CRYPTOGRAFHY FOR IMPROVING THE SECURITY OF CLOUD COMPUTING

Assistant Prof. Dr. Petar Halachev
Department of Informatics,
University of Chemical Technology and Metallurgy – Sofia, Bulgaria

Abstract: *The purpose of this article is to study and analyze the security threats in cloud environment and the applicability of cryptographic systems to protect the access to information resources. The threats to information security and the corresponding means for protection in cloud environment are discussed. Based on research and analysis of asymmetric and symmetric encryption are proposed flowcharts for secure communication over the Internet when using cloud services. The speed of different encryption and decryption of algorithms in cloud environment is measured, and are given recommendations for improving security.*

Keywords: CRYPTOGRAFHY, CLOUD COMPUTING, SECURITY, VULNERABILITY

1. *Introduction*

Nowadays one of the promising directions in the development of information technology are the cloud services. Their application enables companies, educational organizations, and private individuals to reduce the cost for building and maintaining their own IT structure by providing these features to cloud service providers. In parallel, there are questions related to the safe storage and handling of confidential information, the danger of uncontrolled access by the provider of cloud services or third parties to the provided data. Solving these problems can be achieved by encrypting the data before transmission to the cloud.

Protection of information from the infringement of third parties in the way of transformation existed for millennia. Philosophers of Ancient Greece, Ancient Egypt and India have used various codes to prevent third party access to certain information. Only highly educated people were capable of encoding and decoding the information. In practical activities now are applied different cryptographic means that does not allow certain computations over the encrypted data, greatly increase the volume of data, etc., which in turn limits the usage of cloud technologies.

2. *Threats to information in cloud environments*

Ensuring information security in cloud environments is a complex and specific activity related to computer, network and information security, and includes a wide range of technologies to protect the data, the applications and the infrastructure of cloud computing.

Gartner Group [1] outline the guidelines unique to cloud-computing. When contracting with cloud-provider is better the assessment of the security of the service to be assigned to a third party - neutral company. Consumers should trust that provider, who does not hide information about their software for protection of the information in the Cloud-system. Of importance is the physical location of the cloud. The user of the cloud-service often does not know in which country the data is stored. Some countries have specific requirements for access to data by state institutions, which is subject to regional legislation. It is necessary risk assessments of the retention of data integrity, recovery, piracy, legal and legislative issues and others.

There are risks in the implementation of encryption and decryption system - for example, incorrect encryption can destroy the data on which it is applied. Even when properly used encryption complicates the processing of information and the access to it. The process of back up of data from the provider of cloud service should be examined - how often backups are done, in case of emergency how long does take the recovery of data.

Unauthorized access to cloud-resource is difficult to detect, because users often connect from many different places to a given cloud-resource. Therefore are used log files that hold information about each entry and exit of the cloud-system and about the processing of data.

Of importance is the long-term accessibility - cloud service provider can suspend operations or to be absorbed by another larger provider. The question concerning the format of data in the cloud-system and the possibility to be transferred and imported to another provider arises.

Important in the cloud-computing is the security of communication and specifically the elimination of the threat of hacker attacks: Forward secrecy, FREAK, Goto fail, Heartbleed, HTTP Public Key Pinning, Lucky Thirteen attack, Man-in-the-middle attack, OCSP stapling, Padding oracle attack. In practical conditions, most of these threats are unlikely and hardly could lead to security breaches or losses to the economy. Some of these threats are eliminated in latest versions of encryption libraries that are delivered in the form of automatic updates for users of Windows and Linux operating systems.

Real threat is the Man-in-the-middle attack [2], which may arise when the attacker is a provider of internet services. The provider may access the user name, password, and give commands on behalf of the user. Man-in-the-middle attack can be avoided through the use of secure protocols to connect to the cloud - SSL (Secure Socket Layer) and TLS (Transport Layer Security).

There are attacks by the so-called "Trojans" which are email messages sent to users from the attacker and containing malicious code - eg. code for interception of the keyboard, so further can be stolen credit card numbers, passwords, etc.

Security of cloud-services largely depends on the users of the system - if their passwords are too easy to guess, access to a system can be obtained by Brute force attack, which relies on easily recognizable passwords.

Another possible threat may occur during Denial of service attack (DOS) [3]. Many different servers, often infected with a Trojan or a virus begin to submit simultaneously countless false automated requests to a server. Attacked server cannot handle all the requests and this results in crashes and the users temporarily lose access to their data. To deal with the impact of Denial of service attack it is required the administrator to isolate sequentially the IP (Internet Protocol) addresses from which come the requests. This may take some time depending on the number of attacking machines. Often, the subject of such attacks are sites of government institutions or large corporations.

3. *Protocols for transmission of information (TLS / SSL)*

One of the key directions in the safe usage of cloud-services is to create a channel for a secure connection between the participants. Since the use of cloud services are implemented through the web it is necessary to investigate the process of communication between the client and the cloud server.

TLS and its earliest version SSL are cryptographic protocols designed to provide a secure means for communication in a computer network. They use X.509 certificates and a combination of asymmetric and symmetric cryptography. The certificates are issued by known organizations and they contain information about the identity of the company or the person to whom they are issued. There are mechanisms for verification of certificates on the client side, eliminating the possibility the client to connect to a fake server. There are different types of certification - certification of a web service, for signing applications, for identification – eg. electronic signature. In practice a stolen certificate of Adobe was used for signing applications containing Trojans and viruses [4].

There are several versions of the protocols TLS and SSL, and they are used for: browsing the web, e-mail, instant communication, voice-over-IP, securing the cloud. TLS and SSL encrypt data at the presentation level in the seven layered OSI (Open Systems Interconnection) model. In the terminology of OSI [5], TLS / SSL is initialized at the fifth session layer, while working in the sixth presentation layer.

TLS is based on the earlier specification SSL [6] developed by Netscape Communications in order to add support for HTTPS protocol to their web browser Navigator.

TLS protocol enables cloud-based applications to communicate with each other in such a way that they can prevent leaks of information or replacement of original messages.

4. Encryption of information as a means for protection

Data is encrypted by complex computational processes using 128, 512 or 1024 bit keys. This data can not be recognized in the network, because externally it is presented as a set of coded symbols. Deciphering this information is possible only by using the appropriate key and encryption algorithm.

4.1. Symmetric encryption

Symmetric encryption algorithms are AES and DES [7]. In symmetric algorithms encryption of information [8] process proceeds with a predefined key by which information is encoded [9]. After encoding the data is transmitted to the recipient or stored in the database. When is required access to the data, it is decrypted by the same key that was used in its encryption. Therefore it is necessary the sender and the receiver of data to have the same key. This method is used for transfer of data in insecure environments - for example, public servers where arbitrary participant can gain access and read the private messages of other users. Another scenario of usage of symmetric encryption is when a storage for sensitive data on a public cloud server is required. Data is encrypted, uploaded to cloud server and in a case of access to it, it is extracted and deciphered [10]. When the encryption is performed by the end user, he stores the key and use it for encryption and decryption. Security of information in symmetric encryption is relatively high, but when such process is automated by software on the presentation layer, the security of the key depends on who has access to the automating software. Automating software can also be subject to attack, as can it can be decompiled (to be obtained the source code of the application), and the key to leak to third parties.

4.2. Asymmetric encryption

In asymmetric encryption [11] are used two different keys - public and private. Keys are interconnected, but the computation of the private key only by the value of the public is virtually impossible. The private key is known only to the party that performs encryption, and the public is known to the recipients of the message. Asymmetric encryption is done by the private key, and the recipient decrypts the message by the public, while successfully deciphering guarantee that this party who has the private key has sent the message. This method is used for electronic signatures, as well as during secure Web communication that occurs over the HTTPS with SSL or the newer TLS protocols [12].

Web communication takes place between two parties - client and server. At some point, the client sends a request to the web server. If the request is to a public resource on the website, it is executed without the server to identify the IP-address of the user. The authorization is performed when the user enters their username and password on the login page of the site. The server checks in the database if such username and password exists.

If the username and password does not exists, the server issue the message "wrong password", but if such user exists, the server extracts his user ID. The user ID is a number that uniquely identifies each customer of the online system. In some cases, apart user ID is extracted and additional information - as level of user access - normal user or administrator. Retrieved information is packaged in a so-called token, which is an object consisting of three numbers - a user ID, access level and time the token expires.

This data is encrypted asymmetrically with the private key of the server, and then sent back to the client. The client saves this information (formerly cookies, and now HTML5 Local storage function of the web browser). In a subsequent request from the client (such as when the user tries to delete an advertisement), the token is sent back to the server with the request..

The server receives the request and decrypts the token by its public key. If the time of expiration of the tag is not passed, the server retrieves the user ID and the access level and performs the operation requested by the user. Advantage of this technology is that the server can be easily scaled by dividing it into two or more physical computers. Central server for authentication is not needed.

Each physical server has the public and private key pair and decrypts the tag at the time of the request. It is not necessary any of the physical servers to keep a list of the states of all clients - stateless. In this type of secure communication there is a risk third parties to obtain the token - such is the administrator of one of the servers that requests passes through. With this token the attacker can generate a request which will be executed by the server. This is known as Man in the middle attack [13]. To avoid this threat the communication must pass through the protocol HTTPS.

The HTTPS protocol relies on TLS (transport layer security) to provide a secure channel for communication between two endpoints [14].

Communication over HTTPS is protected by means of symmetrical encryption using a temporary encryption key for the session. To exchange the temporary key is used asymmetric encryption.

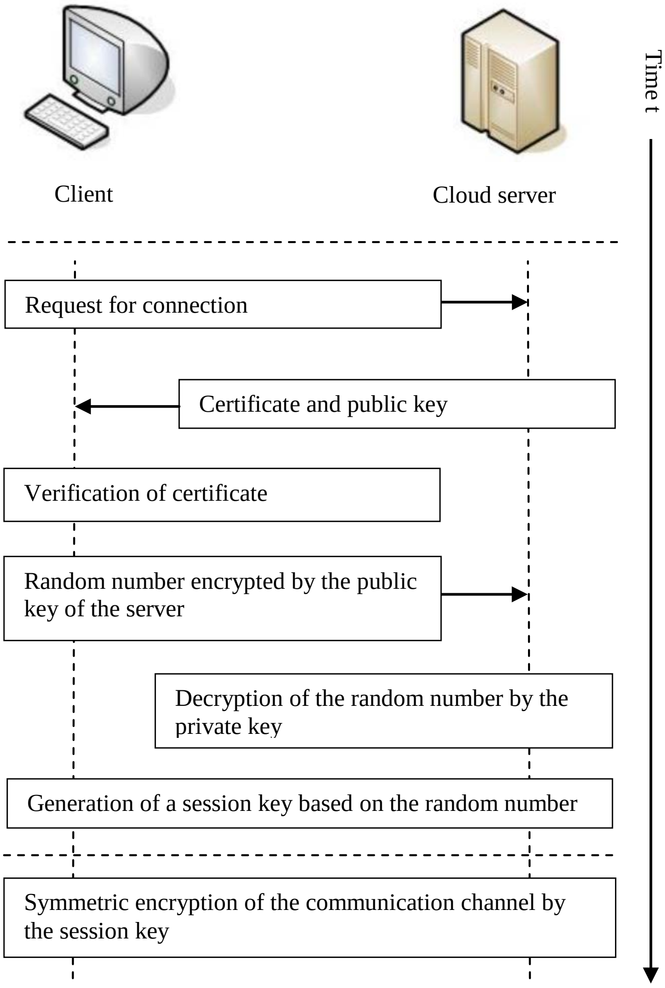
Client
Cloud server
Time t
Request for connection
Certificate and public key
Verification of certificate
Random number encrypted by the public key of the server
Decryption of the random number by the private key
Generation of a session key based on the random number
Symmetric encryption of the communication channel by the session key

Figure 1. *HTTPS Communication channel between client and server*

On Figure 1 shown the establishment of secure communication channel HTTPS between the client and the server, secured by means of asymmetric and symmetric encryption. Communication is established through a handshake procedure. During this procedure the client and server agree on various parameters of communication. Initially, the client connects to the server with a request for a secure connection by providing a list of supported encryption schemes. The server selects an encryption scheme and notify the client. Then the server sends a digital certificate that contains its name and public key. The client can connect to the organization that issued the certificate to verify its validity and the identity of the server. To generate keys for the session, the client encrypts a random number with the public key of the server and sends the result to the server. Only the server can decrypt the packet with the private key. From the random number both sides generates key for the current session by the same algorithm - and the result is that the client and the server have the same keys for current session. With this the process of handshake ends and begins connection protected by symmetric encryption until the connection is terminated. If any of these steps fail, the secure connection is not established.

After securing a communication channel, through the channel the client can send requests for public and protected resources, without the possibility the requests to be decrypted by third parties. Access to protected resources of a web application can have two dimensions: a user who wants to use a cloud web - based service (to upload photos, etc.), and the system administrator who monitors the activity of users, censors user messages; add or alter user interface elements - such as news, banners, etc., of cloud-based web application (fig. 2).

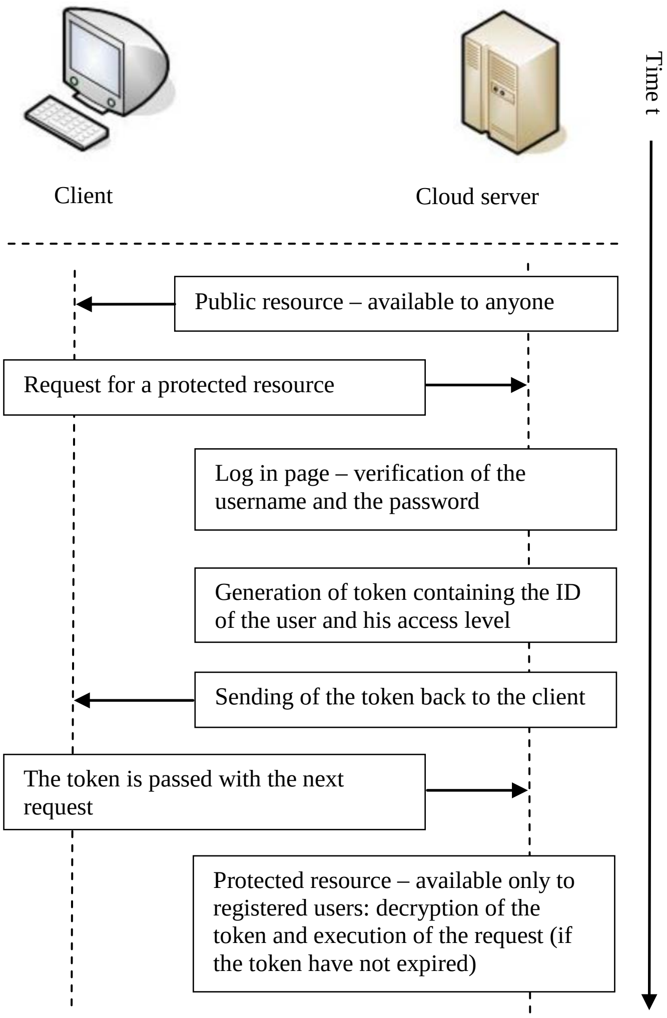
Time t
Client
Cloud server
Public resource – available to anyone
Request for a protected resource
Log in page – verification of the username and the password
Generation of token containing the ID of the user and his access level
Sending of the token back to the client
The token is passed with the next request
Protected resource – available only to registered users: decryption of the token and execution of the request (if the token have not expired)

Figure 2. *Authorization at web application level.*

In order to prevent leakage of the token to third parties, the communication passes through a security protocol - https.

There are many encryption algorithms, and they are generally divided to symmetrical and asymmetrical. In general, symmetric algorithms have significantly better performance than asymmetric. For large amounts of data, it is recommended to use a symmetric encryption and to be used an asymmetric encryption to exchange the key.

Different types of encryption algorithms have different performance, and when measured must be considered a number of factors - for example, on what hardware are running the performance tests? Given model processor may have an integrated circuit that implements some of the encryption and decryption algorithms at hardware level. Such a combination can execute the encryption algorithm very quickly, but there may be delays in other algorithms that are not implemented. Also, the speed depends on the speed of the hard disk - if overloads occur when reading large volumes of information?

When applying encryption functions in order to estimate the speed of the algorithm it must be taken into account the length of the key used for encryption. The length of the key is measured in bytes. It can be estimated at the beginning by the formula:

ByteLength = BitLength / 8;

Data with length that is greater than the length of the key can not be encrypted. Instead, such data is divided into chunks with length ByteLength-11 bytes and each chunk is encrypted separately. The encryption function adds 11 bytes checksum, so the length of the encrypted chunk is ByteLength bytes.

There are many software platforms for WEB that implement encryption functions at a program level. Such is the library of Windows - CryptoApi [15]. It can be programmed in the C++ language. The platforms PHP [16] and NodeJS [17] also have encryption libraries. Below is considered an example of realization of the RSA (Rivest, Adi Shamir и Leonard Adleman) encryption algorithm under Windows in C++ language.

To encrypt data, first it must be separated into chunks according to the length of the key, then each chunk must be encrypted, and the result of the encrypted chunks to be concatenated one after another.

For the purpose from the encryption library is used the function CryptEncrypt:

```
// Setting the size of the buffers
DWORD dwOutSize = ByteLength;
DWORD dwInSize = dwOutSize-11;
DWORD dwSize = dwInSize;
// Preparation of the buffers
TMemoryStream* chunk = new TMemoryStream;
TMemoryStream* encrypted = new TMemoryStream;
bool final = false;
do {
    if (data->Position+dwInSize >= data->Size) {
        final = true;
        dwSize = data->Size - data->Position;
    } else {
        dwSize = dwInSize;
    }
    chunk->Position = 0;
    chunk->CopyFrom(data, dwSize);
    // encryption of data
    if (!CryptEncrypt(hKey, NULL,
                        final, 0,
                        (System::PByte)chunk->Memory,
                        &dwSize, dwOutSize)) {
        // error handling
    }

    // reversing of the bytes
    // and aggregation in the result
    ReverseStream(chunk);
    encrypted->CopyFrom(chunk, 0);
} while (!final);
delete chunk;
```

Listing 1. *Sample realization of encryption with CryptoApi from Microsoft.*

Here the variable which contains data for encryption must be of type TMemoryStream. The variable *HKey* must contain the public encryption key. In a subsequent operation encrypted data is accumulated in the variable *encrypted* from where it is sent to the server.

The code listing shown above may be used to estimate the speed of the encryption of different algorithms. Further except the encryption operation can be evaluated and the decryption operation. Following algorithms were evaluated: RSA [18], LUC (Lucas sequences cryptosystem), LUCELG (EIGamal and LUC) and DLIES (Discrete Logarithm Integrated Encryption Scheme). [19].

In the tables below are the results measured at different types of encryption algorithms, and the time is measured in milliseconds and reflects the time required for encryption or decryption of one block of data with length of 1015 bits. Key which was used in the tests, respectively, has a length of 1024 bits. Lower value in column Milliseconds operation results in faster execution of the algorithm.

Algorithm	Milliseconds per operation
RSA	0.3786
LUC	0.4512
DLIES	0.4728
LUCELG	0.9245

Table 1. Speed of encryption with a key length of 1024 bits.

Algorithm	Milliseconds per operation
RSA	0.5863
DLIES	0.6720
LUCELG	0.8235
LUC	1.0905

Table 2. *Speed of decryption with a key length of 1024 bits.*

The table shows that the best speed of encryption and decryption is obtained by using the algorithm RSA, which is the reason for its relatively broad application.

5. *Conclusion*

Cryptographic means are reliable method to protect information and are applied in public offices and companies to protect information from external intervention. To encode information in the cloud-computing are used various programs incorporated in operating systems, web-browsers, email and more. Thanks to mathematical methods for transformation of information, it becomes incomprehensible by third parties. Cryptographic programs protect confidential data from unwanted access, violation of integrity or destruction.

There are cases when hackers have managed to overcome the protection of information in large companies, but the use of cryptographic means did not allowed them to benefit from the information. The encrypted data can be transmitted by e-mail, stored in the cloud or on disc media, and can be used in virtual shops, payment systems, online auctions and more. Application of methods of encrypting data using asymmetric and symmetric encryption is particularly relevant for companies that should not allow violations on customer database.

Otherwise this would lead not only to financial losses for the company, but also will reduce the image at the public. For state organizations encryption of data in some cases a matter of national security.

6. References

1. Jon Brodkin, Seven cloud-computing security risks, JULY 02, 2008, Gartner Group

2. Lidong Chen, Guang Gong, Communication System Security, 2012, ISBN-978-1-4398-4037-5, Taylor & Francis Group

3. David Dittrich, Jelena Mirkovic, Peter Reiher, Sven Dietrich, Internet Denial of Service: Attack and Defense Mechanisms, 2004, ISBN-0-13-14-7573-8

4. http://blogs.adobe.com/security/2012/09/inappropriate-use-of-adobe-code-signing-certificate.html

5. Mohammed M. Alani, Guide to OSI and TCP / IP Models, 2014, ISBN-978-3-319-05152-9, Springer Cham Heidelberg

6. Ivan Ristik, Bulletproof SSL and TLS: Understanding and Deploying SSL / TLS and PKI to Secure Servers and Web Applications, 2014, ISBN-978-1-907117-04-6

7. Christof Paar, Jan Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, 2009, ISBN-978-3-642-44649-8

8. Mahalingam Ramkumar, Symmetric Cryptographic Protocols, 2014, ISBN-978-3-319-07583-9

9. Alex Osuna, David Crowther, Reimar Pflieger, Esha Seth, Ferenc Toth, IBM System Storage Data Encryption, 2010, IBM Redbooks, Copyright International Business Machines Corporation 2010

10. Vic (JR) Winkler, Securing the Cloud: Cloud Computer Security Techniques and Tactics, 2011, ISBN-978-1-59749- 592-9

11. Mauricio Arregoces, Maurizio Portolani, Data Center Fundamentals, 2004 Cisco Systems, ISBN-1-58705-0230-4

12. James Joshi, Network Security: Know It All: Know It All, 2008, ISBN-978 -0-12-374463-0

13. Charles P. Pfleeger, Shari Lawrence Pfleeger, Security in Computing, 2003, ISBN-0-13-035548-8

14. Rolf Oppliger, Security Technologies for the World Wide Web, 2003 Artech House Inc. ISBN 1-58053-348-5

15. https://msdn.microsoft.com/en-us/library/

16. http://www.php.net - Hypertext Preprocessor

17. https://nodejs.org/en/

18. Anne L. Young, Mathematical Ciphers: From Caesar to RSA, Mathematical World Volume 25, American Mathematical society, 2000

19. Norliana Muslim, Mohamad Rushdan Md. Said, A New Cryptosystem Analogous to LUCELG and Cramer-Shoup, International Journal of Cryptology Research 1 (2): 191-204, 2009