International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 5 (2017), pp. 907-916
© Research India Publications
http://www.ripublication.com

An Attribute based Authentication protocol with Quantum key cryptography in cloud servers

[1]B.N.V. Madhu Babu and [2] Dr. K. Rajasekhara Rao

[1]Research Scholar, Rayalaseema University, Kurnool, Andhra Pradesh, India.

*[2] Director, Usha Rama College of Engineering and Technology, Telaprolu,
Andhra Pradesh, India.*

Abstract

In these days, cyber-crimes are increased drastically it resembles the diminishing of security in sensitive information, to improve the security, cryptography is requires and it plays a vital role. Thus authentication is of at most significance as number of hackers who seek to fraud into authorized users account to obtain sensitive information is increasing; in existing system the identity based secure authentication [1] was implemented in python language which is not a user friendly. In the same way quantum cryptography key distribution [QCKD] [2] is based on several assumptions. Here all the keys are exchanged with the photons so it does not solve all the problems. Our proposed scheme provides essential security requirements and we achieve mutual authentication. At finally this paper proposes an Attribute Based Authentication Protocol for Cloud Server Architecture. Cryptographic key plays a vital technology for securing the privacy and confidentiality in the field of networks.

Keywords: Authentication, Cloud computing, QCKD, Cryptography, Attribute, Identity, Mutual Authentication.

I. INTRODUCTION

Cryptography focused on message confidentiality [15] (i.e., encryption)—conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge. Encryption attempted to ensure secrecy in communications, such as those of spies, military leaders, and diplomats. In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, and interactive proofs and secure computation, among others. The goal of the cryptography is to protect private communication in the public world. The assumption is that two entities wanting to communicate.

Beyond confidentiality - ensuring the secrecy of communication - cryptography is used for many other purposes, such as:

Authentication, Integrity checking [16] - Sender can generate a checksum of the message. Receiver can either extract it from the message or recalculate it and verify that the message has not been changed.

Non-repudiation [17] - if sender signs the message she cannot deny later that she sent it, because no one else could generate that same signature.

Exchanging a secret with someone you have never met before, in a room full of people Proving to someone you know a secret without giving it away Sending secret messages to any m out of n people so that only those m can retrieve them and the rest cannot Sending secret messages to a group of N people, that can be retrieved only if M people work together.

There are three main types of cryptographic functions that are the building blocks of security:

Symmetric cryptography [18] – sender and receiver know the same key and use it for encryption and decryption.

Symmetric crypto can be used to ensure secrecy - sender and receiver exchange the secret key and use it to communicate privately.

It can also be used for secure storage - sender encrypts the files she stores in the cloud. If the cloud is compromised no one can read her files.

Symmetric crypto can also be used for authentication, aka proving that you know a secret without revealing it. Sender and receiver want to prove to each other they know the same secret. Sender chooses a random number and encrypts it with the key.

Receiver decrypts it and sends it back. Then receiver chooses a secret number and encrypts it and sender decrypts it and sends it back. Why do we have to do this twice? Who is assured of what in each round?

Asymmetric cryptography [19] – sender has a pair of keys - a private key known only to her and a public key that anyone can find out. She can encrypt with one key from a pair (any one) and decrypt with another. Asymmetric crypto can do whatever symmetric crypto can but much slower (about 1,500 times slower). However it can also provide some extra functionality.

In symmetric crypto secret communication between sender and receiver is only possible if they know the same secret key. But how do they find out this key? They can use asymmetric crypto to exchange it. Receiver could advertise his public key on his Web page. Sender could use it to encrypt a secret key and send it to receiver - only he will be able to retrieve it.

If sender wants to authenticate receiver she/he encrypts a message with his public key and he decrypts it and sends it back. Sender can do all this without storing any secret information.

If Sender orders something from receiver and signs every message by encrypting it with her private key, anyone can verify her signature and no one can forge it. This ensures non-repudiation - sender cannot deny she sent the message.

Hash functions [20] - these are publicly known functions that reduce a large message to a fixed-size hash, applying a non-linear transformation (aka one-way hashes or message digests). Cryptographic has functions have several important properties:

One-way: knowing M it is easy to compute H (M) but knowing H (M) it is impossible to compute M (except using brute-force)

Collision-free: knowing H (M) it is very hard to find M1 so that H (M1) = H (M)

Collision-resistance: it is hard to find M1 and M2 so that H (M1) = H (M2)

Hash functions are useful to prove message integrity. One can hash the message and display its hash somewhere publicly. When the recipient receives the message they can calculate its hash, compare it with the public one and verify that the message has not been changed. What if I wanted to send the hash with the message? Can I use symmetric or asymmetric crypto to help me generate a signed hash of the message that I can send with the message? How?

Hashing a large message requires the hash algorithm to break it into chunks, and combine each chunk with the hash of the previous chunks to generate new hash. This process is shown in the Fig 1.

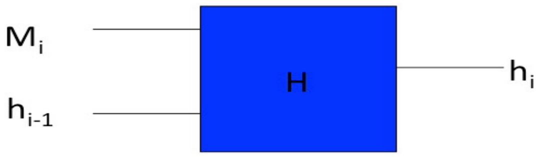

Figure 1. Hash function

Two main cryptographic techniques are substitution and permutation.

Substitution replaces chunks of the message with other chunks according to some mapping (e.g., replaces one letter with another letter). Transposition shuffles the chunks of the message around. Substitution changes the characters in the message so to hinder frequency analysis (e.g., if "the" is the most frequent trigram in English and you use 4-letter chunks and encrypt "the " into "abcd" and "ther" into "kyzh" the frequency analysis will be made more difficult). Transposition ensures that placement of the plaintext differs from the placement of the corresponding cipher text, thus if one knows that each message starts with "HELLO" they have to detect where these characters are in the cipher text. It also dissipates the redundancy of plaintext in cipher text, e.g., seeing "morn" you can guess the rest of the letters in that word but seeing "nmro" makes that hard.

A. Symmetric methods

Symmetric encryption is also known as private-key cryptography [3], and is called so because the key used to encrypt and decrypt the message must remain secure, because anyone with access to it can decrypt the data. Using this method, a sender encrypts the data with one key, sends the data (the cipher text) and then the receiver uses the key to decrypt the data.

B. Hashing

Hashing creates a unique, fixed-length signature for a message or data set. Each “hash” is unique to a specific message, so minor changes to that message would be easy to track. Once data is encrypted using hashing, it cannot be reversed or deciphered.

C. Identity Based Encryption

Identity-based encryption (IBE) is an important primitive of ID-based cryptography. As such it is a type of public-key encryption in which the public key of a user is some unique information about the identity of the user. The receiver obtains its decryption key from a central authority, which needs to be trusted as it generates secret keys for every user by using private key generator (PKG) [4].

D. Limitations:

Because the Private Key Generator (PKG) generates private keys for users, it may decrypt and sign any message without authorization. This implies that IBE systems cannot be used for non-repudiation.

A secure channel between a user and the Private Key Generator (PKG) is required for transmitting the private key on joining the system. Here, a SSL-like connection is a common solution for a large-scale system. It is important to observe that users that hold accounts with the PKG must be able to authenticate themselves. In principle, this may be achieved through username, password or through public key pairs managed on smart cards.

IBE solutions may rely on cryptographic techniques that are insecure against code breaking quantum computer attacks.

II. TERMINOLOGY

A) Quantum Key Distribution:

Quantum key distribution (QKD) uses quantum mechanics to guarantee secure communication. It enables two parties to produce a shared random secret key known only to them, which can then be used to encrypt and decrypt messages. It is often incorrectly called quantum cryptography, as it is the most well-known example of the group of quantum cryptographic tasks.

An important and unique property of quantum key distribution is the ability of the two communicating users to detect the presence of any third party trying to gain knowledge of the key. This result from a fundamental aspect of quantum mechanics. The process of measuring a quantum system in general disturbs the system.

Quantum Key distribution is only used to produce and distribute a key, not to transmit any message data. This key can then be used with any chosen encryption algorithm to encrypt or decrypt a message, which can then be transmitted over a standard communication channel.

Even quantum cryptography doesn’t solve all of cryptography .The keys are exchanged with photons, but a conventional mathematical algorithm takes over for the actual encryption.

B) Attribute Based Encryption:

Attribute-based encryption is a type of public-key encryption [5] in which the secret key of a user and the cipher text are dependent upon attributes (e.g. the country in which he lives, or the kind of subscription he has). In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text. A crucial security aspect of Attribute-Based Encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access.There are mainly two types of Attribute-Based Encryption schemes: Key-Policy Attribute-Based Encryption (KP-ABE) and Cipher text-Policy Attribute-Based Encryption (CP-ABE).

C) Key-policy Attribute Based Encryption:

The key-policy attribute-based encryption scheme of the attributes has been proposed. Encrypted data is described by a set of attributes, and access rule contained in the user's private key. If a set of attributes of data matches the structure of access to the user's private key, the data can be decrypted. Attributes of encrypted data correspond to the structure of the access user's private key, so the user can decrypt the data. The encryption algorithm is different from the original version of the ABE generating the private key and accordingly decrypts: the user's private key is generated by according to the structure of the necessary access.

D) Cipher text-policy Attribute-based Encryption (CP-ABE)

Cipher text-policy attribute-based encryption, in measures of IEEE seminar on Security and Privacy proposed encryption scheme based on cipher text-policy attribute-based encryption. Policy to access data not contained in user's private key, and the encrypted data itself (cipher text). Private Key corresponds to a set of attributes. If the attributes contained in the user's private key corresponding to the structure of the cipher text access, the user can decrypt the data. The main purpose of this algorithm is to achieve such a situation that few people, being in collusion, can decrypt the data only when at least one of them could do on their own The algorithm consists of four steps: preliminary steps, encryption, decryption and key generation. Preliminary action is to describe the security settings and access attributes. The output is a public key and a master key. The encryption algorithm accepts as input the public key, data to be encrypted, and access structure. The data will be encrypted so that only the user who has the necessary set of attributes, which in turn satisfy the structure of access can decrypt the data. We assume that the encrypted text implicitly contains Key generation. Key generation algorithm accepts as input a universal key and a set of attributes. The output is a secret key. Decryption algorithm accepts as input the public key cipher text and a secret key. If a set of attributes contained in secret key satisfies structure of access, the data will be decrypted. The main feature of the scheme is to facilitate key management and cryptographic access.

III. SYSTEM MODELS

In 2007 J. Bethencourt, A. Sahai, and B. Waters has worked on security and privacy in Cipher text-policy attribute-based encryption [11]. Set of attributes were generated a user’s private key to access the encrypted data. They can decrypt the data if the structure of cipher text is equivalence to private key of the user’s.

According to the bilinear cyclic group function [12], we briefly review about the imperative facts of bilinear maps and bilinear map group’s .Let M and M’ be two (multiplicative) cyclic groups of prime order q and let m be a generator of M. An alternative pairing of Weil pairing is bilinear mapping.

i.e; $\hat{b} : M \times M \rightarrow M'$

Therefore,

Bilinear: for all $x,y \in M$ and $x',y' \in Z$

We have $\hat{b}(x^{x'}, y^{y'}) = e(x, y)^{x'y'}$.

Where M is a bilinear group if the group operation in M can be computed resourcefully and there exists a group M’ and an efficiently computable bilinear map $\hat{b} : M \times M \rightarrow M'$ as above.

In An attribute based encryption scheme (ABE), each user is recognized by a set of attributes.

Let us assume that set of universe attributes can be partitioned in to ‘n’ disjoint sets. Each set will be observed by a unique authority in this there is only one central authority and ‘a’ attribute authorities. This Trusted central authority does not observes any attribute sets.

Note: Assume that Au to denote the attribute set of user u and A_C to denote the attribute set of a cipher text.

A_{nu} and A_{nC} are the attributes handled by authority n in the attribute sets of the user and the cipher text respectively.

A Multi Authority ABE [13] system is composed of ‘n’ attribute authorities and

one central authority. Each attribute authority is also assigned a value d_n.

The system uses the following 5 steps to implement Multi Authority based ABE:

They are:

1. Setup
2. Attribute key generation
3. Central key generation
4. Quantum key distribution
5. Encryption
6. Decryption

1. Setup:

It is an arbitrary algorithm and it invokes by trusted third party (e.g. Central authority). It accepts security parameter as input, public key, secret key pair as a output of the attribute authorities, and also it results a system public key and master secret key which will be used by the central authority.

2. Attribute Key Generation:

This algorithm bound by an attribute authority. It accepts authority’s secret key, d_n as a authority value, global identifier of the user and a group of attributes in the authority’s domain A_{nC}. It result secret key for the user.

3. Central Key Generation :

This algorithm can run by the trusted third party. And master secret key and a user’s GID [14] as input and yields secret key for the user.

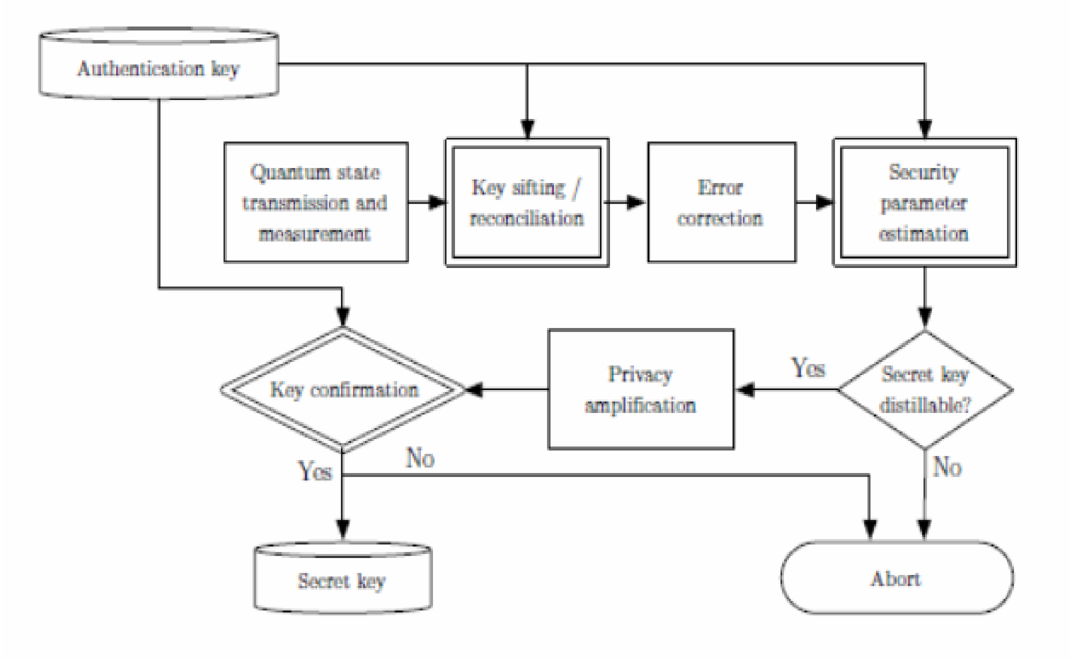

Figure. 2 Flowchart of QKD Protocol, Source [DS09]

4. Quantum key distribution

Quantum key dissemination (QKD) utilizes quantum mechanics to ensure secure correspondence. It empowers two gatherings to create a mutual irregular mystery key known just to them, which can then be utilized to scramble and unscramble messages. It is regularly erroneously called quantum cryptography.

5. ENCRYPTION :

The encryption mechanism is performed at sender side. It accepts input as a set of attributes for each authority, a message, and public key of the system. Outputs the cipher text.

6. Decryption:

The decryption algorithm is performed at user side. It takes cipher-text as a input, which was encrypted under attribute set A_C and decryption keys for an attribute set A_u. Outputs a message m, if $|A_{nC} \cap A_{nu}| > d_n$for all authorities 'n'.

IV. CONCLUSION

In this paper, integrated multi authority attribute-based encryption schemes: ABE that by using a variety of access strategy in the cloud system. We are aiming to integrate a system with Quantum key distribution along with attribute based encryption that gives a dual authentication in cloud servers. We then attempt to generate Quantum key by considering Shannon proof with symmetric encryption with One Time Pad (OTP), with a key that is non-repeating and never reused. This system helps information to maintain more in safe hands and protected, with the external attackers. The implementation details will be presented in the next paper.

V. REFERENCES

[1] International Journal of Private Cloud Computing Environment and Management "Identity based secure authentication scheme based on Quantum Key Distribution for cloud computing" by Geeta Sharma1 and Sheetal Kalra1 1 Department of Computer Science and Engineering, Guru Nanak Dev University.

[2] https://en.wikipedia.org/wiki/Quantum_key_distribution

[3] https://en.wikipedia.org/wiki/Key_(cryptography)

[4] https://simple.wikipedia.org/wiki/Key_generation

[5] https://en.wikipedia.org/wiki/Public-key_cryptography

[6] "A Survey on Attribute Based Encryption Scheme in Cloud Computing" by Minu George, Dr. C.SureshGnanadhas, Saranya.K published in International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 11, November 2013

[7] "Multi-Authority Attribute Based Encryption" by Melissa Chase Computer Science Department Brown University Providence , RI 02912 mchase@cs.brown.edu.

[8] "Analysis and Security based on Attribute based Encryption for data Sharing" by Ms. Snehlata V. Gadget Dept. of Computer Engineering, University of Pune,

[9] Pune, India published in International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-3, Issue-3) March 2014

[10] [9]. "Threshold Cipher text Policy Attribute-Based Encryption with Constant Size Cipher texts" by Aijun Ge, Rui Zhang, Cheng Chen, Chuangui Ma, and Zhenfeng Zhang.

[11] "Privacy-Preserving Decentralized Cipher-Policy Attribute-Based Encryption" by Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, Jiangsu 210003, China.

[12] https://en.wikipedia.org/wiki/Attribute-based_encryption

[13] Theory.stanford.edu/~dfreeman/cs259c-f11/final papers/vrf.pdf

[14] cs.brown.edu/~mchase/papers/multiabe.pdf

[15] https://en.wikipedia.org/wiki/Universally_unique_identifier

[16] Securitycerts.org/review/cryptography-confidentiality.htm

[17] https://www.usenix.org/.../integrity-checking-cryptographic-file-systems-constant

[18] Searchsecurity.techtarget.com/definition/nonrepudiation

[19] https://www.ibm.com/support/knowledgecenter/SSB23S_1.1.0.14/.../s7symm.html

[20] Searchsecurity.techtarget.com › Encryption technology › Network security

[21] https://simple.wikipedia.org/wiki/Cryptographic_hash_function