ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

A Review on Attribute-Based Encryption based Integrity Auditing for Secure Outsourced Storage in Cloud

Saloni Atre [1], Mayank Namdev[2]

P.G. Student, Department of Computer Science and Engineering, SIRT, Bhopal, India[1]

Professor, Department of Computer Science and Engineering, SIRT, Bhopal, India[2]

ABSTRACT: Cloud computing is an enormous area which shares huge amount of data over cloud services and it has been increasing with its on-demand technology. Since, with these versatile cloud services, when the delicate data stored within the cloud storage servers, there are some difficulties which has to be managed like its Security Issues, Data Privacy, Data Confidentiality, Data Sharing and its integrity over the cloud servers dynamically. Also, the authenticity and data access control should be maintained in this wide environment. Thus, Attribute based Encryption (ABE) is a significant version of cryptographic technique in the cloud computing environment. Data integrity, one of the most burning challenges in secure cloud storage. Data auditing protocols enable a verifier to efficiently check the integrity of the files without downloading the entire file from the cloud. In this paper cloud data integrity checking is performed by introducing attribute-based cloud data auditing where users can upload files to cloud through some set of attributes and specify auditor to check the integrity of data files. Hence, in this, review it has been discussed about the various security techniques and relations based on Attributes Based Encryption over data attributes which explains secured methods & its schemes related to time specifications.

KEYWORDS: Cloud Computing, Data Integrity, Data Auditing, Data sharing, Security, ABE

I. INTRODUCTION

Now-a-days, in this big environment, data sharing over the cloud servers are highly accessible but has not remained secured, as the cloud service providers can't be trusted for long, but there are various developments which have been done by different authors in this cryptographic area. Cloud computing is defined as 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 [1]. Services catered by cloud computing are software as a service (SaaS), platform as a service (PaaS) and hardware as a service (HaaS). Amazon, Google, Microsoft, IBM are key companies in cloud computing. At present a lot of users outsource their data to the websites hosted by these companies. According to IDC the overall expenditure on software, storage structures, and licensed business by the public cloud service providers will escalate at a 21.9% Compound Annual Growth Rate (CAGR) to $12.2 billion in 2016 [2]. Lending data storage space is pivotal service of cloud computing. This service allows business organizations and individuals to shift their data from personal data centers to cloud based data servers. Moving data into the cloud servers lends much contentment to organizations and individuals since they need not to anguish about the management of complex hardware systems. However, once the ownership of data is dropped, it brings security and privacy issues with data. Without data security, success of cloud computing is abridged. Maintaining data integrity is one of the vital security concerns.

By outsourcing data, the data owner gave right to cloud service provider to perform any operation on data. Hence data owner suffers from loss of possession of data. Possession of data states the control of data which means that if data is on local systems then data owner has full control over any operation performed on data including block deletion, modification, and insertion. But if the data is on cloud storage server then cloud provider has all the power to control

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

any operation performed on the data. Cloud provider can stop any operation on data, process any operation incorrectly and may produce incorrect results. The major problem with loss of data possession is that the cloud provider can hide such mistakes from data owner for some benefits. The cloud server may also face internal and external security issues including components failure, administration problems, and software bugs which can harm data owner's critical data.

This third party data controlling has endangered data integrity and thereby hindering successful adoption of cloud environment by individuals and organizations [3]. Checking data integrity when accessed is common for assuring data possession, but considering the amount of data stored at cloud, checking data integrity when accessed is not efficient. Moreover it is inapropos to let cloud providers or the data owners to audit data integrity as there is no guarantee for neutral auditing. Also, in these complex, voluminous data storage systems, the data may be refurbished from time to time and the prior data auditing protocols devised for static data archives may not be appropriate for data auditing in present scenario.

Here in this scenario an authoritarian auditing service is required to audit data integrity in cloud periodically. In recent years checking data integrity at remote server without having to access whole data has gained much attention of researchers.

In this paper the various data integrity auditing protocols have been studied. Our main contributions are as follows:

a. The various system models and threat models for outsourcing data in cloud have been discussed.
b. A comparative analysis of these protocols on the basis of security methods, storage overheads, computation costs, communication cost, etc.
c. The challenges for actualizing an efficient data integrity auditing protocol have been highlighted.

II. VARIOUS MODELS FOR DATA INTEGRITY AUDITING IN CLOUD

During recent years the issue of data auditing in cloud computing has scored more significance and a number of protocols have been suggested by many researchers. There are various models of a auditing protocol. These models are discussed below:

A. Private Auditing Protocols

Few prior auditing protocols allow only data owners to audit data integrity. Such systems involving data owner and cloud server is termed as private auditing system. Role of two entities in private auditing system is explained below:

Data owner: is the owner of data, consist of both individuals and organizations. Data owner is dependent on cloud service provider for proper maintenance of data.

Cloud Storage Server (CSS): provides data storage space to data owner. Cloud service provider is responsible for handling the cloud storage servers. This entity is considered to be untrusted.

This type of auditing consists of only two entities: Data owner and CSS. In this model we presume the case of an individual writer and numerous readers. This system provides authority only to data owner/writer to interact with the cloud storage server to audit data integrity and perform data structure operations on outsourced data, while the readers just have the authority of reading data.

B. Public Auditing Protocols

The protocol that allows a third party other then data owner to audit data integrity is termed as public auditing system. This system includes data owner, CSS and a third party auditor. Role of Third party auditor in public auditing system is explained below:

Public auditable system model is shown below in Figure 1. It consists of two types of interactions. First interaction is between Data owner and TPA demonstrated by Figure 1a. During first interaction, the data owners generates cryptographic keys, computes metadata of their data to be stored on cloud server. Then data owner exchange the cryptographic keys with the TPA and stores data on cloud and goes off line. Second interaction is between TPA and CSS as shown in Figure 1b. The integrity auditing is done via an interrogation/feedback auditing protocol. This interrogation feedback auditing protocol has three phases: Challenge, Proof, and Verification. Whenever TPA needs to check possession of data it throws a challenge to the remote server. As a response, the server prepares a proof of possession of data, which is replied to the TPA. The TPA authenticates the proof for its accuracy using the

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

cryptographic keys available and prepares a report of possession of data. This report is then communicated back to the data owner as demonstrated in Figure 1c. This audit report from TPA will guide the data owner to judge their enrolled cloud service provider, and it is beneficial for the cloud providers to lift up their cloud based platforms [5].

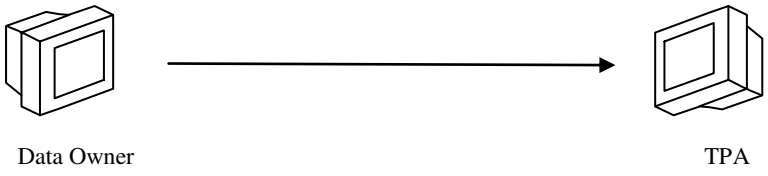

Figure 1a: System Initialization phase

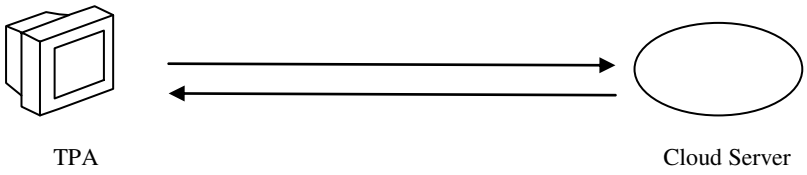

Figure 1b: Challenge-Response phase

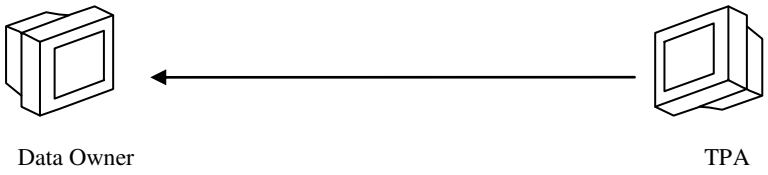

Figure 1c: Integrity auditing outcome

III. THREAT MODELS

As previously discussed, in public auditing of data, there are three entities involved: data owner, TPA, CSS. The TPA and CSS can pose some threats to owner's data as follows:

A. Threats from TPA

In the public auditing of data, the data owner completely relies on a TPA for the storage correctness of data assuming this independent TPA to be honest and reliable. However it is possible that the TPA is inquisitive about the cloud data. Thus the owner has risk to privacy of data in public auditing mechanism. Therefore along with storage correctness assurance of owner's data at the cloud server there is a need of a privacy protection mechanism so that no data may expose to the TPA during the integrity auditing process.

B. Threats from cloud storage server

a. The petty strangers have the proficiency to intrude the cloud server and may contaminate or erase owner's data without being discovered.

b. The cloud storage server may intentionally remove rarely accessed data without any notification to data owner in order to save space.

c. The cloud storage server can process some operation on data incorrectly due to system failure or any other reason and hide its mistake from data owner to protect its image, thereby causing damage to owner's data.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

d. It is possible that the challenged data block and its metadata are damaged at cloud server and during an auditing query the cloud storage server may use some other legitimate and authentic pair of a data block and its corresponding metadata as a substitute to the queried data blocks just to pass out the audit.

e. The cloud storage server may generate the proof of possession of data which is frequently challenged by the auditor from the previous stored results, rather than actually querying the owner's data.

IV. METHODS FOR DATA INTEGRITY AUDITING PROTOCOLS

The data integrity auditing protocols analyzed in this paper is different from each other. The difference lies in different underlying concepts used in realizing a auditing protocol. This section provides a brief introduction to the underlying mathematical and cryptographical concepts used in most of the data integrity auditing protocols. It is postulated that the following concepts may provide a base to understand a auditing protocol.

A. Mathematical methods

In this section we briefly highlight the background mathematical methods used for constructing a data integrity auditing protocol. This includes bilinear pairings, discrete logarithmic problem, diffie hellman problem, homomorphism and homomorphic encryption.

Bilinear Pairings

Let G1, G2 denote two cyclic groups of prime order 'r', generated by P, where G1 being an additive G2 being a multiplicative group. A pairing is defined as a map e: $G1 \times G1 \rightarrow G2$ satisfying the properties below:

a. Bilinearity: $\forall a, b \in Zq , \forall P \in G1, \forall Q \in G1 : e (Pa, Qb) = e (P, Q)^{ab}$
b. Non-degeneracy: $e(P, Q) \neq 1$ where $P, Q \in G1$
c. Computability: ∃an efficient algorithm which can compute 'e'

These pairings have applications in cryptographic schemes. These allow users to create new schemes such as BLS signature scheme. Among all the signature methods in cryptography, BLS signatures have shortest length [6].

B. Diffie Hellman Problem (DHP)

Let Zp* denotes a cyclic group of multiplication with g as generator and p being a prime number. The discrete logarithm problem defined in ZP* is considered to be a computationally hard problem. DHP is stated as: For random $a, b \in \{0, 1,\ldots, p-1\}$, $g^a \pmod p$, $g^b \pmod p \in Z^*$ to find $g^{ab} \pmod p$ is computationally hard. DHP is closely related to the difficulty of computing the DLP over a cyclic group. [7] provides detailed study on DHP.

C. Homomorphism and Homomorphic Encryption

Mathematically group homomorphism is defined as: Given two groups $(G,\oplus)$ and $(H, \otimes)$, where $\oplus$ and $\otimes$ denotes addition and multiplication operations in G and H, there exist a function $e: G \rightarrow H$ such that for all a, b in G following holds:

$$e (a \oplus b) = e (a) \otimes e (b)$$

Homomorphic encryption is a type of encryption which permits users to perform some operations on cipher text and get a result which on decryption is equivalent to the result of operations carried out on plaintext. The homomorphic property of various cryptosystems is useful for developing collision-resistant hash functions, secure voting systems, and personal information recovery systems and it allows extensive usage of cloud computing by assuring the confidentiality of refined data. In 2009, Gentry's proposed a Fully Homomorphic Encryption (FHE) scheme. IBM's big step in cryptography is the release of HElib where HE stands for homomorphic encryption. [8] provides details on homomorphism and homomorphic encryption. Specific to this

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

review the homomorphic tags (BLS-based) used by Shacham et al. and the homomorphic tag (RSA-based) used by Ateniese et al. have been specified using homomorphism [9].

D. MAC Based Solution

The massage authenticator code is used to authenticate the block of data in which client upload block of data where uploaded data go through MAC to cloud sever which provide secret key to third party auditing. So, limitation on these solution are TPA acquire priori information of data blocks for identification , data files to be verify as secret key are fixed.

E. RSA-based Solution

Sebe et al. improved Filho's protocol by dividing data into blocks and generating an RSA-based homomorphic hash value for each data block. Data possession checking protocol is based on the Diffie-Hellman key exchange method[10].
Parameter initialization: The algorithm use the Verifier, the Prover to denote the owner who has data stored in the cloud as well as has responsibility for data auditing, and the cloud server provider (or cloud server) respectively. Let N = pq be an RSA modulus generated by the Verifier, where p and q are primes such that p = (p − 1)/2 and q = (q − 1)/2 are also primes. N is public while φ(N)=(p − 1)(q − 1) is secret and private (only known by the Verifier). Let l be the positive integer chosen to trade off between the storage requirements at the Verifier and the computational cost at the Prover. Let t be a security parameter and PRNG be a pseudorandom number generator generating t-bit integers. Splitting the input data m into l-bit blocks m_1, m_2, ..., m_n(n =[m/l]). The last block is padded with 0s in the most significant bit positions if its length is less than l. A homomorphic hash value $M_i = m_i \bmod \varphi(N)$ is computed and stored for each data block m_i.
Challenge-response verification protocol: Initially, the Verifier generates a random seed S and a random element $a \in Z_N \setminus \{1, N-1\}$, then sends the challenge (a, S) to the Prover. After receiving the challenge (a, S), the Prover generates the n pseudorandom numbers $ci \in [1, 2^t]$ (i = 1, ..., n), using PRNG seeded by S, computing $r = \sum c_i m_i$ and $R = a^r \bmod N$. After that, the Prover sends the proof R to the Verifier. Upon reception, the Verifier regenerates the n pseudorandom values $c_i \in [1, 2t]$, for $1 \le i \le n$ by using P RNG seeded by S, computing $r = \sum c_i m_i \bmod \varphi(N)$ and $R' = a^{r'} \bmod N$. Finally, the Verifier checks whether R = R'. If true, the Verifier is convinced that m is stored correctly.
Algorithm: Firstly, the number of data blocks is computed, followed by splitting the data m into blocks. Then generating two prime numbers p and q with the same size of k bits, computing φ(N) and N. After that, a homomorphic hash value is computed for each data block. The next step is generating random values (a, S), generating pseudorandom values c_i , computing r, R. Lastly, the pseudorandom values c_i are regenerated and r', R' are computed.

F. Paillier-based Solution

Parameter initialization: Here there is reuse of Verifier, the Prover to denote the owner who has data stored in the cloud as well as has responsibility for data auditing, and the cloud server provider (or cloud server) respectively. Let p and q are two large prime numbers chosen randomly and independently of each other such that gcd(pq,(p − 1)(q − 1)) = 1. N is public while $\varphi(N^2) = N(p-1)(q-1)$ is secret and private (only known by the Verifier). Let l be the positive integer chosen to trade off between the storage requirements at the Verifier and the computational cost at the Prover. Let t be a security parameter and PRNG be a pseudorandom number generator generating t-bit integers. Splitting the input data m into l-bit blocks m1, m2,...., mn(n =[m/l]). The last block is padded with 0s in the most significant bit positions if its length is less than l. A homomorphic hash value Mi = mi mod $\varphi(N^2)$ is computed and stored for each data block mi. 2)
Challenge-response verification protocol: To begin with, the Verifier generates a random seed S and a random element $g \in Z^*_{N^2}$, $a \in Z^*_N$, then sends the challenge (g, a, S) to the Prover. After receiving the challenge (g, a, S), the Prover generates the n pseudorandom values $c_i \in [1, 2^t]$(i = 1, ..., n) , using PRNG seeded by S, computing r = n i=1 cimi and $r = \sum c_i m_i$ and $R = g^r a^N \bmod N$. Subsequently, the Prover sends the proof R to the Verifier. After reception, the Verifier regenerates the n pseudorandom values $c_i \in [1, 2^t]$($1 \le i \le n$) by using P RNG seeded by S, computing $r' = \sum c_i m_i \bmod \varphi(N^2)$ and $R' = gr'aN \bmod N^2$. The last step is that the Verifier checks whether R = R'. If true, the Verifier is convinced that m is stored correctly.
Algorithm: Firstly, the number of data blocks is computed followed by splitting the data m into blocks. Generating two prime numbers p and q with the same size of k bits, computing N,N^2, $\varphi(N^2)$. After that, a homomorphic hash value is computed for each data block. The following step is generating random values (g, a, S), generating pseudorandom values c_i, computing r,R. Lastly, the pseudorandom values c_i are regenerated and r',R' are computed[11].

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

Yong Yu et al. [12] proposed an attribute-based cloud data integrity auditing protocol to simplify the key management issues. The proposed technique have less calculation in verifying the Response of auditing and thus cause less time consumption.
Y. Li et al. [13] designed an fuzzy identity based attribute-based cloud data auditing protocol. Protocol offers the property of error-tolerance.
Ming-quan et al. [14] proposed Homomorphic Encryption Scheme Based on Elliptic Curve Cryptography for Privacy Protection of Cloud Computing. This algorithm achieves better efficiency in terms of computation and communication cost as compared to RSA & Paillier scheme.
Yong Yu et al. [15] investigated data privacy issues in remote data integrity-checking protocols. This algorithm practically proved that the best size of the block is between 4 and 8 KB, which delivers the best performance
G. Yamamoto et al. [16] proposed an efficient scheme by offering batch processing based on the homomorphic hash function.

V. PROPOSED SYSTEM

An attribute-based cloud data integrity auditing protocol involves four entities:
Key Generation Centre (KGC) : KGC takes charge of generating user's private key according to their attribute set.
Cloud Users
Cloud Servers
TPA : TPA is a third party designated to verify the cloud data's integrity on behalf of cloud users upon audit request.

The details of an attribute based cloud data integrity auditing protocol are described below:

a. A cloud user forwards set of attributes to KGC to request secret key.
b. KGC generates a secret key for the user with the master key and the user's attributes.
c. The cloud user generates metadata of the on encrypted data file i.e. signature. The user then uploads the encrypted data file together with the corresponding Signature to the cloud.
d. Upon receiving the auditing request, TPA and the cloud server execute a homomorphic algorithm based challenge-response protocol to verify the stored file.

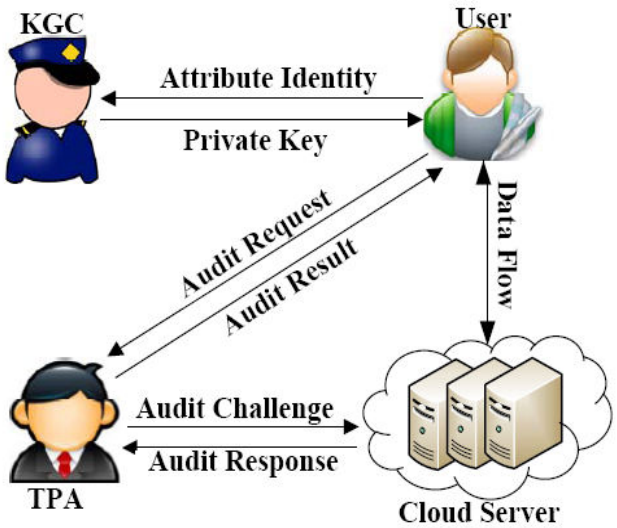

Figure 2: Proposed Architecture

This primitive consists of the following four algorithms:

a. Setup(k): This algorithm generates master key MK and public parameters PK.
b. Extract(MK, A): This is a algorithm which takes a master key MK and attributes set A as input to generates secret key SKA for the user.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

c. Sign(PK, SKA, M): This is a algorithm which takes the public parameter PK, a secret key SKA, and a message M as input. It outputs a signature of the data file.
d. Verify (PK, B, M): This is a deterministic algorithm which takes the public parameter PK, an attribute set B, the message M and its alleged signature as input. It returns 1 or 0 to indicate the signature is valid or not.

VI. CONCLUSION

In this survey various data integrity auditing methods for cloud computing has been discussed. As described clearly in the analysis that most of the existing protocols targets to provide integrity verification for various data storage systems however they lack in providing full support to data dynamic operations, public auditability and preserving data privacy. The fundamental requirements that an integrity protocol must meet are also discussed. In designing data integrity auditing protocol great care is needed to ensure that it is efficient and secure and fulfils fundamental requirements. The proposed protocol can achieve the property of attribute privacy-preserving of data files which simplify the key management issue in traditional cloud data auditing schemes. This implementation of proposed system will illustrates the practicality and efficiency of the system. The proposed system may provides a privacy-preserving guarantee that reveals nothing to TPA but the attributes chosen by cloud server when executing the auditing protocols.

REFERENCES

1. M. Xie, H. Wang, J. Yin, X. Meng, Integrity auditing of outsourced data,in: Proceeding of VLDB'07, University of Vienna, Austria, Sep.23-27, 2007,pp. 782-793.
2. Mell, Peter, and Tim Grance. The NIST definition of cloud computing, 2011.
3. Zissis, Dimitrios, and Dimitrios Lekkas, "Addressing cloud computing security issues", Future Generation computer systems, 2012, pp. 583-592.
4. Cong Wang, Sherman SM Chow, Qian Wang, Kui Ren, and Wenjing Lou. Privacy Preserving Public Auditing for Secure Cloud Storage. Computers, IEEE Transactions, 2013, pp. 362–375.
5. Shah MA, Baker M, Mogul JC, Swaminathan R., "Auditing to Keep Online Storage Services Honest", In: The Proceedings of USENIX HotOS, 2007.
6. Zhang F, Safavi-Naini R, Susilo W, "An efficient signature scheme from bilinear pairings and its applications", Public Key Cryptography–PKC. Berlin: Springer-Heidelberg; pp. 277-290, 2004.
7. Boneh D. "The decision diffie-hellman problem, Algorithmic number theory", Berlin: Springer-Heidelberg, pp. 48-63, 1998.
8. Homomorphic Encryption,http://en.wikipedia.org/wiki/Homomorphic_encryption [accessed in 2015].
9. Shacham H, Waters B., "Compact proofs of retrievability", Advances in Cryptology. Berlin: Springer-Heidelberg, pp. 90-107, 2008.
10. Hovav Shacham, Brent Waters, Compact proofs of retrievability, Advances in Cryptology, 2008, pp. 90-107.
11. Cong Wang, Sherman S.-M. Chow, Qian Wang, Kui Ren, Wenjing Lou, Privacy-preserving public auditing for secure cloud storage, Computers, IEEE Transactions, 2011, pp. 362-375.
12. Yong Yu, Yannan Li, Bo Yang, Willy Susilo, Guoming Yang and Jian Bai, "for Reliable Cloud Storage Systems". IEEE Transactions on Dependable and SAttribute-Based Cloud Data Integrity Auditing for Secure Outsourced Storage", IEEE Transaction on Emerging Topics in Computing, Vol. 14, No. 8, 2017.
13. Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, K-K. R. Choo. "Fuzzy Identity-Based Data Integrity Auditing ecure Computing, 2017.
14. Ming-quan Hong,Wen-bo Zhao, Peng-yu Wang, "Homomorphic Encryption Scheme Based on Elliptic Curve Cryptography for Privacy Protection of Cloud Computing", IEEE, 2016.
15. Yong Yu · Man Ho Au · Yi Mu · Shaohua Tang · Jian Ren · Willy Susilo · Liju Dong, "Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage", Springer, 2014.
16. G. Yamamoto, S. Oda, K. Aoki. "Fast integrity for large data". Proc. ECRYPT workshop Software Performance Enhancement for Encryption and Decryption. Amsterdam, Netherlands 2007, 21-32