Full Life Cycle Protection and Secure Distribution of Files in Clouds

Vladimir Hovsepyan
National Polytechnic University of Armenia
Yerevan, Armenia
e-mail: vladimirhovsepyan@gmail.com

ABSTRACT
Cloud based application which can provide functionalities such as secure file sharing, distribution and collaboration do not currently exist. All existing applications provide only some part of the required functionality, which make impossible to reject and not use files in decrypted state. In this paper is proposed to prevent cloud access to pure data in order to provide enhanced security for files stored in cloud. File distribution and multi user collaboration create many challenges which are solved by using combination of pear to pear and server-client networks.
Cloud based application which provides file full lifecycle security including stored data encryption and secure distribution is designed. The P2P secure data exchange, http collaboration and few other solutions are also presented in this research.

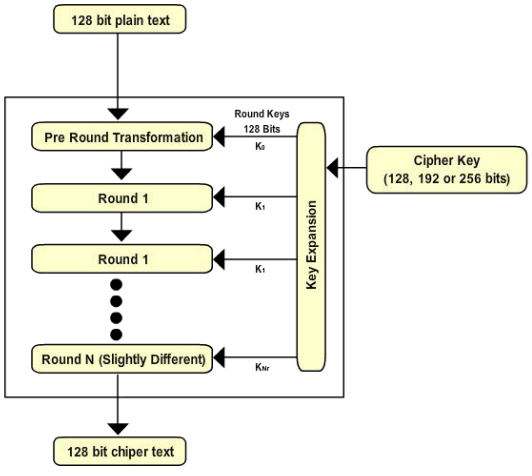
Keywords
cloud security, authentication, p2p connection, cross platform, server client network

General Terms
Security, File Distribution, Encryption

1. INTRODUCTION
Most of the existing applications that provide files protection are used only for certain cases. For example Folder Lock and many other leading file protection software provide file backup in cloud, but they do not support secure file distribution / sharing. File sharing among users should be done manually which requires provision of files in decrypted state or provision ofkeys with encrypted files. Both operations are quite vulnerable. It became obvious that in order to provide maximum protection to files, application should also provide some set of additional service that will make no sense to use files in non-decrypted way. The functionality of such file distribution in cross platform devices is the most important for users. The user should be able to access and share his/ her files quickly and easily on any device. Also to earn trust of any customer regardless of the customer nationality, work type and other characteristics of all information should be sent and stored in cloud only in encrypted way without providing information how to decrypt it. It is hard to find a single application which has all these functionalities and can be run on most devices. It is possible to achieve the desired functionality by using combination of different applications but it is quite inconvenient and besides it requires more resource consumption. So in this paper it is suggested to create an application that will support all the above mentioned functionalities and will perfectly fit to all user requirements.

TECHNOLOGY SELECTION
Decision on using flash as development environment was made, because it is supported by the major operation systems and can be executed on most of the desktops, mobile and tablet devices. This technology is supported by 3 major operation systems and can be executed within browser directly without any additional software installation. Easy user interface development is another good advantage that flash has.

2. EFFICIENT ENCRYPTION
Advanced Encryption Standard (AES) also referred as Rijndaelis is one of the efficient algorithms in the world. The AES standard has a constant block size of 128 bits with 3 different key sizes of 128 bits, 192 bits and 256 bits, where 10, 12 and 14 encryption rounds will be applied for each key size, respectively. During the encryption and decryption processes, the 16 bytes of data will form a changeable (4*4) array called the state array. During the encryption process, the state array initially consists of the input data. This array will keep changing until reaching the final enciphered data. In the decryption process the state array will start by the enciphered data and will keep changing until retrieving the original data. The encryption of AES is carried out in blocks with a fixed block size of 128 bits each. The AES cipher calculation is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform the cipher text back into the original plaintext using the same encryption.



Using AES for files encryption looked like a good idea until speed testing. Results that give this symmetric algorithm are too slow. The length of key for testing has been selected 128. For 1mb data encryption different libraries use 4–7 seconds which is quite a big amount of time. In case of smartphones this time will increase several times, as for, the user that records small video (100mb) on his mobile device and wants to encrypt it should wait more than 20 minutes. Also during encryption time most of the device resource will be used and the device will become non usable. Based on this test we can come to conclusion that pure AES is not a possible use. In order to make AES work faster several improvements has been done.

Parallel Computing

One of the things that have been done to improve speed of AES is using parallel computing. Parallel computation is a method which allows carrying out several computations simultaneously on two or more microprocessors. Parallel computation can be performed by using multicore and multiprocessor computers having multiple processing elements within a single machine. Flash Player starts from version 11.3 supports parallel computing in the form of workers. To achieve the desired functionality, to cover some limitation and make application work on versions under FlashPlayer 11.3 a custom library has been created implementing java concurrency library. This modification does not change the original algorithm itself. Appended written code divides the plaintext into blocks which can be encrypted and decrypted independently. This way the multiple blocks can be processed simultaneously. The first operation here is separation of the plain text or cipher blocks into independent streams and then application of the AES encryption or decryption procedures. The speed improvement measurement depends upon device (the number of the processing units). It can be faster up to 3 times. In case of mobile devices speed improvment is very important.

Intel AES

Intel AES instructions are a new set of instructions available beginning with the all new 2010 Intel® Core™ processor family based on the 32nm Intel® micro architecture codename Westmere. The architecture consists of six instructions that offer full hardware support for AES. Four instructions support the AES encryption and decryption, and the other two instructions support the AES key expansion. They offer a significant increase in the performance compared to the current pure-software implementations. ALSO the AES instruction provides important security protections. In order to support Intel AES instruction for flash player native extension has been written that delegates several operations. AESENC, AESENCLAST, AESDEC, AESDECLAST are defined by the pseudo code. These instructions perform a grouped sequence of transformations of the AES encryption/ decryption flows (in fact, they perform the longest sequence possible, without introducing a branch in an instruction). The above described changes can increase performance by more than an order of magnitude for parallel modes of operations. Beyond improving performance, the new instructions help address software side channel vulnerabilities, because they run with data-independent latency and do not use lookup tables.

APPLICATION ENVIRONMENT

A decision on using a Flash as development environment has been made, because it is supported by major operation systems and can be executed on most of the desktops, mobile and tablet devices. It is supported by 3 major operation systems and can be run on the most existing browsers directly without any additional software installation.

Flash player support multi threading through worker api. The last flash player versions bring that ability to mobile platforms (Android and IOS). Each Flash Player application can run 10 independent workers(threads) and work with them in same environment. Adobe continue developing this technology in order to provide better support and higher efficiency to flash interpreter. Easy user interface development is another good advantage that flash has. Adobe Flash content can easily and consistently move between the browser and native operating systems to reach users on the devices of their choice.

3. CLOUD PROTECTION

Cloud storage is a convenient, accessible technology that gives us access to our data anywhere, on multiple devices. Privacy is very important when it comes to the cloud storage. There always arise 2 questions when cloud security is discussed. First question is about strongest of cloud security and the second one is about trust to that cloud. Currently there are many good practices and ready solutions which provide a complete security for cloud servers. Nowadays the second question is more actual. Whether the user uses cloud storage for music, tax returns, or backups, it is still important to know that provided service is not rifling through the user files. Many cloud storages like iCloud, dropbox and Google Drive can access or even provide the user information to the government of different countries in certain cases. This creates leak of trust between the user and cloud storage provider. To avoid such situation cloud access to user files should not be possible. To achieve it all files before sending to cloud should be encrypted on client side. Also private key should not be sent to cloud. If these rules are kept the user will get the desired security without trusting to cloud. Cloud service also should take a care of cases where servers get hacked and files have been damaged or removed. That can be done using File Backup Distribution system. Cloud supports user authentication in order to provide convenient way for file sharing. Data will remain encrypted when it is shared with other users. Cloud will be responsible for supplying information about how directly to connect the user devices for key exchange through the corresponding api. In order to connect api the client side application should be authenticated. Authentication will be done in 2 steps. First, the user should pass authentication through its credentials. If the 1st step is succeeded the user should also send security token which is unique for each application. After successful validation of the client application key, cloud will grant corresponding permission.

4. FILE DISTRIBUTION / SHARING

In order to provide security during sharing, files are exchanged only in encrypted way. Access to shared files is possible after the owner sends a private key for that file. Encrypted files are downloaded from cloud. Key exchange among users goes directly to prevent cloud access to it. Client application uses cloud api to get information about devices that need access to files and establish connection with them through p2p. To reject the unknown 3rd party connections through p2p, security token validation is used. Client should request from cloud application token which is generated for each authenticated client application. During connection establishment phase tokens are exchanged between the users. By using cloud api the user validates the token and based on validation rejects or accepts connection. If connection is accepted the owner sends the file private key. The private key for each file is unique and decrypts of other users file with received private key are impossible. If p2p connection is not supported on one of devices, then the clients applications exchange the key through cloud api where Diffie–Hellman algorithm is used.

5. ENHANCED FUNCTIONALITY FOR ADDITIONAL SECURITY

Vertual Keyboard

Because of big amount of keyboard malwares and to provide enhanced security for password the authentication will avoid using standard keyboard. Instead the application will suggest using a virtual keyboard or camera. Many users reject to use a virtual keyboard, because it is not convenient to use it and it takes more time than the standard keyboard. In the designed application that problem is solved with the help of user camera. To use camera for the password inputting the user should have camera with resolution of 640x480 or higher. This requirement takes care to supply wide range of possibilities for password generation. In order to authenticate image close to previously recorded one it should be recorded again by camera (it can be user fingers, face, work table…). Then the user should mark (with the help of fingers or mouse) 3 different areas on that screen, which are matching the previously marked areas. Marked pixels will be rounded and transferred to position which will contain key.

6. CONCLUSION

The paper addresses two problems in application of cryptography for cloud computing, such as: speed and limitation in usage. Noticeable improvement of efficiency of the AES cipher is achieved through parallel computing, partial encryption and Intel AES instructions. Such an improvement makes encryption process stay unnoticed for users. Also, secured encrypted files sharing is obtained. Instead of sending private keys or decrypted files through not trusted network, encrypted files' distribution and secure key exchange are realized. The method proposed prevents cloud access to raw data meanwhile maintaining data backup and sharing.

REFERENCES

[1] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing", Cloud Security Alliance, pp. 49-70, 2009.

[2] Cloud Standards Customer Council, "Practical Guide to Cloud Computing", Cloud Standards Customer Council, pp. 1-10, 2011.

[3] Keiko Hashizume1, David G Rosado2, Eduardo Fernández-Medina2 and Eduardo B Fernandez, "An analysis of security issues for cloud computing", Springer, pp. 1-10, 2013.

[4] Abbadi I and Martin, "Trust in the Cloud", Information Security Technical Report, pp. 108-114, 2011.

[5] Agarwal A and Agarwal A, "The Security Risks Associated with Cloud Computing", International Journal of Computer Applications in Engineering Sciences, pp. 257-259, 2011.

[6] Joan Daemen and Vincent Rijmen, "AES submission document on Rijndael", Federal Information Processing Standards Publication, pp. 1-51, 1998.

[7] Joan Daemen and Vincent Rijmen, " The Design of Rijndael, AES - The Advanced Encryption Standard", Springer-Verlag, pp. 238, 2002.

[8] Henning Heitk¨otter, Sebastian Hanschke, and Tim A. Majchrzak "Evaluating Cross-Platform Development Approaches for Mobile Applications", Department of Information Systems University of M¨unster, Germany, 2012.

[9] Charland, A, Leroux B, "Mobile application development: web vs. native.", Commun. ACM 54, 2011.