Review of Implicit Security Mechanisms for Cloud Computing

Makhan Singh
University Institute of Engineering & Technology, Panjab University, Chandigarh, India

Sarbjeet Singh
University Institute of Engineering & Technology, Panjab University, Chandigarh, India

ABSTRACT
Cloud systems refer to the collection of interconnected servers that are provisioned dynamically on demand, for execution of applications, to the customer like electricity grid. Cloud computing has been drawing the interest from industry but there are still many issues that are in their primitive stage which are hampering the growth of cloud. One of these issues is security of data stored in the servers of datacenters of cloud service providers. Many schemes have been developed till date for ensuring security of data in distributed systems. In this paper implicit security using information dispersal and secret sharing algorithms have been reviewed which provides data security, reliability and availability of information.

Keywords
Cloud Computing, Data Security, Information Dispersal, Secret Sharing, Explicit Security, Implicit Security

1. CLOUD COMPUTING
Cloud Computing is the latest distributed computing standard which is drawing the interests of many researchers both in the academia and the industry. The term cloud denotes the infrastructure having large pool of dynamically reconfigurable virtualized resources, which are easy to use and accessible from anywhere in the world on demand over internet and uses the pay-per-use model. Cloud computing, a young and potential standard, is providing IT services as computing utilities. But several issues need to be looked after to make cloud computing acceptable to everyone. Security of data stored at different servers is important among them.

There are several definitions of cloud available in the literature. Ian Foster et al. in [1] have defined cloud as "a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [1]". Michael Armbrust et al. in [2] have defined cloud computing as "a term that refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services". Rajkumar Buyya et. Al. in [3] have defined cloud as "a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers". The NIST (National Institute of Standards and Technology) [4] has described the term cloud in an even way and this definition highlights several important characteristics of cloud available and hence has been followed.

The NIST Definition of Cloud Computing: "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 [4]."

Cloud Computing is perceived as a new technology, but it is not. It is actually a new working model that combines together a set of existing technology to run a business in a different way. In fact, most technologies like virtualization and utility-based pricing that is used by cloud computing are old concepts. Cloud computing influences these existing technologies to satisfy the technical and economic necessities of today's information technology.

2. CLOUD STORAGE CONCEPT
Data storage is one of the fundamental and basic implementation of cloud Computing. Rather than storing data on dedicated servers, as it is used in existing networked data storage, in cloud data storage, it is stored on many third-party servers. When saving data, the user sees a virtual server that is the data is getting stored at a particular location with a specific address. But in actual, no such place exists. It's just an alias indicating to certain virtual space cut out of the cloud. But actually, the user's data is stored on any one or more of the computers that are used to build the cloud environment. The actual storage location in cloud may change from time to time as cloud manages available storage space dynamically [5]. The user sees a "static" memory location for his saved data and can manage the storage space as if it is connected to his own computer, even though the location is virtual in nature. Cloud storage is both economical and secure in nature. In terms of cost, the virtual resources in the cloud environment are more economical than any other dedicated physical resources connected to a computer system or network. As per the security requirements the data stored in the cloud environment is more secure from accidental erasure or hardware crashes, as data is replicated across multiple dedicated machines. Since many copies of the data exist in the cloud, even if one or more computer goes offline, the cloud system continues to work as normal [5]. If one machine goes offline due to hardware crashes, the data is available on other machines in the cloud. Hence, these are copy based systems which provides reliability by mirroring data on 'n' individual storage servers and hence 'n-1' storage servers can crash without any data loss.

Cloud storage allows the users to save data on the cloud without worrying how it is stored. Reliability and security are the two main issues associated with cloud storage. Customers generally do not trust sharing their data on third party servers

unless assured that no one else would be able to access the data and only they can access it whenever required [6].

Now, many cloud storage providers use encryption techniques for securing customer data and the key management are usually done by themselves only. This is the most commodious way to access customer's data from any place. It also allows them to share their data with others. Since the users have no control on the file encryption process hence they also have no control over who may access to their data. Seeing this type of situation, the customers have to put a high level of trust on cloud storage providers. Due to the unsafe encryption techniques and data leakage at storage provider's side have confirmed that the issue of secure off-site data storage is still an important and not completely solved issue and hence it draws the attention of academia and industry [7].

The use of encryption algorithms and firewalls for securing customer data are the explicit security techniques which are not efficient during natural and man-made disasters. Using implicit security is another approach to provide data security, dependability and availability of information. Here data to be stored is divided into numerous different pieces and these pieces are stored on individual servers in the cloud. Users recreate the data by accessing individual servers through password; downloading the different partitions that are then combined to reconstruct the data. The data partition and reconstruction algorithm does not involve any use of encryption or decryption algorithms which eliminates the task of key management. The reconstruction of data using data partitions does not reveal any information about data until and unless some predetermined number of pieces are accessed. These predetermined numbers of pieces are known as threshold number. Thus, in order to access the data at least a threshold number of individual servers are to be compromised. Consider data D which can be divided into n pieces such that m pieces are required for reconstruction of data. The attacker does not get any information about data if the number of pieces compromised is less than m. The information dispersal algorithms [8], [9], [10], [11] and secret sharing schemes [12] are the example of implicit security and they does not uses encryption algorithm. Usually, most implicit security systems use a (l, m, n) data distribution algorithm, where l, m, and n are integers and $l \leq m \leq n$. The division of data is such that if less than l pieces are compromised it does not reveal any information about the actual data, if l or more than l but less than m pieces are lost, it reveals some information about the data and m or more pieces reveals complete data. So, if $n > m$ in that case $(n - m)$ pieces are redundant. On one side, a (1, 1, n) data distribution algorithm creates n duplicates of the original data, and on the other side, a (1, n, n) algorithm splits the data into divisions of 1/nth the size of the actual data, without redundancy. Likewise, a (n, n, n) algorithm division creates n pieces, where each piece is of the same size as the original data and all the pieces are required for reconstructing the data, therefore there are no redundant pieces. (m, m, n) data partitioning schemes are secret distribution schemes, here in order to reconstruct the original data m is the required threshold number of pieces of data and if the number of pieces is less then m it do not reveal any information about data.

Information dispersal algorithms have (l, m, n) data partitioning scheme where $l < m$. It creates partitions smaller in size than the original data and hence reveals some information once more than l shares have been accessed. But they may provide adequate security [13], [14] and can be used in survivability of data storage systems and protection of data loss due to any disasters and intrusions.

3. IMPLICIT SECURITY ALGORITHMS

An information dispersal algorithm distribute the information being stored, F, into n pieces among n active servers, in such a way that the retrieval of F is possible even in the presence of up to t failed (inactive) servers [11]. An information dispersal algorithm enhances the confidentiality and availability of data without requiring any additional storage space. In 1979 Adi Shamir [12] proposed a scheme to share a secret among several parties and in order to re-construct the contents of secret some threshold number of parties hove to co-operate each others. The information is fragmented in n parts and stored across different servers. To re-construct the data, a previously defined threshold number m (m<=n) of data pieces are to needed need. No information will be re-constructed if less than m fragments are available. Michael O. Rabin [11] used this idea in 1989 and presented an initial scheme for efficient and secure distributed storage of data called "Information dispersal algorithm". In this paper five implicit security algorithms are reviewed.

3.1 Shamir's Algorithm

According to this scheme [12] data is fragmented into 'n' pieces such that 't' fragments are required to make data and less than 't' fragments does not construct any information about data. This scheme is based on polynomial interpolation. This scheme finds polynomial of degree't-1' for given't' data points. The polynomial f(x) is given as

$$f(x) = \sum_{i=0}^{t-1} a_i x^i$$

here ai is set to the secret and coefficients a_1 to a_{t-1} given random values. The value f(i) is assigned to user 'i'. Here if 't'out of 'n' users join together then they will generate the polynomial using Lagrange interpolation. Knowledge of just 't-1' values does not construct any data that is hidden.

Advantage of this scheme is:

a) Since it requires at least 't' shares of data and not less than this number gives any information about the data hence the security of data increases.

Shortcomings of this scheme are as follows:

a) Size of each piece is approximately equal to the size of data. Hence this method is space inefficient.
b) If data is modified during its stay in cloud servers, then this method will not indicate which piece is modified and user also not guaranteed that data is modified or not during its stay..

3.2 Rabin's efficient dispersal of information for security, load balancing, and fault tolerance.

In this scheme [11], the way of dividing secret into pieces is different from Shamir's algorithm. Robin makes 'n' fragments from the file 'F' by assuming $F = b_1, b_2 \ldots\ldots\ldots\ldots\ldots b_n$.
In this file b_i represents some character. There are N numbers of characters in file. Each bi character denotes a single byte as $0 \leq b_i \leq 255$.However the character b_i can represent any number of possible values at the implementer's discretion.
Let us now select a prime number P such as 255<P. Now, we can take F as a string of residues modulo P. All calculations preformed subsequently on modulo P, and each such value

may exceed 255. Taking example if P=257, the calculation may calculate values as high as 256.
In next step select x and y such as $x \leq y$ so that required space usage or fault tolerance can be achieved. File F divided into x sections of length. $F = (b_1, b_2 \ldots\ldots\ldots b_x)$ $(b_{x+1} \ldots\ldots\ldots b_{2x})$.
Here it is important to note that if n is not multiple of x, zeros must be added with b_i. And therefore we can write the sequence of F as $F = S_1, S_2$ ---------- $S_{(n/X)}$
Now in order to create fragment F_i we consider alpha vector $a_i = (a_{i,1}, a_{i,2}, \text{-------- } a_{i.m})$ which is associated with fragment. Hence $1 \leq i \leq n$ as there are n fragments .With the help of alpha vector and the file which was divided into sequence of bytes,the fragment F_i is considered where $1 \leq i \leq n$
$F_i = c_{i,1}, c_{i,2}, \ldots\ldots\ldots\ldots\ldots c_{i,(n/x)}$ where $c_{i,j} = a_i . S_j$. Let us assume that B is the unknown data which is to reconstruct and C is the data from fragments where C_i is the data from i^{th} fragment. This can be written as A . B = C
Hence $B = A^{-1}. C$
As per dispersal algorithm B is having data from original file.

Advantages of this scheme are as follows:

a) Size of each piece of the secret is small which makes it space efficient.
b) If any piece of data is modified during its stay on servers, fingerprinting will help in determining which piece is modified.

Shortcomings of this scheme are as follows:

a) There is need for management and storage of secret keys.
b) Verification of the fingerprint requires knowledge of the secret key, but then whoever can read the information can also modify it without being detected.

3.3 Distributed fingerprints and secure information dispersal

This scheme [9] is based on distributed fingerprints. It uses Rabin's Information dispersal algorithm for fragmentation of data into pieces. In this scheme distributed fingerprints are used to find the fingerprints of information in distributed environment. These are known as public fingerprints which ensure data integrity. These fingerprints of data are stored without encryption keys. Using public fingerprints everyone in system can calculate fingerprinted information and if same is altered it is being noticed. To illustrate this technique let us assume there is a secure storage in which information is stored for later use and same is not modified here. This safe space is used for integrity validation by fingerprinting the whole information by the help of short string and store the fingerprint result in this space. The information here can be modified and without being noticed by replacing information by different piece of data whose fingerprint is the same. This type of fingerprints functions are hard to find and are called as collision free or one-way hash functions. In particular, no encryption keys are involved for handling them, and the same function can be used to fingerprint information of different sizes. Hence, the problem of creating the secure storage, that give guarantee of the fingerprint integrity is through distributing the fingerprint result between the users in the system using error correcting codes. The function H has a short description that is shared by all parties in the distributed system, and this description uniquely defines for each string of information, its corresponding hash value. The length of this value is independent of the length of information. The infeasibility for any party in the system to generate colliding pairs is assumed.

In addition, this scheme uses an error correcting code with its coding and decoding functions denoted by C and D, respectively. Let n, k and d be the code parameters such that: the coding function C maps strings of length k into a sequence of n strings, i.e. C(S) = Sl, S2, ... Sn, and the decoding function D can reconstruct the string S from any sequence s'_1, s'_2,. . .,s'_n, as long as there are at least (d-1)/2 indices i for which $s'_i = S_i$. In this formulation d is the distance of the code, and n is the number of parties to which the fingerprint is to be distributed. The fingerprinting method assumes a piece of information L is to be stored in some location that is distributed in nature. Before it is stored, L is fingerprinted by the following steps:

Table 1: Steps for fingerprinting the file [9]

1.	Compute H (L)
2.	Compute $s_1, s_2, \ldots\ldots, s_n$ = C (H (L))
3.	Distribute to each party L, i=1, 2, ……..,n, the corresponding share s_i

At a later stage, whenever a party P in the system needs to verify the integrity of that information it performs the following steps (L' denotes the information in its current form, which may be different than the original, L):

Table 2: Steps for retrieving original file [9]

1.	Require from each party i, i = 1, ... n, its piece of fingerprint corresponding to the information L; denote the returned piece by s'_i
2.	compute $D(s'_1, s'_2, \ldots, s'_n)$ and H(L')

if both computations agree accept L' as the correct information L, otherwise reject it as corrupted.

Advantages of this scheme are as follows:

a) It helps to know which piece of data is modified by comparing fingerprints.
b) Encryption keys are not used.
c) Size of shares is small and thus this scheme is space efficient as well.

3.4 A tree based recursive information hiding scheme.

In [13], another scheme for dividing secret into shares and reconstructing the secret back from its shares is explained. In this scheme, additional information is added in the shares of the secret. This additional information is a message and the message is retrieved along with file (secret) on reconstructing the file (secret). Reconstruction of correct message shows the data is same as it was submitted which insures integrity of data (secret file). It is a k-out-of-n recursive secret sharing scheme which is based on n-ary tree data structure. In this scheme the user encodes extra secrets in the partitions of secret without increasing the size of original partition hence decreasing the effective size of partitions per secret hence

increases the space efficiency of sharing the secret. This space efficiency is achieved with a trade off in security. This scheme is having application in area of secure distributed storage and Information dispersal algorithms.

Advantages of this scheme are as follows:

a) Integrity of data/secret is ensured.
b) Data/secret can be reconstructed even if some (n-k) pieces of secret are damaged. The limit is (n-k), where n is the total number of shares of the secret and k is the least number of shares required to reconstruct the secret. According to this algorithm, if (n-k) servers are damaged out of n servers, where shares of secret are available, even then file can be retrieved. Hence, if a hacker or espionage employee tries to destroy some servers and is able to destroy very few, the data can still be retrieved. In cloud systems, this kind of attack can occur especially in public cloud where everyone is welcome to use the services of the cloud. Such attacks are less in number in private cloud; because, of added security features which are not available in public cloud. Like in private cloud, the computers or nodes that can take services have their IP addresses registered. So, only registered IP addresses are allowed to use the services.
c) If unauthorized person will access data from within a particular number of servers then the person will not be able to get information about the secret file. This algorithm comes under the category of k-out-of-n scheme such that at least k shares are required to reconstruct a message and file. If unauthorized person will gain access to even (k-1) shares, then also file and the message will not be retrieved. In cloud systems, this kind of attack can also occur. The unauthorized person can be a hacker, criminal or espionage employee. Whoever the person be; but, if the person is able to access only upto (k-1) shares, then still the file will not be retrieved.

Shortcomings of this scheme are as follows:

a) Space efficiency is achieved with a trade off against security.
b) If some pieces of secret are modified, it is determined that secret is modified with the help of status of message; but, it cannot be determined which piece is modified. If number of modified servers along with number of destroyed servers together are less than or equal to (n-k), then the file/secret can be retrieved. This is done by considering the modified share as damaged. But, the problem is that it cannot be determined which share is modified by implementing this algorithm. If any share is modified and it is retrieved from server to reconstruct the file, then the value of message will be incorrect. This ensures that one of the available shares is modified. But, it cannot be known which one is modified.

3.5 Online data storage using implicit security

In this scheme [14] data partitioning scheme is described for implementing security using roots of polynomial in finite field. The servers on the cloud are randomly chosed in order to store fragments of data. The data is reconstructed by access the predefined number of servers. This scheme has two parts the first part is known as (k, k) parting scheme. Here all the k parts are required to create the data. Second part is the extension of first part known as (k, n) partition method. Here k is less than or equal to n and k greater than or equal to 2 means k partitions are needed from n partitions to reconstruct the data.

Advantage of this scheme is as follows:

a) Partitioned data pieces cannot reveal any user information.

Shortcoming of this scheme is as follows:

a) In case user forgot in which server the partitioned data is stored, it will become difficult to reconstruct original data.

4. CONCLUSION

As of now the security has been implemented in clouds using firewalls and encryption techniques. Confidentiality of data is ensured with the help of strong cryptographic techniques that use encryption keys, availability is ensured by adding redundant copies and integrity is ensured by comparing the checksums of the redundant copies. These techniques are the examples of explicit security. On the other hand Information Dispersal Algorithms and Secret Sharing algorithms provide implicit security which helps in survivability and protection of system against data loss. Implicit security is not being used for cloud systems. Hence, there is a need to implement implicit security in clouds which support multi-tenancy architecture.

5. REFERENCES

[1] Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud computing and grid computing 360-degree compared. In Grid Computing Environments Workshop, 2008. GCE'08 (pp. 1-10). IEEE.

[2] Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., ... & Stoica, I. (2009). Above the clouds: A Berkeley view of cloud computing. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, 28, 13.

[3] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616

[4] Mell, P., & Grance, T. (2009). The NIST definition of cloud computing. National Institute of Standards and Technology, 53(6), 50.

[5] Miller, M. (2008). Cloud computing: Web-based applications that change the way you work and collaborate online. Que publishing.

[6] Rimal, B. P., Choi, E., & Lumb, I. (2009, August). A taxonomy and survey of cloud computing systems. In INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on (pp. 44-51). IEEE.

[7] Seiger, R., Groß, S., & Schill, A. (2011, September). SecCSIE: a secure cloud storage integrator for enterprises. In Commerce and Enterprise Computing (CEC), 2011 IEEE 13th Conference on (pp. 252-255). IEEE.

[8] Garay, J. A., Gennaro, R., Jutla, C., & Rabin, T. (2000). Secure distributed storage and retrieval. Theoretical Computer Science, 243(1), 363-389.

[9] Krawczyk, H. (1993, September). Distributed fingerprints and secure information dispersal. In Proceedings of the twelfth annual ACM symposium on Principles of distributed computing (pp. 207-218). ACM.

[10] A. De Santis and B. Masucci. (2002). On information dispersal algorithms. In Proceedings of IEEE International Symposium on Information Theory, (410). IEEE.

[11] Rabin, M. O. (1989). Efficient dispersal of information for security, load balancing, and fault tolerance. Journal of the ACM (JACM), 36(2), 335-348.

[12] Shamir, A. (1979). How to share a secret. Communications of the ACM, 22(11), 612-613.

[13] Parakh, A., & Kak, S. (2010, May). A tree based recursive information hiding scheme. In Communications (ICC), 2010 IEEE International Conference on (pp. 1-5). IEEE.

[14] Parakh, A., & Kak, S. (2009). Online data storage using implicit security. Information Sciences, 179(19), 3323-3331.