ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Secure and Efficient Cloud Data Storage Using Homomorphic Schema

Mr. Venkanna. B
Department of Computer Science and Engineering,
Vaageswari College of Engineering,
Karimnagar, Telangana - 505481, India.

Dr. Ravisankar Malladi
Department of Computer Science and Engineering,
Vignan's Lara Institute of Technology & Sciences,
Vadlamudi, Guntur-Dt-522213, A.P, India.

Abstract

Distributed computing is a fascinating innovation which gives a stage to utilize remote sources to store client's data, oversee and practice information, rather than neighborhood frameworks. There are many points of interest from distributed computing and furthermore a few impediments, similar to it isn't secure to keep the information on cloud since it is taken care of by cloud specialist co-op. In this stockpiling framework, information is moving starting with one end then onto the next end going through numerous hubs which may cause Data Manipulation. There is a shot of unlawful alterations for an encoded bundle. That will spread all finished system this is all a direct result of switch joins the got parcels and contaminated ones. This is otherwise called the contamination assault. To conquer this assault, this paper presents the Homomorphic Schemes for Encryption and Decryption utilizing Network Coding.

Introduction

Distributed computing [1] is a sort of range where clients can ready to store information and can be gotten to effectively. Cloud server will deal with the information that will be put away under the client account. This oversees client's information securely and beat the conventional downside of putting away the information in frameworks equipment. Cloud is being utilized to store information as well as a reasonable, effective, and adaptable to get to. By utilizing cloud we can do chip away at the information whenever at wherever. Cloud server gives an opportunity to store our information in server through web as opposed to putting away at our nearby server. As we typically store our information in the cloud server there might be a possibility of information misfortune which is hard to recoup. Along these lines, so as to take care of this issue there is a need of secure cloud which will deal with these sorts of issues.

In view of this issue one of the specialists presented the idea called Network coding so as to enhance organize limit, however it prompts contamination assaults. It is an intense assault which can succeed notwithstanding when the quantity of ordinary hubs is more than the lower bound. In this assault the assaulted hub is misdirect to an uncommon area, which prompts disarray of traded off hub with ordinary hub. It is additionally includes noxious bundles while going in the systems. To conquer this, we actualize the Homomorphic plot; this procedure gives information security while information is going through hubs [2].

Related work

Network coding: It is a routing model where a router in the network sends out encrypted data packets which are a function of received packets, instead of traditional store and forward approach. (One drawback of this approach is that the number of total "Sentinels" is finite. Thus, auditing can only be done a finite number of times.) If an encoded packet is modified illegally the modification can quickly spread to the whole network. Secure network coding can be explained through this paradigm. Sender sends the data in the form of packets by authenticating it with some keys. The router en-routes the packets in the form of linear combinations and the receiver decodes the packet data received by the router. By using the Homomorphic Mac technique secure network coding has been generated. It includes the five algorithms (setup, keygen, Auth, Combine, and Verify).

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

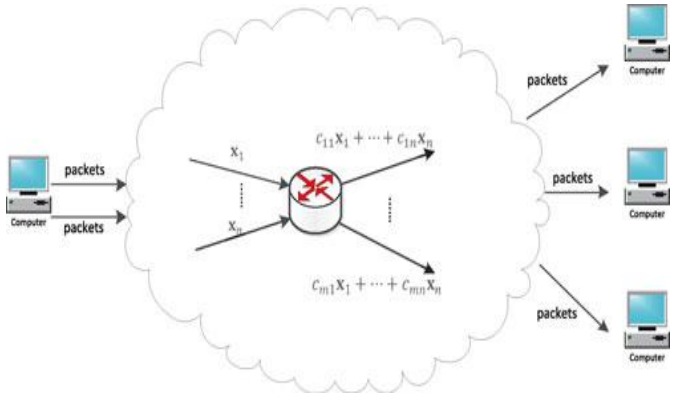

Fig:- A secure network coding system

The setup algorithm is carried by data owner to setup an account on a doubtful server. (On input a security level parameters and the number of cipher text classes). Keygen algorithm is carried by the owner of data to generate a random public key or master secret key. Authentication [3] can be done for input of first packet to be sent out in the network, the sender computes the authentication information as follows: First, compute the values and the authentication information is sent out. Combine can be done for received a group of packets and their authentication information. A router runs this algorithm to generate a combined packet. A router also computes the combined authentication information and the router sends out. Verify algorithm is run by the receiver on input packet and its authentication information, if packet is correct it represents 0 otherwise 1 as outputs. However, Network coding has been shown to be ideally use bandwidth in a network. A node can inject malicious packets that can quickly affect many receivers, but the scheme is very inherently accessible to pollution attacks by infected nodes in the network. A node is injecting a garbage can quickly affect many receivers. An attacker can easily pollute a packet even if it is encrypted by a forging signature or by a collision under hash function.

Pollution Problem

For example, if user u1 wants to send the data through the network, let's just assume malicious user m1 controls the router r1 (first router). Sender selected a file to send and that file is divided into set of packets [4]. Using Homomorphic scheme the linear combination of packets is travelled through the network. Because of router r1 is malicious, instead of forwarding the original packet, r1 forwards the malicious packet. Upon receiving the malicious packet, every router mixes that malicious data with the packets that the router have. So, very quickly using one malicious packet can pollute the whole network and cannot recover the original file. This is called a pollution problem. Even if the malicious user does inject a malicious packet in to the network, the very next node that receives the malicious packet can be able to detect the malicious packet and drop that packet. So, whenever the router in the network receives a combination of packets it needs to check whether the packets are valid or not. So in order to address this problem we used a Homomorphic Scheme.

Proposed Approach

In network coding, the job of intermediate nodes is to actively mix input packets in order to produce output packets. While mixing, the attacker nodes may inject malicious packets into the network, which leads to "pollution attack". The polluted packets spread quickly into the network and they totally destroy all the packets which are not malicious. An attacker may produce collision under hash function or forging the signature to easily pollute a packet even if the packet is encrypted. In the proposed approach, we address pollution attacks which are occurred in network coding systems. In order to overcome this problem, we proposed a scheme called "Homomorphic Schemes for Encryption and Decryption" [5]. The Homomorphic property of the signatures allows nodes to sign any linear combination of the incoming packets without contacting the signing authority. In this scheme, it is difficult to find which linear combination was used in the generation of packet. That's why it is computationally tough to crack the packets. Using this technique, we can perform computations on encrypted information without decrypting it. Because, it could make network coding more secure.

During this, we need to keep our cloud files cryptographically scrambled using a secret key. This

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

helps to decrypt the files after encryption. The main advantage of scrambling is, no attacker can able to crack the data. The user performs computations on the encrypted data so that he can decrypt it easily. User cannot decrypt the data in network. Instead he must download the data and perform computations locally. By this, the attacker cannot be able to decrypt the packets in network and add malicious ones. With this technique, cloud can perform computations locally rather than the user and returns the encrypted result. It allows arbitrary computations on encrypted data. The process of computing on encrypted data means if a user has a function „f" and want to obtain f(p1,p2..pn) for some inputs (p1,p2….pn).

We need to do computations on f(p1,p2…pn) and obtain (r1r2,…r n) results for the input. This process allows producing a secret key using a master key which is dependent on a function „f". Given a cipher text, the secret key allows the user to know the value of „f" that is applied to the plain text. This technique grants control over the functions that can be applied on the data via a master key holder. The encrypted text consists of cipher text along with the authentication. The authenticated user can decrypt the result into plain text using the secret key. Network coding consists of the following algorithms [6]:

A. Keygen (1λ) → (PK, SK) With the help of a security parameter λ, this algorithm generates a public key and a secret key.

B. Encryption (PK, M) → E (M) The encryption algorithm encrypts the message „M" with the help of the public key.

C. Authentication (Ci , SK) → CT By taking ith packet as input a string of cipher texts and the secret key to produce the authentication to the user along with the cipher text (CT).

D. Combine (Pi , CT) → CP It considers group of packets (Pi) and their authentication information (CT) in order to generate the combined information and sends to the authenticated user.

E. Decryption (CT, SK) →This algorithm takes as input the cipher text and the secret key. The user who is authenticated can be able to decrypt the text.

Defence against pollution attack

In network coding, there is a problem called pollution attack in order to address this problem there are standard methods or signatures. But, in the setting of network coding those methods or signatures are not directly applicable. So, we developed a Homomorphic mechanism to prevent pollution attack in network coding. Homomorphic scheme comprises of four phases (Key generation, encryption, authentication, combine and decryption) [7].

Every file has a file identifier and whenever a user sends the file, file identifiers helps to bind all the related packets into a given file. In key generation phase, based on the user credentials and file identifiers, three keys are generated (keys for encryption, decryption and authentication). After that in encryption phase, by considering the encryption key and plain text we generate a cipher text along with the authentication which is ready to send through the network. As this Homomorphic scheme explains the calculations that can be done for encrypted text without decrypting it, the sender computes some calculations before sending it to the network.

User sends the encrypted data to server (enc (a), enc (b) where a and b are packets) and sends a request to the server asking for calculation of f(a,b). Server stores the encrypted data and it calculates the result of request sent by the sender (f(a,b)) based on the ciphers of a and b. Server also computes (f(enc(a), enc(b))) without decrypting a and b. Upon receiving the request of f(a,b) server sends (f(enc(a), enc(b))) to the sender. Now, user decrypts the result of (f(enc(a), enc(b))) using private key. Based on the above procedure, for every file that user sends, the server can make sure that the calculation is for the encryption data and not for the actual data. It helps to ease the decryption process for a genuine user and complicated for the malicious user. So, that we can prevent pollution attacks in the network [8].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Results

Audit Check	Processing Time (in sec)
Existing	1.16
Proposed	1.053
Enhanced	0.85

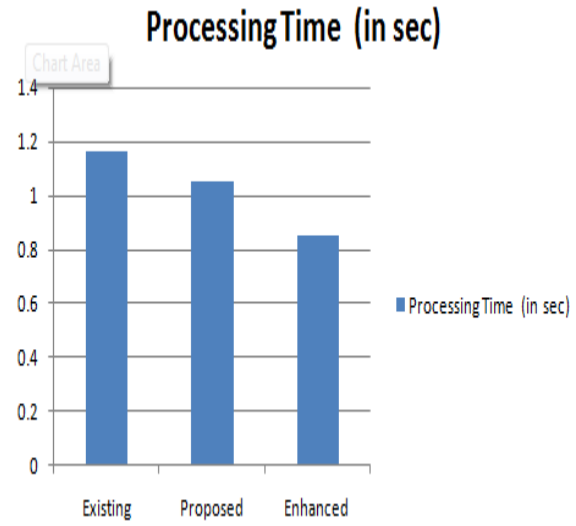

Fig:- Audit checking processing

Conclusion

With the knowledge of relationship between the secure cloud storage and secure network coding, this Homomorphic scheme helps in increasing the security for the files of user. We used Homomorphic mechanism to make the security stronger. It provides security in addition to detecting pollution attacks. Using Homomorphic Scheme only the authenticated user can decrypt the data. By using this technique we get the feasible time for both encryption an also the decryption process which helps the users to upload as well download the files in a stipulated time

References

[1] Fey Chen, Tao Xiang, Yuanyuan yang, and Sherman S.M. Chow "Secure cloud storage meets with secure network coding" in Proc. IEEE Transactions on computers, Vol.65, No 6. 2016

[2] Q. Li, J. C. Luis, and D.-M. Chiu, "On the security and efficiency of content distribution via network coding," IEEE Trans. Dependable Secure Comput., vol. 9, no. 2, pp. 211–221, Mar./Apr. 2012.

[3] S. Agrawal and D. Bone, "Homomorphic maces: Mac-based integrity for network coding," in Proc. Int. Conf. Appl. Cryptography Newt. Security, 2009, pp. 292–305.

[4] F. Zhao, T. Talker, M. M_edard, and K. J. Han, "Signatures for content distribution with network coding," in Proc. IEEE Int. Symp. Inf. Theory, 2007, pp. 556–560.

[5] D. Charles, K. Jain, and K. Lauter,"Signatures for network coding," Int. J. Inf. Coding Theory, vol. 1, no. 1, pp. 3–14, 2009.

[6] Nuttapong Attrapadung, and Benut Liberty, "Homomorphic network coding signatures in the standard model"

[7] S.-Y. Li, R. W. Young, and N. Cain, "Linear network coding," IEEE Trans. Inf. Theory, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[8] N. Cain and R. W. Young, "Secure network coding," in Proc. IEEEInt. Sump. Inf. Theory, 2002, p. 323.