ISSN 2278-3091
Volume 7, No.1, January – February 2018
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse01712018.pdf
https://doi.org/10.30534/ijatcse/2018/01712018

Security Protocol for Data Transmission in Cloud Computing

Dr. Abdelrahman ElSharif Karrar, Mohamed Fadl Idris Fadl
[1]Taibah University, Saudi Arabia, akarrar@taibahu.edu.sa
[2]Omdurman Islamic University, Sudan, m.fadl@outlook.com

ABSTRACT

Security is one of the significant challenges which limit users from taking the full advantage of cloud computing. Accordingly, many users have concerns about saving their sensitive data in insecure place. Therefore, we need a protocol that verifies confidentiality, authenticity, integrity, and non-repudiation of data transmission in the cloud. In this paper we will present a protocol that help to secure data access during transmission. Accordingly, users can send their data to the cloud safely and quickly. Thereby, encourage the users to take the full advantage of the cloud computing services. We used in this paper a private and public key scheme to verify data confidentiality. Additionally, we used a hash function to verify data integrity. Finally, data authenticity and non-repudiation have been verified by applying the digital signature mechanism.

Keywords: Cloud Computing, Data Transmission Security, Encryption, Elliptic Curve.

1. INTRODUCTION

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models [1].

1.1 Cloud Characteristics:

- **On-demand service.** The consumer can unilaterally provision computer services, such as email, applications, network or server, without requiring human interaction with each service provider.
- **Ubiquitous network access.** Cloud services are available and accessible through standard mechanisms that support use by various platforms such as mobiles, laptops, and tablets.
- **Independent Resource pooling.** Cloud providers can provision computing capabilities to serve many customers using a multi-tenant model, with mixed virtual and physical resources dynamically allocated and reallocated according to customer request.
- **Rapid elasticity.** Cloud capabilities can be provisioned and released, in some cases automatically. For the customer, the capabilities available for provisioning regularly appear limitless and can be designated in any quantity at any time.
- **Measured service.** Cloud services usage can be measured and reported, which is providing transparency for both the provider and consumer.

1.2 Cloud Service Models:

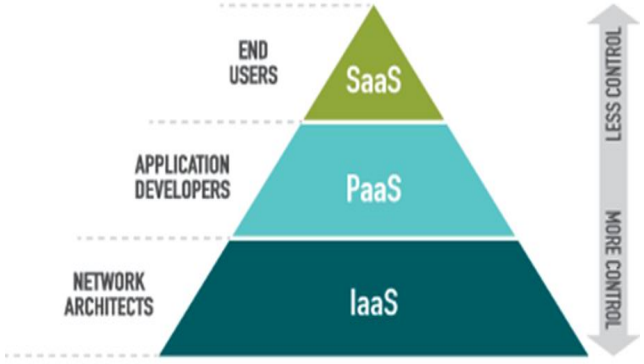

Figure 1 : Cloud Computing Service Model

- **Software as a Service (SaaS).** The services presented to the customer are to use the provider's applications that operate in the cloud. These Applications can be accessed by using the web browser or the software interface. Here, the customer does not maintain or manage the cloud infrastructure, such as the servers, network, operating systems, with the potential exception of the user-defined configuration settings.
- **Platform as a Service (PaaS).** Here, also the customer does not maintain or manage the cloud infrastructure but can uses tools managed by the provider such as programming languages, libraries, services to deploy on the cloud infrastructure.
- **Infrastructure as a Service (IaaS).** In this service model, the customer can deploy and run arbitrary software, such as operating systems and applications. Also, the customer has authority to control operating systems, storage, implemented applications, and limited control of select networking components. However, does not maintain or manage cloud infrastructure.

1.3 Cloud Computing Deployment Models:

- **Private cloud.** The cloud's provider provides infrastructure for a particular single Customer. It may be maintained, controlled, and operated by the Customer, a third party, or some combination of them, and it may exist on or off premises. [1]
- **Community cloud.** The cloud's provider provides infrastructure for exclusive use by a community of customers from groups that have shared concerns. It may be owned, controlled, and operated by one or more of the groups in the community, a third party, or some combination of them, and it may exist on or off premises. [1]
- **Public cloud.** Here, the cloud's provider provides infrastructure for public use. It may be owned, controlled, and operated by a business, academic, or government group, or some combination of them. It exists on the premises of the cloud provider [1]
- **Hybrid cloud.** The cloud infrastructure is a combination of two or more different cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability [1]

2. Related Works

This research matched with [2] and [3] on ensuring data authenticity. Furthermore, this research agreed with [4] on proving of data confidentiality, authenticity, integrity, and non-repudiation. However, the difference is the recommendation on this research, to use elliptic-curve which is use short-key length compare it to RSA that is applied a long-key length.
On the other hand, it disagrees with [2] and [3] on confirming the data integrity and non-repudiation, which is not approved on their researches.

3. Hypothesis

The research is undertaken to test the following hypothesis:

- If we use the private and public key schemes, then the data will be confidential.
- Using hash function will verify data integrity.
- If we use Digital Signature, then the data will be authentic.
- Using Digital Signature will verify non-repudiation.

4. Proposed Security Protocol

Security is one of the significant challenges which limit users from taking the full advantage of cloud computing. Accordingly, many users have concerns about saving their sensitive data in insecure place. Therefore, we need a protocol that verifies confidentiality, authenticity, integrity, and non-repudiation of data transmission in the cloud. The proposed algorithms have not focused on the main components of data security (confidentiality, authenticity, integrity, and non-repudiation); also, some researchers did not give much concern about the complexity of the algorithm they have used. The complex nature of the algorithm directly affects the speed of data access, especially when users access the cloud using limited resources devices. Hence, we need a protocol that will help in speedy and efficient secure data access.
We use in this protocol a hybrid scheme consists of the private and public key schemes to ensure confidentiality of data. The private Key scheme is used to encrypt and hide the data from unauthorized users and the user who knows the key can decrypt the data and read. But the private key scheme has lack of key distribution, which is solved here by using the public key schemes to distribute keys securely. Also, in this protocol, we prove data authenticity and non-repudiation by using digital signature scheme. Additionally, the hash function has been used to confirm data integrity.

4.1 Symmetric-key cryptography

Symmetric-key cryptography is an encryption scheme in which the sender and receiver of data share a unique key that is used to encrypt and decrypt the data. Compare this with the public-key scheme, which uses two keys - a public key for data encryption and a private key for data decryption.

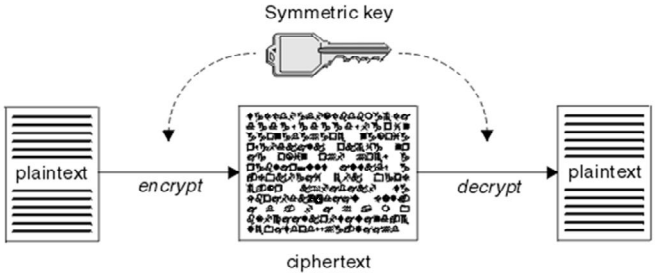

Figure 2 : Symmetric-key Cryptography

Symmetric-key schemes are simpler and faster, but their main disadvantage is that the two parties need somehow exchange the key securely. Which is avoided in the public-key scheme [5].

4.2 Asymmetric-key Cryptography

In Asymmetric cryptography secret key is divided into two portions, a public key, which can be distributed to anyone, and a private key. That should be saved secret.

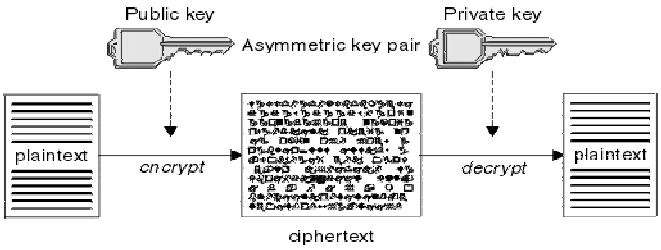

Figure 3 : Asymmetric-key Cryptography

In this scheme, someone with the public key can encrypt a data, providing confidentiality, and then only the one who had the private key can decrypt the data.

4.3 Diffie–Hellman Key Exchange

The Diffie–Hellman key exchange (DHKE), proposed by Whitfield Diffie and Martin Hellman in 1976, was the first asymmetric scheme published in the open literature [6]. the two inventors were also influenced by the work of Ralph Merkle. It gives a practical solution to the key distribution problem; it allows two parties to obtain a shared secret key by transmitting over an insecure channel.

4.4 Digital Signature

A digital signature is an authentication mechanism that enables the message sender to attach a code which acts as a signature. The signature is formed by taking the message hash and encrypting the message with the sender's private key. Which proves the source and integrity of the message [7].

4.5 Cryptographic Hash Function

A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value h = H(M). A "good" hash function has the property that the results of applying the function to a large set of inputs will generate outputs that are fairly distributed and apparently random. In general terms, the principal object of a hash function is data integrity [7].

4.6 AES algorithm

AES algorithm is declared by NIST as new Encryption Standard in 2001. The AES block and key size vary between 128, 192 and 256 bits. However, the AES standard only requests block size of 128 bits. The internal rounds of the cipher vary between 10, 12, and 14 according to the key length [6].
The following table compares AES with DES.

Table 1: Comparison between AES and DES encryption algorithms

	DES	**AES**
Developed	1977	2000
Key Length	56 bits	128, 192, or 256 bits
Cipher Type	Symmetric block cipher	Symmetric block cipher
Block Size	64 bits	128 bits
Security	Proven inadequate	Considered secure

4.7 Elliptic Curve Cryptography

Elliptic Curve Cryptography lies under the category of asymmetric encryption algorithms. In elliptic curve cryptography, points on elliptic curve are used to derive a public key i.e. a generator point in an elliptic curve group is agreed upon by the communicating parties. By multiplying this generator point by a randomly generated number, corresponding private key is generated. In case, the generator point and public keys are compromised, it is very hard problem for the intruder to get the private key by backtracking [9].

The following table compares Elliptic Curve with RSA.

Table 2: Comparison between Elliptic Curve and RSA

Security level of symmetric key	80	112	128	192	256
EC key length (bits)	160	224	256	384	512
RSA key length (bits)	1024	2048	3072	7680	15360

4.8 Tamarin Prover

The Tamarin prover is a powerful tool for the symbolic modeling and analysis of security protocols. It takes as input a security protocol model, specifying the actions taken by agents running the protocol in different roles. Tamarin can then be applied to automatically construct a proof that, even when arbitrarily many instances of the protocol's roles are interleaved in parallel, together with the actions of the adversary, the protocol fulfills its specified properties [8].

4.9 Protocol conclusion

As we mentioned earlier, in this protocol we used a hybrid scheme to secure data transmission in the private cloud that consists of a private key scheme, a public key scheme, and hash function. Accordingly, the protocol steps will be as the following:

- First, the users exchange the keys using Diffie-Hellman key exchange scheme.
- Second, they calculate the session keys, which will use to encrypt the messages.
- Third, the sender calculates the hash value of the message using the hash function scheme.
- After that, he encrypts the message using the shared session key.
- Then he signs the message using his private key and attaches the signature to the message.
- Finally, he sends the encrypted message with the signature and the hash to the other party.
- On the other hand, first, the user receives the message and tries to decrypt it using the shared session key.
- Second, he calculates the hash value of the received message and compares it with received hash. If they equal, he will accept the message. Otherwise, he will ignore it.
- After that, he verifies the sender's signature using the sender's public key. If the verification is correct, he can accept the message; else the message will be ignored.

5. RESULTS

This section will describe how the founded results have proved the hypotheses as follows:

- **Hypotheses NO 1:** If we use the private and public key schemes, then the data will be confidential.

If the data sent in clear text, anybody can interrupt it and read its content. Thus, the data is not confidential. This issue is solved here by encrypting the data before sending to the cloud. The main schemes for data encryption are private-key and public-key systems. But if we use private-key scheme for data encryption only, it will be difficult to exchange the secret keys between the two parties securely. That is why we use a combination of the private-key and public-key schemes to provide data confidentiality. Furthermore, the results show that the data is confidential.

• **Hypotheses NO 2:** Using hash function will verify data integrity.

The data can be altered and modified while transmission, which affects data integrity. In this research, we use a hash function to solve this issue, and the results verify data integrity.

• **Hypotheses NO 3:** If we use Digital Signature, then the data will be authentic.

When the transmitted data sent without a signature, the receiver cannot guarantee that the data has been sent from an authentic user. The digital signature can master this issue and verify data authenticity. Furthermore, the results show when the sender signs the data, the receiver can validate that the data sent from an authentic user.

• **Hypotheses NO 4:** Using Digital Signature will verify non-repudiation.

Non-repudiation issue occurs when the two parties own the same secret keys. Accordingly, no one can verify who has sent the data. Therefore, one of the solutions to this issue is signing the data before transmitting. The sender can use his secret private key to sign the data, and no one can send the data alternatively which will solve non-repudiation issue.

6. RECOMMENDATIONS

Here is a summarized list of our recommendations for deploying this protocol.

In order to provide data confidentiality:

1. The transmitted data in the cloud should encrypt under a shared key computed by the two parties. Preferably AES should be chosen because it is a standard private key scheme. We recommend in this protocol to use 128-bit key lengths for data encryption because it is difficult to brute force this key length using the available resources for today.
2. We recommend using the elliptic curve for the public key scheme because we can use a 160-bit key which is required low bandwidth and computation resources compare it to 1024 bit for RSA. Also, if we use a 256-bit for maximum security, we need 15360-bit to get similar security level on RSA.
3. Also, we recommend using the public key certificate to prove the ownership of the public key. Otherwise, a man in the middle attack can exist.

In order to provide data integrity:

In this protocol, we recommend using SHA-3 as a hash function to verify data integrity and determine whether data has changed or not. Also confirms that data received are exactly as sent (i.e., contain no modification, insertion, deletion, or replay). Because MD5 and SHA-0, both of which have been broken, SHA-1 is considered insecure and has been phased out for SHA-2. SHA-2, particularly the 512-bit version, would appear to provide unassailable security, we recommend using SHA-3 256-bit as a hash function for this protocol.

In order to provide data Authenticity and non-repudiation:

1. In this protocol, the transmitted data should be signed by the sender private key to provide data authenticity. We recommend using elliptic curve digital signature for signing the data. As we mentioned earlier, the main benefit guaranteed by elliptic curve cryptography is a smaller key size, that reducing storage and transmission requirements.
2. We recommend using cryptographic nonce to guarantee that old communications cannot be reused in replay attacks. They can also be helpful as initialization vectors and in cryptographic hash functions.

7. FUTURE WORKS

Many different adaptations, tests, and experiments have been left for the future due to lack of time; for example, we need to test the recommendations of the proposed protocol in a real cloud environment to measure the time and compare the results with other protocols time results.

8. CONCLUSION

Security is one of the significant challenges which limit users from taking the full advantage of cloud computing. The proposed protocol aims to secure data transmission in cloud computing and verifies data confidentiality, authenticity, integrity, and non-repudiation. We have used private and public key schemes to verify data confidentiality. Additionally, we have used the hash function to verify data integrity. Data authenticity and non-repudiation have verified by apply digital signature.

We recommend using AES for data encryption and elliptic curve as public-key scheme. Moreover, we recommend SHA-3 as a hash function and elliptic curve digital signature for signing the data.

REFERENCES

1. Mell, Peter, and Timothy Grance. "**The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology.**" http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (January 29, 2018).
2. Gampala, Veerraju, Srilakshmi Inuganti, and Satish Muppidi. 2012. "**Data Security in Cloud Computing with Elliptic Curve Cryptography.**"

International Journal of Soft Computing and Engineering (23): 2231–2307. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.456.5840&rep=rep1&type=pdf (May 10, 2017).

3. Tirthani Ganesan, Neha R. "**Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography.**"
4. As 'habi, Keivan, Arman Vafabakhsh, and Saeed Borji. 2016. "**DATA TRANSMISSION SECURITY IN CLOUD COMPUTING.**" *Indian Journal of Fundamental and Applied Life Sciences* 6: 2231–6345.
5. Beal, Vangie. "**What Is Symmetric-Key Cryptography? Webopedia Definition.**" : symmetric_key_cryptography. http://www.webopedia.com/TERM/S/symmetric_key_cryptography.html.
6. Paar, Christof, and Jan Pelzl. "**Understanding Cryptography: A Textbook for Students and Practitioners.**"
7. Stallings, William. "**Cryptography and Network Security (4th Edition).**" http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5680/material-cripto-seg/2014-1/Stallings/Stallings_Cryptography_and_Network_Security.pdf (May 20, 2017).
8. "**Tamarin-Prover Manual Security Protocol Analysis in the Symbolic Model.**" 2017. https://tamarin-prover.github.io/manual/tex/tamarin-manual.pdf (November 18, 2017).
9. Ahmad, Irfan, Irfan Ahmad, and Muhammad Waseem. 2016. "**Implementation of 163-Bit Elliptic Curve Diffie Hellman (ECDH) Key Exchange Protocol Using BigDigits Arithmetic.**" 5(4). Retrieved From http://www.warse.org/IJATCSE/static/pdf/file/ijatcse08542016.pdf