Secure Cloud Storage Techniques: A Review

Ms.Srilakshmi U Sirurmath[1] , Dr. Deepashree Devaraj[2]

[1] Student, Dept of Electronics and Instrumentation, R.V. College of Engineering, Bangalore, India

[2]Assisstant Professor, Dept of Electronics and Instrumentation, R.V. College of Engineering, Bangalore, India

[1]srilakshmius.ei17@rvce.edu.in, [2]deepashree@rvce.edu.in

Abstract: *Cloud technology has exponentially seen a rise in its absorption for various applications. Cloud users with limited storage might transfer their information to remote systems. In return for monetary compensation, these servers provide access to their clients' data. Cloud storage protocols verify the integrity of this data which is hosted on the cloud. Broadly there are two types of data – static and dynamic. While many efficient protocols are already present for static data, much research is being undertaken to build a secure cloud storage system for dynamic data. This paper analyzes these existing and proposed cloud storage protocols for both static and dynamic data. Important performance parameters are identified and a comparison is drawn between the chosen methods in order to draw a contrast between the efficiency of the techniques chosen.*

Keywords: **Cloud Storage, Protocol, Static data, Dynamic data, Comparison**

1. INTRODUCTION

Due to the prevalence of cloud computing, cloud storage is becoming increasingly popular. Most consumers and businesses are shifting to the cloud since it is considerably cheaper and more convenient. Finding sufficient storage capacity to keep all of the data acquired by some computer users is a serious difficulty. As a result, consumers are more likely to purchase enormous amounts of data or bigger hard drives, despite the fact that they are still running out of storage space. People are finding it so much simpler to acquire a huge amount of space mostly on cloud, which refers to storing information to an off-site storage system maintained by a third party, thanks to this new cloud computing technology. However, investigations [1] suggest that loss of data in cloud providers is a possibility. As a result of all of this, cloud users begin to worry about the security of their data kept on all these 3rd party servers, fearing a data breach.

Secure cloud storage protocols are two party protocols between user and server, they offer a way to tell if server stores client's information securely. These protocols are categorized as secure cloud storage protocols for static data (SSCS) and dynamic data storage protocol (DSCS) depending on the characteristics of data that is outsourced. In these protocols for

protecting cloud storage, user can inspect outsourced information without having to read the entire file and nevertheless detect undesired data modifications made by a rogue server. During an inspection, clients challenge the servers and server responds with evidence of storage (based on data stored).

Verifying the authenticity of data is referred to as secure network coding [2]. Each intermediary node in a path (excluding few nodes) in a network protocol mixes receiving packets to generate additional packet. These protocols have superior productivity and economy than store-and-forward routing, but are vulnerable to contamination attacks from hostile nodes that send incorrect packets. These packets generate more downstream which the receiver may not be able to decode. To counter these attacks, secure network coding (SNC) protocols use cryptographic techniques: the sender verifies every packet by appending a tiny tag to it. Homomorphic message authentication codes (MACs) or homomorps are used to create these authentication tags. The topic of building a safe cloud storage strategy for dynamic data (DSCS) was analyzed and a SNC protocol was used to create effective DSCS protocol. A link between safe cloud storage as well as reliable network coding was discovered previously.

They show, in particular, that several of the methods used in an SNC protocol may be used to establish a safe cloud storage protocol for static data. However, because its design does not manage dynamic data, it is inadequate in several applications for which a client has to quickly update (input, remove, or edit) data.

Distributed storage systems that propagate user data across numerous servers have also been built using network coding approaches. However, if few of servers malfunction, they primarily attempt to lower the restoration bandwidth. In another scenario, an investigation was carried out to check if the techniques used in SNC protocol could be utilized to build a safe and reliable cloud storage system for dynamic data (for a single server). Though dynamic data is general and supports arbitrary update (addition, removal, and alteration) actions, append-only data (new data relating to a data file is appended to an existing data file) find their application as well.

These applications generally keep archived and present data up to date by adding new data to existing databases. Append-only information can also be used to keep track of various log patterns. The data owner in most of these applications need a server to save bulk data in an untainted and readily accessible manner, with attachment being the only permitted modification.

2. METHODOLOGY

2.1 Secure Network Coding (SNC) Protocol

In this protocol, there are essentially three important modules, namely: router, receiver and sender. A sender intends to distribute data to a set of recipients or in this model, receivers. The sender splits the information into packets while sending these packets in a combination which is linear through a network. A network router also delivers a linear mix of the packets of data it receives to its following hops. On the other hand, receiver receives enough data

packets that are encoded, it utilizes linear equations in order to decoded those packets of data in order to transform it back to original data. Sender appends authentication information for each packet of data in order to prevent a hostile router from distorting it. A router gets a sequence of data packets and it verifies them to ensure that they are correct. It then proceeds to integrate the right packets received, and then transmits the merged packet along with the consolidated authentication data. The aggregated authentication metadata is calculated based on the protocol's particular features.

The essential concept that enables the generic architecture is quite simple. The user is thought of as a sender that desires to deliver data to a few recipients and on the other end the user is also a receiver within the network and the cloud as a network router. When data is outsourced by user, it fragments it into packets, each of which may be assumed to be a vector in a set of finite fields. After that, the packets of data is authenticated by the user by means of appending authentication data to it. The cloud is used to store the authenticated data. The cloud is essentially regarded as a router which takes packets of data from the network and produces a packet which is encoded linearly when an audit inquiry is submitted by the user. The user sends audit query that includes the data packet's indices and its encoding coefficients.

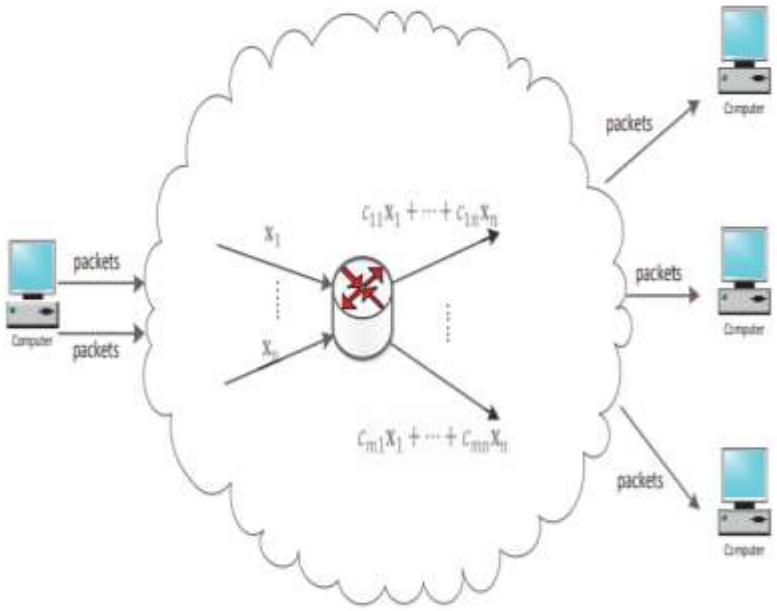

Fig.1 A SNC System

The user is then provided with a proof of the packet which is encoded and its associated authentication data. The user takes on the role of data receiver in this

scenario. User determines if the cloud preserves the previously outsourced data unharmed by evaluating if the data packet which is returned is legitimate or not. [2]

2.2 Dynamic Secure Cloud Storage – I (DSCS I) Protocol

A DSCS I protocol behaves as a PDP or POR protocol in accordance the data retrievability. The protocol uses rank based list of authenticated skips in order to guarantee that the dynamic data is fresh. The hash function which is used in this rank based list of authenticated skips is resistant to collisions. The assumption made is that file that requires outsourcing is a group of vectors each with a particular fixed and uniform dimension. Authentication tag is allocated to every block or vector such that the block of data are basically the units into which the whole file is split into. A segment here is defined as some number of components of the vectors. There are four algorithms in this protocol, namely KeyGen, Verify, Prove and Outsource and these algorithms in turn call the SNC.KeyGen, SNC.Verify, SNC.Combine and SNC.TagGen from the parent SNC protocol respectively. [4]

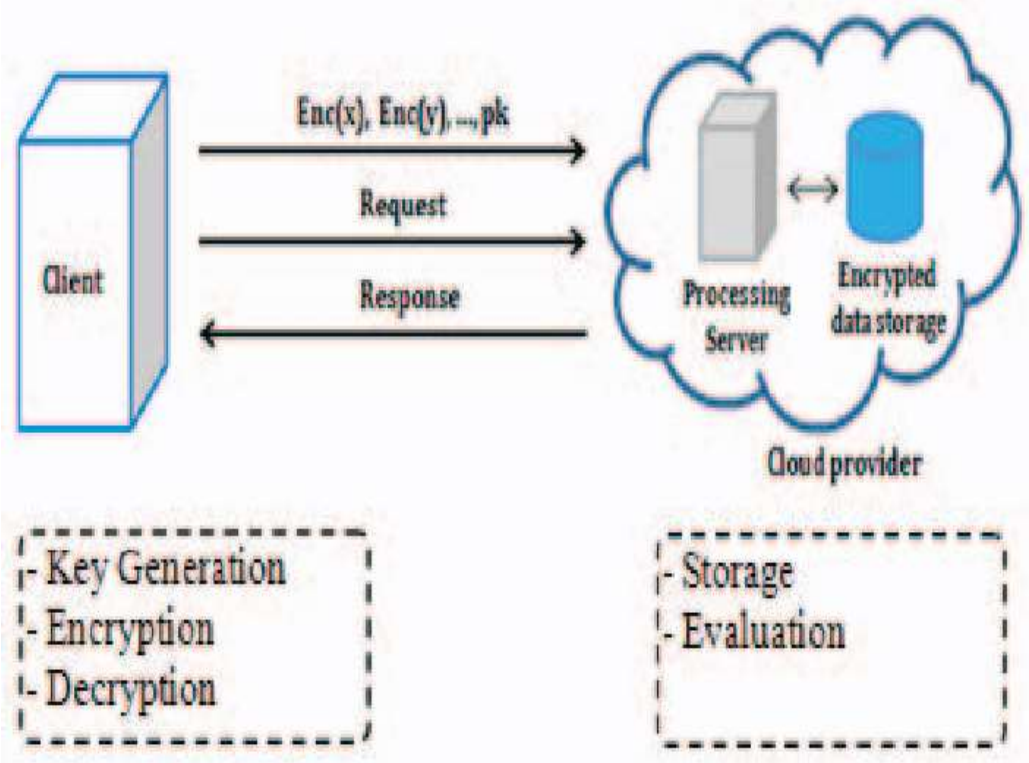

Fig. 2 A DSCS-I System

The integrity of the file is assured in this protocol only if the SNC protocol which the DSCS protocol is built upon is first, secure and the rank based list of authenticated skip list and the hash function which builds it is resistant to collisions. The public key size in this protocol O(m + n) instead of being constant. The tag in this protocol larger than a tag in DPDP I protocol by n + 1 bits. In DSCS I, the value public key has to be modified for every removal and addition of new data.

For the protocol to be secure, it needs to have the following characteristics:

Authenticity: The server produces proof of verification of the data only if the tags and vectors are stored without being tampered.

Freshness: The data on the server must be up to date with the actual file. [5]

2.3 Dynamic Secure Cloud Storage – II (DSCS II) Protocol

SNC protocols which can be utilized to create DSCS protocols for append type of data were investigated. The DSCS-II protocol was especially proposed for append type of data derived from SNC protocol which was implemented [3]. The reason why SNC protocol cannot be used is because for data which is dynamic, block indices are stored in the form of tags. Moreover, for append type of data, the block of data is inserted in the end and hence it does change the index of a block which exists before.

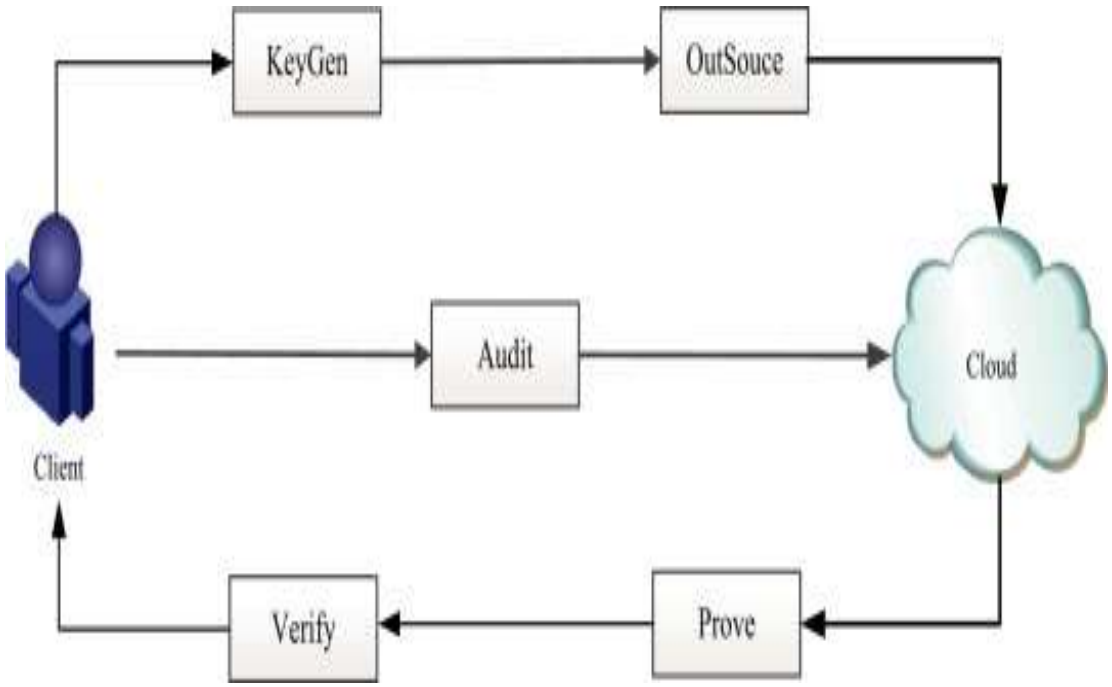

Fig. 3 A DSCS-II System

The DSCS-II protocol does not depend upon the data freshness or the structure of data. This happens since the block of data is not modified for any type of append action and hence the server does not have the responsibility to retain the older block of data. Hence, this protocol does not include a Verify Update function which was used in DSCS-I protocol. DSCS-II is verifiable publicly in the sense that anybody with access to public key may audit it. In random oracle model [4], DSCS II is safe. The safety procedure and evidence in DSCS II are identical to those in DSCS I, with the exception of append being the only permitted update and data freshness not being necessary. The integrity of the underpinning SNC protocol [3], which is safe in random oracle paradigm, ensures authenticity. The following algorithms are

included in this protocol: KeyGen, Outsource, AuthRead, VerifyRead, InitUpdate, PerformUpdate, Challenge, Prove, Verify.

To create the valuation of authentication tag to every vector, the client must do a multi exponentiation while running the outsource of the algorithm. For calculation of this number, the server uses one multi exponentiation. In DSCS-II, the size of the public key is O(m) which is considerably less than the size of public key in DSCS-I. Here only few blocks of data need to be changed during append actions. [5]

3. PARAMETER OF MEASUREMENT

3.1 Storage Overhead

Storage overhead is often referred to as the additional computational time, bandwidth, memory or other parameter required for a certain task to be performed. In this system, storage overhead is to be measured for both client side and server side in order to ensure consistency. The client simply has to save the private key and metadata which requires ongoing storage cost. The server, in contrast must store both the data file as well as the file that contains the user details for authentication.

3.2 Cost of Communication

The cost of communication when in an inspection is calculated by the size of disputed blocks, which is generally constant. Server responds by sending proofs to user. The size of an aggregated block as well as size of proofs for queried blocks determine the size of proof. For an update of data, the cost of communication cost is dependent upon the type of update. In order to insert data, the user must send a public specification along with the new data block to be added. In order to delete some data, the user must send the block index. For every update that is made, the server sends verification to the user.

3.3 Cost of Computation

The cost of computation is essentially the measure of time taken in order to run or execute the steps that are to be evaluated in the system. Here the computation cost is evaluated for:

1. Time taken to outsource data is often expensive when done for the first time but the consecutive trials are far less costly
2. Time taken to challenge data
3. Time taken by server to prove data after each verification
4. Time taken to verify data
5. Time taken to append new data

4. RESULTS

This section will briefly use a comparison table in order to draw a comparison between the discussed methods in terms of the parameters, namely – Storage Overhead, Cost of Communication and Cost of Computation.

Table 1. Comparison Table

Sl No.	METHOD	PARAMETERS		
		Storage Over head	**Cost of Communication**	**Cost of Computation**
1.	Secure Network Coding (SNC) Protocol	The storage cost is due to the secret key, data and the authentication data. The extra storage over head is due to authentication data.	Communication cost is dependent upon the size of constant audit query, linear amalgam of blocks of data that is queried and information for authentication.	The computation cost is made of, the time taken for outsourcing of data, auditing of data, proving and verifying the data.
2.	Dynamic Secure Cloud Storage – I (DSCS I) Protocol	The storage cost is due to the storing of secret key and meta data on client side and storing of authentication data along with the files on the server side.	The communication cost is dependent upon, size of a constant block which is aggregated and the tag that it includes as well as the size of skip list for data that is queried.	The computation cost is due to outsourcing data to clients, generation of challenge, generation of proof and verification of proof along with the updates made
3.	Dynamic Secure Cloud Storage – II (DSCS II) Protocol	The extra storage cost in this protocol is only due to the authentication tag	The communication cost is due to size of the tag which is aggregated and the size of block which is aggregated is not included.	The computation cost is due to the first outsourcing which is undertaken and due to audit and append actions.

5. CONCLUSION

While SNC protocol is reliable for static data, for dynamic type of data the DSCS protocols are necessary. All the DSCS protocols are derived from the existing SNC protocol. The DSCS protocols are both highly reliable and can be verified publicly as per the model. While DSCS protocols do bridge the short comings of SNC data, it is extremely important to comprehend that the quality of the DSCS protocol depends upon the rigidity of the underlying SNC protocol, hence if a SNC protocol is more efficient, that will automatically render the derived DSCS more effective and sturdier. The DSCS-II protocol bridles the shortcomings of the DSCS-I protocol, in terms of the public key size and the data block changes necessary during

append action. Moreover, in terms of the measuring parameters, DSCS-II is more efficient than DSCS-I and SNC protocol.

6. REFERENCES

[1] Y. News, “Cloud computing users are losing data, symantec finds,” http://finance.yahoo.com/news/ cloud-computing-users-losing-data-205500612.html, 2013.

[2] F. Chen, T. Xiang, Y. Yang, and S. S. M. Chow, “Secure cloud storage meets with secure network coding,” in IEEE International Conference on Computer Communications, 2014, pp. 673–681.

[3] D. Boneh, D. M. Freeman, J. Katz, and B. Waters, “Signing a linear subspace: Signature schemes for network coding,” in International Conference on Practice and Theory in Public Key Cryptography, 2009, pp. 68–87.

[4] M. Bellare and P. Rogaway, “Random oracles are practical: A paradigm for designing efficient protocols,” in ACM Conference on Computer and Communications Security, 1993, pp. 62–73.

[5] Binanda Sengupta, Akanksha Dixit, Sushmita Ruj. Secure Cloud Storage with Data Dynamics Using Secure Network Coding Techniques in IEEE Transactions in Cloud Computing.

[6] A. Juels and B. S. Kaliski, “PORs: Proofs of retrievability for large files,” in ACM Conference on Computer and Communications Security, 2007, pp. 584–597.

[7] H. Shacham and B. Waters, “Compact proofs of retrievability,” Journal of Cryptology, vol. 26, no. 3, pp. 442–483, 2013.

[8] C. C. Erway, A. K¨upc¸ ¨ u, C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” ACM Transactions on Information and System Security, vol. 17, no. 4, pp. 15:1–15:29, 2015.

[9] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, “Enabling public auditability and data dynamics for storage security in cloud computing,” IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, 2011.

[10] D. Cash, A. K¨upc¸ ¨ u, and D. Wichs, “Dynamic proofs of retrievability via oblivious RAM,” in EUROCRYPT, 2013, pp.279–295.

[11] E. Shi, E. Stefanov, and C. Papamanthou, “Practical dynamic proofs of retrievability,” in ACM Conference on Computer and Communications Security, 2013, pp. 325–336.

[12] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, “Network information flow,” IEEE Transactions on Information Theory, vol. 46, no. 4, pp. 1204–1216, 2000.

[13] S. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” IEEE Transactions on Information Theory, vol. 49, no. 2, pp. 371–381, 2003.

[14] S. Agrawal and D. Boneh, “Homomorphic MACs: MAC-based integrity for network coding,” in International Conference on Applied Cryptography and Network Security, 2009, pp. 292–305.

[15] D. X. Charles, K. Jain, and K. E. Lauter, “Signatures for network coding,” International Journal of Information and Coding Theory, vol. 1, no. 1, pp. 3–14, 2009.