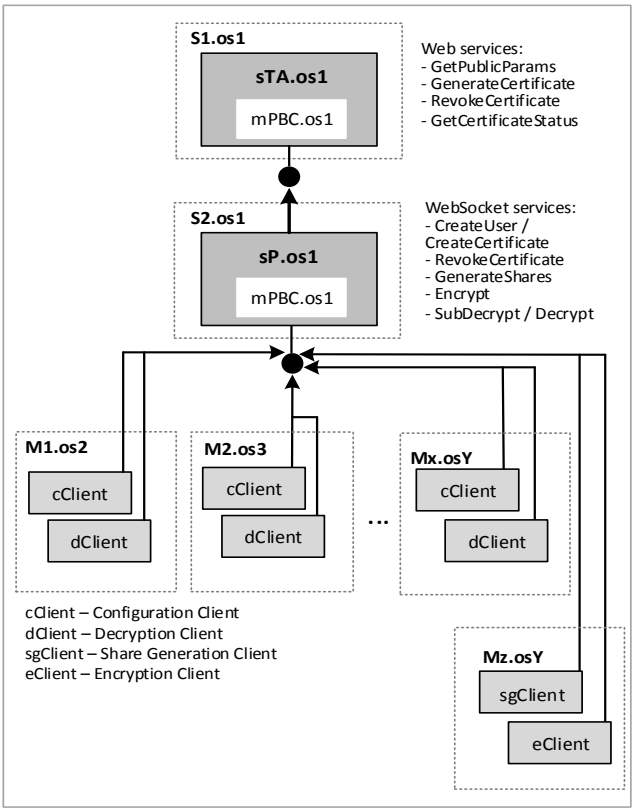

Figure 1. Cloud-based architecture.

The mPBC library can be easy used on Linux and Windows operating system as they natively support C language. The Android, Windows Phone 8.1 and iOS also supports C language, although mPBC library will require some minor modifications to work in each of these systems.

III. CLOUD-BASED IMPLEMENTATION

The traditional way to implement a group encryption scheme in a mobile environment would be a client-server model. However, nowadays when mobile devices use many different operating systems and cloud solutions are available, the cloud-based approach has several advantages. The two are the most important. The first one, is a simple design of mobile clients. The second one, is that the mPBC library needs to be implemented only in one programming language.

A. Architecture

The cloud-based architecture consists of two logical servers, which provide two sets of services (Figure 1). The first one, Trusted Authority server (*sTA*) deployed on the server *S1* is responsible for management of system parameters, users and certificates as it is required by the CIBE-GAS scheme and provides appropriate services. The *sTA* server uses mPBC library in the version for *S1.os1* operating system.

The second server, secret Protection (*sP)* server, provides web socket services. The services enable cryptographic operations (from CIBE-GAS scheme) that normally would be executed on the mobile devices. The services provide operations like encryption and partial decryption. Transferring cryptographic operations to *sP* server eliminates the need to port mPBC library for every mobile operating system, but requires creating secure communication channels to mobile devices. Also, it requires that *sP* server is trusted and provides the same level of security as *sTA* server. This might be seen as drawback, but it also simplifies the development of client applications for mobile devices. Particularly reducing the number of security issues that must be considered.

A client application, deployed on the mobile device, can be developed as native or web browser application. In both cases, user's private files (keys, parameters) are stored locally by *sP* server. Private files from all users are managed by *sP* server and are used indirectly by users through the cryptographic services provided by the *sP* server.

B. Demonstration System

The demonstration system consists of two servers written in C# language using MS Visual Studio 2013:

- the certification server, which provides *sTA* services;
- auxiliary server *sP* which provides functionality required to, among others, initiate certificates generation, encrypt, partially decrypt or decrypt a document; the server uses WebSocket technology to provide that functionality.

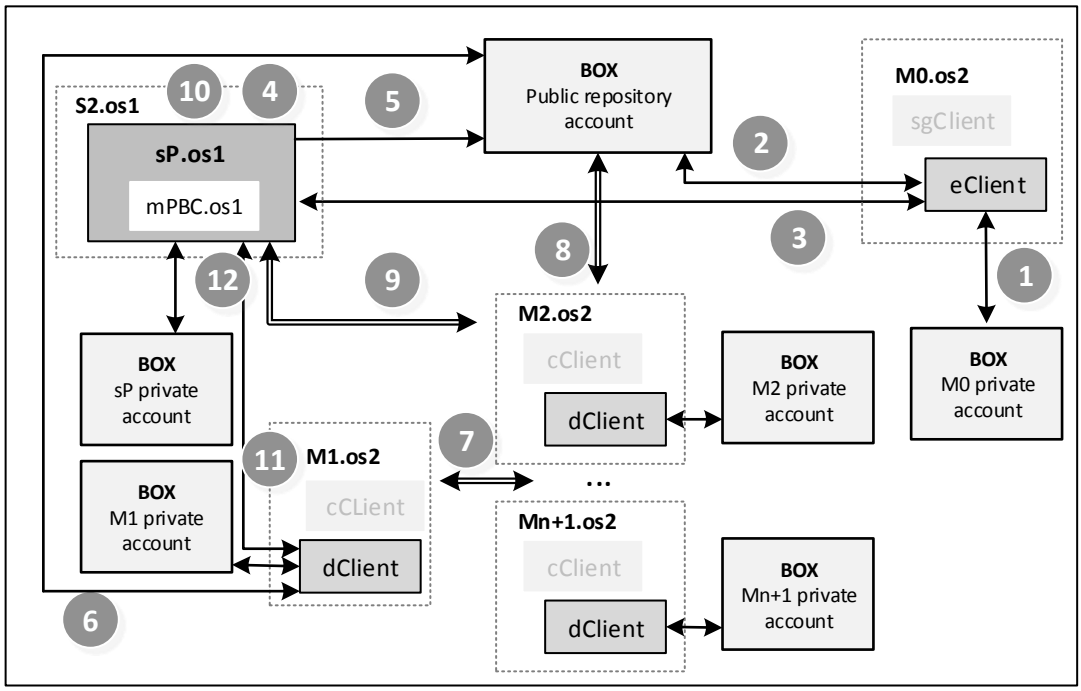

Figure 2. Encryption and decryption processes in the demonstration system

Client applications *cClient, dClient* (run by users) and *sgClient, eClient* (run by a dispatcher) are native Windows Phone 8.1 applications. The communication channel between mobile devices and *sP* server is not secured to simplify system development. In a working system technologies, like some version of TSL (Transport Layer Security), would be probably used. Also, other security measures, e.g., presented in [15], should be deployed to protect *sTA* and *sP* servers and client applications.

The demonstration system uses Box cloud drive to store data. Each mobile device, i.e., mobile device user, and each server has one associated box.com account. Also, there is one cloud drive for the public repository, which is publicly available for all client applications to temporary store generated files and then to share their public address to a specific authorised user.

The encryption and decryption processes main steps are presented in the Figure 2. The encryption process is as follows:

1. Dispatcher (a user with dispatcher rights in a dispatcher role) using *M0.eClient* application retrieves a public link to a file that he wants to encrypt.
2. A file with access structure information is retrieved from public repository, an access policy is created based on that file and then the policy is stored in *M0* private Box account.
3. *M0.eClient* sends to *S2.sP* the public links to the file and to the access policy.
4. The *S2.s:* downloads files from the links, gets user keys form *S2.sP* private Box account, downloads from the public repository required users' public share information, and using mPBC library executes CIBE-GAS-H Encryption algorithm.
5. The *S2.sP* stores an encrypted file and the access policy in the public repository Box account.

The decryption main steps are:

6. The *M1.dClient* searches repository and finds links to the selected encrypted file and to accompanying access policy. Then downloads the access policy.
7. The *M1.dClient* chooses *n* number of devices which are required to do partial decryption (based on the access policy) and sends them partial decryption requests.
8. Each *Mi.dClient* where *i=2..n+1*, downloads using provided links the encrypted file and the access policy.
9. Each *Mi.dClient* sends request to *S2.sP* to execute CIBE-GAS-H SubDecryption-H algorithm.
10. The *S2.sP* for each *Mi.dClient* executes the requested algorithm using keys associated with each *Mi.dClient* and sends to each device a public link to the partially decrypted file stored on *S2.sP* private Box drive.
11. The *M1.dClient* collects the public links from each *Mi.dClient* and sends to *S2.sP* links with requests to combine partially decrypted files.
12. The *S2.sP* executes the CIBE-GAS-H Decryption-H algorithm and returns to *M1.dClient* a public link to a decrypted file in its Box drive.

IV. CONSLUSION

In this paper, firstly, the CIBE-GAS scheme was described. Subsequently, the cloud-based architecture for the CIBE-GAS scheme was presented together with the presentation of encryption and decryption processes in the demonstration system.

The cloud computing have essential properties such as rapid elasticity and resource pooling [1]. This enables an operator to easily adjust number of server instances and other resources to change number of users. From the other side, resource pooling characteristic says that users generally do not know where physically their data are processed. The architecture presented in this paper is a typical hybrid cloud. The public repository can be in public cloud in contrast to servers and private drives which must be held in a private cloud. In presented demonstration system, clients use also the public cloud (Box.com drive), because it simplifies the implementation process.

The performance tests have shown, that in cases of encryption and decryption using *sP* server the time of cryptographic operations from CIBE-GAS scheme is significantly shorter in relation to the time required to retrieve documents from cloud drives. However, the total time of these operations is acceptable and mostly depends on Internet connection speed. It must be noted, that in a working system the authentication of users before usage of clients on mobile device is required.

The scalability is an important issue in cloud-based application development. The ability to scale up the application to millions of users depends mostly on the mutual relation between application instances. In the proposed architecture, the servers in the cloud can perform calculations independently for each request from client applications on mobile devices. This is very good situation as it is possible to run many instances of servers in the cloud with minimum effort.

The key advantages of cloud-based approach for encryption system implementation are: better scalability of the system when number of users increases and also faster and simpler implementation for different mobile operating systems. Especially, mPBC library which contains complicated cryptographic operation needs only to be implemented in the version for one operating system.

The main drawbacks are the necessity to create another trusted auxiliary server for cryptographic operations (*sP* server) and the need to create more trusted channels. The channels must be created between clients on mobile devices and between clients and *sP* server. Also, security threats associated with the cloud must be considered during system development. Particularly, the *sP* server must have the same security level as *sTA* server that manages user certificates.

The further works will mainly focus on general access structures as currently they are implemented in the simplest way that is required by CIBE-GAS scheme.

 ISBN: 978-1-61208-399-5

ACKNOWLEDGMENT

This scientific research work is supported by National Centre for Research and Development (NCBR) of Poland (grant No. PBS1/B3/11/2012) in 2012-2015.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST SP-800-145, September 2011.

[2] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing v3.0", P. Simmonds et al. (Eds.), 2011.

[3] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," Electronics and Communications in Japan (Part III: Fundamental Electronic Science), vol. 72.9, 1989, pp. 56-64.

[4] T. Hyla and J. Pejaś, "Certificate-Based Encryption Scheme with General Access Structure," In: Cortesi, A. et al. (Eds.), CISIM 2012, LNCS, vol. 7564, Springer-Verlag, 2012, pp. 41–55.

[5] T. Hyla and J. Pejaś, "A practical certificate and identity based encryption scheme and related security architecture," K. Saeed, R. Chaki, A. Cortesi, S. Wierzchon (Eds.), CISIM 2013, LNCS, vol. 8104, Springer-Verlag, 2013, pp. 178-193.

[6] Y. Sang, J. Zeng, Z. Li, and L. You, "A Secret Sharing Scheme with General Access Structures and its Applications," International Journal of Advancements in Computing Technology, Vol. 3, No. 4, May 2011, pp. 121-128.

[7] Y. Long and Chen Ke-Fei, "Construction of Dynamic Threshold Decryption Scheme from Pairing," International Journal of Network Security, Vol.2, No.2, March 2006, pp. 111–113.

[8] R. Sakai and M. Kasahara, "ID based cryptosystems with pairing on elliptic curve," Cryptology ePrint Archive, Report 2003/054.

[9] E. Fujisaki and T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes," In Proceedings of CRYPTO '99, Santa Barbara, CA, 1999, pp. 537–554.

[10] T. Hyla, J. Pejaś, I. El Fray, W. Maćków, W. Chocianowicz, and M. Szulga, "Sensitive Information Protection on Mobile Devices Using General Access Structures," The Ninth International Conference on Systems (ICONS 2014), IARIA, Feb. 2014, pp. 192-196, ISSN: 2308-4243, ISBN: 978-1-61208-319-3

[11] I. El Fray, T. Hyla, and W. Chocianowicz, "Protection Profile for Secure Sensitive Information System on Mobile Devices," K. Saeed and V. Snasel (Eds.): CISIM 2014, LNCS 8838, Springer-Verlag, 2014, pp. 636–650.

[12] B. Lynn, "On the implementation of pairing-based cryptosystems," PhD Dissertation, Avaliable from: http://crypto.stanford.edu/pbc/thesis.pdf, June 2007, [retrieved: February, 2015].

[13] B. Lynn, "Pairing-based cryptography library," Avaliable from: http://crypto. stanford.edu/pbc/, v-0.5.14, C language, LGPL license, [retrieved: February, 2015].

[14] The GNU Multiple Precision Arithmetic Library, Avaliable from: https://gmplib.org/, Edition 6.0.0, [retrieved: February, 2015].

[15] K. Salah, J. M. Alcaraz Calero, S. Zeadally, S. Al-Mulla and M. Alzaabi, "Using Cloud Computing to Implement a Security Overlay Network", IEEE Security and Privacy, Vol. 11, No. 1, January/February 2013, pp. 44-53.

 ISBN: 978-1-61208-399-5