Developing Secure Cloud Storage Through Quantum Secure Socket Layer Protocol

[1]N. Pandeeswari, [2]R. Sivakami, [3]R. Karuppathal

[1]Associate Professor, [2,3]Professor, PSNA College of Engineering and Technology

Dindigul, India. [1]pandeeswari@psnacet.edu.in

Abstract: Cloud computing is the recent technology promotes the users to store their sensitive data in external storage system through Internet. Online storage may cause many new security threats in cloud environment. The cloud storage system suffers from the security risks such as data breach, data loss, data leakage and etc. The proposed scheme exclusively concentrates on one of cloud security risks, data leakage due to the loss of cryptographic keys .To protect against data leakage; it is extensive to suggest secure key distribution technique through a secure channel which detects eavesdropping. The proposed work uses secure socket layer quantum key distribution (QKD) protocol to transfer the secret key between users and key server in cloud environment .One of the QKD protocols, Bennett and Brassard 84 (BB84) protocol is used to generate the secret key. This work uses the key as one-time pad for encryption/decryption which ensures unconditional security. Extensive theoretical and experimental analysis exhibits that the proposed method is secured against data leakage and detects man-in middle attack and eavesdropping while exchanging secret key.

Key words: authentic channel, data leakage, Mutual authentication, One-time pad, Quantum key distribution protocol.

I. INTRODUCTION

Cloud computing has emerged as a large pool of computing resources, storage, network, security, programming and execution environment that is provided on pay per use basis. Cloud computing provides the on demand services over Internet. Virtualization [34], on-demand services, rapid elasticity, instant service and pay-for-use are the characteristics of cloud computing. The cloud system has global remote servers to store and process data (everything is done in the cloud). Cloud storage promotes users to store their sensitive data without worrying about any local maintenance of data. Preserving user's sensitive data [33] safe and secure is the obvious challenge in rapid development of today's information technology. Storing sensitive data onto cloud storage causes severe concern over security problems [31,32]. While organizations use the cloud to store their data, the secure network access to cloud is significant. Obviously the encryption algorithm is the best metric to measure the Information security. In situations, where encryption is used as for data confidentiality promise, the key management is the critical issue in information security. This paper particularly discuss about data leakage due to the loss of secret key where insecure key distribution through an insecure channel. SSL offers a secure [28] channel which prevents prying eyes from reading information and verifies authentication [28] to identify the server and client involved. Besides, the cloud service provider can be compromised by any adversaries.

Hence the proposed scheme uses a separate key server to take care of the key management tasks. The secure socket layer is the network security protocol which provides encrypted channel [1, 27] for private communication between the peer entities. The accurate implementation of SSL can protect key material as it is being transmitted from key server to client and vice versa. The SSL protocol uses public key cryptography based key distribution techniques namely Diffie-Hellman key exchange algorithm and RSA key exchange algorithm to generate and distribute the secret key in its handshaking phase is replaced by the quantum key distribution protocol [1]. The traditional key distribution techniques transferring discrete information does not give any indication about eveasdropping [7].So the quantum key distribution protocol is used to generate and distribute the secret key which can detect eavesdropping [10].

The quantum cryptographic technique prevents hackers from reading confidential information of users via Internet. As discussed in [2], the quantum cryptographic techniques are secure against quantum computer attack. To ensure secure authentic communication in cloud, the proposed work uses SSLQKD [1] protocol to distribute the key. In this proposed work one of the QKD protocols, Bennett and Brassard 84 (BB84) [2] is used with SSL to provide security in cloud storage system. This work uses quantum

 DOI : 10.35291/2454-9150.2020.0182

key distribution protocol to generate a secret key and securely transmits over an authentic channel. The secret key generated is used as one-time pad that ensures unconditional security [16] and SSL handshake protocol provides mutual authentication [1] to authenticate SSL enabled client and key server. This scheme uses the SSL connection only to generate and distribute the key between key server and user.

II. SECURE CLOUD ARCHITECTURE

The secure cloud environment model is represented in Fig.1. The cloud storage provides a convenient platform for storing user's information onto the cloud. Data security specifically depends on the secrecy of the shared secret key. Consequently, the proposed work aims to generate a secret key using secure key generation technique through a secure channel .For every data uploads at different times the user has to generate a new secret key with the key server.This work uses a separate key server to take in part of key generation and key management. Assume that key server managed by cloud administrator cannot be compromised. Hence the secret key shared by key server and client cannot be modified. The secure cloud architecture is shown in fig 1.

The keyserver stores the secret key of every user with their identity and public key.The secret Key information of different users is encrypted by keyserver's private key(No one can trace it.).To download data from the cloud server;in case ,if the user missed his secret key the user can contact the key server by giving his public key id and user id provided by key server.The key server verifies the details given by user and gives the corresponding secret key encrypted by the user's public key.

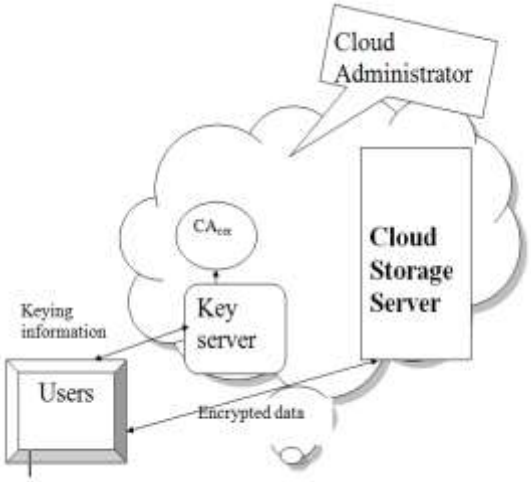

Fig 1. Secure Cloud Architecture

The user can decrypt it and collects his secret key to decrypts his data.This model restricts that only the owner of the data can do either encryption or decryption where the users can use this secure envirnoment to store their private information not to reveal anyone.. The sample database maintained by key server is shown in table which encrypted using keyserver's secret key.

user id	public key user	preshared key	secret key
uid13	15678	1001	100111100
uid15	45622	10100	1001100100
uid17	678999	101010	1010100100
uid19	795395	101001111	11111001001

Figure 2. Sample database of keyserver

III. SSLQKD PROTOCOL: SECURE KEY DISTRIBUTION

The SSLQKD protocol works as follows. The SSLQKD session starts with Quantum SSL handshake protocol .Before starting the communication the client and server has to share a key which is used for unconditional authentication [15] and to generate shared secret key. Assume that there is no preshared key for the new client with cloud server initially. The new user has to share his key with cloud server using server's public key which is signed by the certificate authority (CA).The CA is controlled by key server.

The SSL certificate is in the form:

$$cer = E(PR_{CA}, (ID_{ser} | PU_{ser}))$$

This work discusses the two different protocol designs; one is without using preshared secret key (SSLQKD-I) for the new client and cloud sever and the other one uses preshared secret key (SSLQKD-II) to generate secret key. The algorithm explains how SSLQKD protocol works. The session starts with the handshake protocol to instantiate the security parameters between client and server.

Algorithm: SSLQKD protocol

1. /**… Session starts with handshake between client and key server.
2. Handshake uses QKD protocol to distribute the key...*/
3. *Client: Hello (client: Random bits) → Server.*
4. *Server: Hello (server: Random bits) → Client.*
   Execute step (5) or (6)
5. / * For the initial session with cloud server */
   //SSLQKD-I
   i. *Server: Send (SSLcertificate) -> Client.*
   ii. *SSLcertificate:*
   $cer = E(PR_{CA}, (ID_{ser} | PU_{ser}))$
   iii. *Decrypt (SSLcertificate) = PUserver.*
   iv. *Client: Encrypt (Shared Key) → Server*
   v. *Server: ACK (Shared Key) → Client.*
   vi. *Goto step (7)*
6. /* If a key is PreShared between Client and Server */
   //SSLQKD-II

 DOI : 10.35291/2454-9150.2020.0182

i. *Server: Hash (PreSharedkey) → Client*
ii. *Client: Verify (PreSharedKey) → Server.*
iii. *If (verify (Hashcode) ==true)*
iv. *Client: ACK (positive) → Server.*
v. *Goto step(7)*
vi. *else*
vii. *Ignores the connection.*
viii. *Goto step (5).*

7. *Server: Enco= polarizationbasis (000100….0010) → client.*
8. *Client : Measure_polarizationbasis(Enco)*
9. *Confirmation: Generated key.*
10. *Server: ChangeCipherSpec, TLSFinished → Client.*
11. *Client: ChangeCipherSpec, TLSFinished → Server.*
12. *if (Client(TLSFinished) == Server (TLSFinished))*
    a. *Client and Server mutually authenticated.*

Key Server's choice of random bits	1	1	0	1	0	0	1	0	1	1	1	1	0	0
Key Server's choice of random basis	+	+	+	×	×	+	×	×	×	×	+	+	+	+
Server sends	_	_	׀	\	/	׀	\	/	\	\	_	_	׀	׀
Client's choice of random basis	+	×	+	+	×	+	×	+	×	×	+	+	+	+
Client's string	1	0	0	1	0	0	1	1	1	1	1	1	0	0
Same basis?	Y	N	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Sifted bits	1		0		0	0	1		1	1	1	1	0	0
Public discussion														
Compare the subset of sifted bits			0				1							
Comparison result			Y				Y							
Secret key generated	1				0	0			1	1	1	1	0	0

Table. 1 Key generation using QKD is represented.

Key generation using QKD is represented in table1. From table 1, the key generated is: 100111100 and is shared between client and server. The generated key is used as one-time pad to provide unconditional security. The generated key will be used as shared secret for the next session.

IV. DETECTION OF EAVESDROPPING

In this work the user who wishes to store their data in external cloud storage uses quantum computation and encryption for processing their data. The cloud user uses QKD protocol for generating and distributing the secret key over secure channel. While generating the secret key there is a possibility for the eavesdropper to listen on the communication. The key generation is done in two different channels through quantum channel and public channel. After the key has been generated, a part of key generated has to be verified (sifted phase) through public channel. Client and server publicly agreed upon the random subset of bits from the sifted bits and compare the corresponding bits. During sifting phase, there is a possibility for the eavesdropper to listen to the communication and disturbs the communication. In the absence of noise, if the comparison shows an inconsistency, then there is a possibility for the presence of eavesdropper.

The probability to find the presence of eavesdropper (p_d) is calculated using equ.4

$$p_d = 1 - \left(\frac{3}{4}\right)^n \quad (4)$$

Where,

N-number of key bits for public comparison.

V. DETECTING MAN-IN-MIDDLE ATTACK

Quantum Key distribution protocol is vulnerable to man-in-middle attack [11] and [20] when used without authentication. This works uses SSLQKD protocol which provides secure transaction between SSL enabled client and SSL enabled key server. SSLQKD uses quantum SSL handshake protocol to instantiate the communication. At the end of the SSL handshake protocol, ChangeCipherSpec message and finished message have been exchanged. The SSLQKD ensures mutual authentication at the end of quantum SSL handshake protocol by calculating SSL finished message at both sides.

SSL finished can be calculated as follows.

$$pre\_mastersecret = K + S \quad (5)$$

1. At client side:

 DOI : 10.35291/2454-9150.2020.0182

$$mastersecret = PRF(pre\_mastersecret, "mastersecret", clienthello.random)$$ (6)

$$PRF(mastersecret, finished\_label, hash(handshakemessage))$$ (7)

2. At server side:

$$mastersecret = PRF(pre\_mastersecret, "mastersecret", serverhello.random)$$ (8)

$$PRF(mastersecret, finished\_label, hash(handshakemessage))$$ (9)

SSL finished message is calculated using pseudo Random number generator function. SSL finished message from client side has to be calculated by client using equ 7 and has to be verified at server side .SSL finished message from server side has to calculated by server using equ 9 and also has to verified at client side. In this SSLQKD the man-in-the-middle attack can be detected at the final stage of SSL handshake protocol. If any mismatches on the calculated SSL finished, then the attacker is detected.

VI. PERFORMANCE ANALYSIS

a. Measuring QBER:

The client and server wish to have secure communication use the SSLQKD protocol to generate and distribute the secret key. The SSLQKD protocol uses the BB84 protocol for quantum key distribution. During public announcement, if the eavesdropper is present the bit values of server and client will be differed. After a sifted key is transmitted through quantum channel, a small portion of the sifted key is verified through public channel to find the anti-correlation [19] between shared bits. This error is referred as quantum bit error rate (QBER). In presence of eavesdropper or hacker the Quantum bit error rate (QBER) increases. The QBER can be calculated [19] by dividing the number of errors by the size of the sample and multiplying by 100.QBER is used to identify how efficient the system is and to identify the presence of eavesdropper. The Quantum bit error rate (QBER) can be calculated using the equation (10) as

$$QBER = Pwrong / (\text{Pwrong} + \text{Pcorrect})$$ (10)

And the key generation probability is measured as in equation (11),

$$KGP = Pcorrect + Pwrong$$ (11)

Assuming that the photon number per pulse satisfies a poissonian distribution $P_{correct}$ and P_{wrong} can be calculated using the formula (25) and (26) respectively,

$$Pcorrect = 1 - \sum_{n=0}^{\infty} \frac{\mu^n e^{-u}}{n!} \left(1 - \frac{Yo}{2}\right) (1 - t\eta a)^n$$ (12)

$$Pcorrect = 1 - \left(1 - \frac{Yo}{2}\right) e^{-\mu t \eta a}$$ (13)

$$Pwrong = 1 - \sum_{n=0}^{\infty} \frac{\mu^n e^{-u}}{n!} \left(1 - \frac{Yo}{2}\right) (1 - t\eta(1-a))^n$$ (14)

$$Pwrong = 1 - \left(1 - \frac{Yo}{2}\right) e^{-\mu t \eta (1-a)}$$ (15)

This work shows the security of quantum key distribution [20] against the man-in-middle attack. If any hacker on the secure channels then QBER (Q bit error rate) increases and finally the hacker will be detected. The error correction is done through public channel; some of information of secret key may leak to hacker. The process of privacy amplification ensures that no information about final secret key has been leaked to hacker during error correction. The error correction and privacy amplification are to make sure the secret key is secure.

Performance analysis:

The Netscape navigator and internet explorer supports SSL. The SSLQKD protocol is implemented based on software simulation. The proposed protocol is secure against eavesdropping and man-in-middle attack by the law of quantum physics. The efficiency of the secure system can be verified by measuring the value of QBER. In Fig.3 all QKD sessions have been performed at a quantum bit error rate (QBER) of 5% and also 20% of sifted bits have been used for public comparison to estimate the amount of QBER. In Fig.4 describes the probability of detecting eavesdropping.

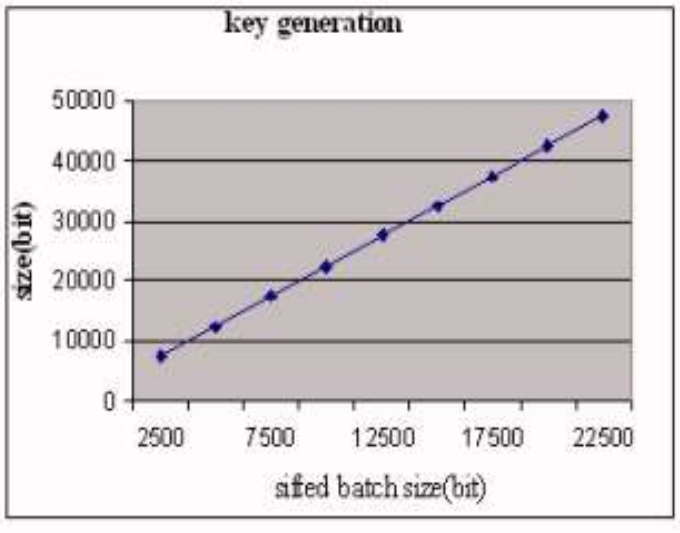

Fig .3. Quantum key generation with QBER = 5%

 DOI : 10.35291/2454-9150.2020.0182

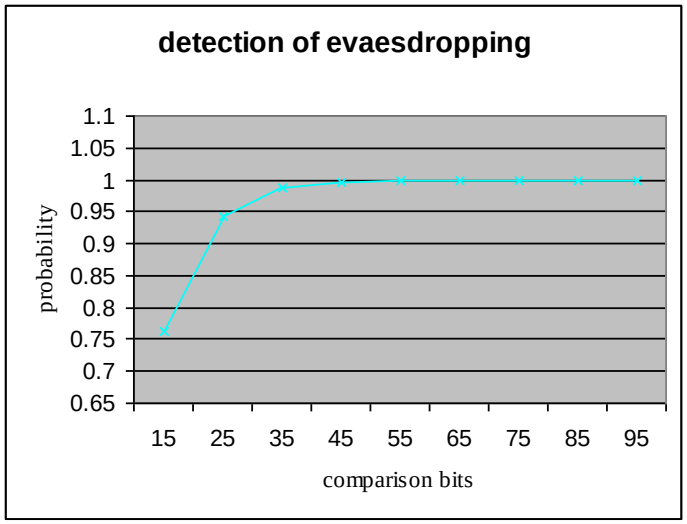

Fig .4. Probability of detecting eavesdropping

VII. CONCLUSION

This scheme deploys a secure transaction cloud environment using secure socket layer quantum key distribution protocol (SSLQKD).The paper explains the SSLQKD for distributing secret key between key server and client securely. BB84 QKD protocol was implemented to generate secret key. The generated secret key is used as one-time pad that provides unconditional security. For every upload of user's data, new secret key is generated to encrypt the data. The SSLQKD protocol has proved to be secure against man-in-middle attack and detects eavesdropping by mutual authentication and measuring QBER. In future, the new feature SSL termination may be used with cloud load balancer to achieve a significant performance increase while dealing with high volume of SSL traffic.

REFERENCES

[1] Sufyan T. Faraj, "*Integrating Quantum Cryptography into SSL "*.Special Issue of Ubiquitous Computing Security Systems *in* UbiCC Journal - Volume 5

[2] D. Binosi, Ed: "*Quantum Information Processing And Communication" ,* Strategic Report On Current Status, Visions and Goals for Research in Europe, ERA-Pilot Work Package 1 – QUROPE Work Package 2, Version 1.5, February 2008

[3] C.H. Bennett and G. Brassard. *"Quantum cryptography: Public-key distribution and coin tossing"*. In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, IEEE Press, pages 175–179, Bangalore, India, 1984.

[4] R. Hughes, Ed.: "A *quantum information science and technology roadmap; Part 2: Quantum cryptography*", Report of the quantum cryptography technology expert panel, *ARDA*, LA-UR-04-4085, Version 1.0, (2004).

[5] C. Elliott: "*Building the quantum network*", New Journal of Physics, Vol. 4, pp. 46.1- 46.12 (2002).

[6] C. Elliott et al:" *Current status of the DARPA quantum network*". BBN Technologies, *arXiv: quant-ph/0503058 (*2005).

[7] R. Alleaume, Ed.: "SECOQC *white paper on quantum key distribution and cryptography*",*Secoqc-WP-v5*, Version 5.1 (2007).

[8] M. Dianati and R. "*Alleaume: Architecture of the Secoqc quantum key distribution network*", GET-ENST, France, *arXiv: quantph/ 0610202v2* (2006).

[9] M. Sfaxi, S. Ghernaouti-Helie, and G. Ribordy:" Using quantum key distribution within IPSec to secure MAN communications", *Proceedings of the IFIPMAN 2005 Conference on Metropolitan Area Networks*, Vietnam (2005).

[10] Solange Ghernaout-Helie Mohamed Ali Sfaxi," Applying QKD to reach unconditional security in Communications".

[11] T. Nguyen, M. Sfaxi, and S. Ghernaouti- Helie: "802.11i encryption key distribution using quantum cryptography", *Journal of Networks*, Vol. 1, No. 5, pp. 9-20 (2006).

[12] A. Pasquinucci: "Authentication and routing in simple quantum key distribution networks", UCCI.IT, Italy, *arXiv: cs.NI/0506003v1* (2005).

[13] H. Bechman- Pasquinucci and A. Pasquinucci: "Quantum key distribution with trusted quantum relay", *arXiv: quantph/ 0505089v1* (2005).

[14] D. Collins, N. Gisin, and H. de Riedmatten: "Quantum relays for long distance quantum cryptography", *arXiv: quant-ph/0311101* (2003).

[15] R. Taylor: "Near *optimal* unconditionally secure authentication", *EUROCRYPT'94*, LNCS, Springer-Verlag, Vol. 950, pp.244- 253 (1995).

[16] DOMINIC MAYERS," Unconditional Security in Quantum Cryptography". Journal of the ACM, Vol. 48, No. 3, May 2001, pp. 351–406.

[17]. Xu Huang, "Implementation of Quantum Key Distribution in Wi-Fi (IEEE 802.11) Wireless Networks **".**ISBN 978-89-5519-136-3, Feb. 17-20, 2008 ICACT 2008.

[18] Mehrdad Dianati, "Architecture and protocols of the future European quantum key distribution network". Security and Communication Networks, *Security Comm. Networks.* 2008; **1**:57–74 Published online in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/sec.13

 DOI : 10.35291/2454-9150.2020.0182

[19] Nikolina Ilic, "The Ekert Protocol ".Journal of. Phy334,

NUMBER, pp.1-4, 2007.

[20] Fei, Y., Meng, X., Gao, M. *et al.* Quantum man-in-the-middle attack on the calibration process of quantum key distribution. *Sci Rep* **8,** 4283, 2018.

[21] Mohamed Elboukhari, Mostafa Azizi and Abdelmalek Azizi *"Improving TlS Security by Quantum Cryptography "*,International Journal of Network Security & Its Applications (IJNSA), Vol.2, No.3, July 2010.

[22] Ki-Woong Park, Kyu Ho Park "*ACCENT: Cognitive Cryptography Plugged Compression for SSL/TLS-Based Cloud Computing Services*". ACM Transactions on Internet Technology, Vol. 11, No. 2, 2011.

[23] Yoshiaki Shiraishi, Masami Mohri, Youji Fukuta "*A Server-Aided Computation Protocol Revisited for Confidentiality of Cloud Service*" *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, volume: 2, number: 2, pp. 83-94

[24] Matthias Heid, Norbert Lutkenhaus "*Security of coherent state quantum cryptography in the presence of Gaussian noise* ", (arXiv: quant-ph/0608015v4 19 Apr 2007).

[25] C. Williams et al: "A high speed quantum communication testbed", *NIST proceedings*, 2002.

[26] V. Fernandez et al: "Passive optical network approach to gigahertz-clocked multi-user quantum key distribution", *IEEE Journal of Quantum Electronics*, Vol. 43, No. 2, (2007) (pre-press version, *arXiv: quantph/ 0612130*).

[27] Pankaj Patidar et al. "Network Security through SSL in Cloud Computing Environment",(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (6), 2011, 2800-2803

[28]Https://blog.cloudsecurityalliance.org/2011/09/30/when-it-comes-to-cloud-security-don%E2%80%99t-forget-ssl/

[29] Https://www.globalsign.com/en-in/cloud/

[30]Http://comluv.com/ssl-encryption-a-protocol-that-authenticate-cloud-computing/

[31]Pandeeswari N & Ganeshkumar P 'Anomaly detection system in cloud environment using fuzzy clustering based ANN', Mobile Networks and Applications. Springer, pp. 1-12. Doi: 10.1007/s11036-015-0644-x. volume 21, Issue 3, June 2016

[32]N. Pandeeswari, R. Karuppathal ," Hypervisor Based Anomaly Detection System in Cloud Computing Using ANFIS", Journal of Internet Technology, Volume 18 , No.6, 2017.

[33] V. Swathi; M.,P. Vani , "Secure cloud computing using homomorphic construction", International Journal of Cloud Computing, Vol.8 No.4, pp.354 – 370, 2019.

[34]*Keisuke Inokuchi, Kenichi Kourai."*Secure VM management with strong user binding in semi-trusted clouds", Journal of Cloud Computing: Advances, Systems and Applications. Vol.9, issue. 1, pp.1-22, 2020.

 DOI : 10.35291/2454-9150.2020.0182