A Secure Protocol for Data storage Security in cloud computing

Kartik Sharma, Renuka Sharma, Gitesh Dalal

Abstract— In cloud computing, security is the biggest challenge and also a big issue to many cloud service providers and researchers. In cloud, two major challenges to service providers are - How we can give better data security and how can we keep user private data highly confidential? We design this protocol with the aim of solving these types of problems. Our protocol is based on the E.C.C method and Random sampling sobol sequence method. By using this method we can minimize the data loss and user becomes confident about the data security. This protocol is highly applicable for those users who have less resources and limited computing capability. For the cloud service provider, this protocol allows third part auditor or verifier to periodically verify the data integrity without receiving the original data again and again. This protocol concerns the data security, and confidentiality, integrity and it never reveals the data contents to the attackers. This protocol takes less time to detect the data corruption and data loss and verify the integrity of data by using Sobol method. The proposed protocol is very safe, secure and efficient and it provides same security level for dynamic data operation too.

Index Terms— Cloud computing, Protocol, Security, Integrity, Confidentiality, Eliptical Curve Cryptography, Dynamic Operation.

1 INTRODUCTION

Cloud computing is a technology which is aimed at delivering the technological services on demand, on a pay as- per your usage. Abilities like scalability, availability, reliability, flexibility, security and service make cloud computing essentially a technology for future. As per estimates more than eighty percent of the world's computing and data storage is supposed to occur in Cloud. Cloud computing technology has certain positivity and negativity associated to it regarding the data security and privacy of services to consumer. The most important technique in the field of I.T sector is cloud computing where the data is stored in large amount and we can access this data from anywhere. But here also a big question arises i.e. how secure the data is in cloud? So here we have proposed a fine scheme to reach utmost safety of data from the assaulters with the use of cryptographic method. In security issue we have emphasized on the significance of ensuring remote data integrity. Security is always a major concern and in cloud computing this security level is fulfilled by exploring it in various security challenges. But still some problems occur like, loss of control of data from the user in cloud computing. The cryptographic primitives can not be adopted directly thus the verification of data is done with the devoid of actual data, an immense drawback. Thus the verification process becomes more challenging. The data stored in cloud is open to the attackers or brokers no matter how much the data is secured and protected. So for highly secured data we have projected the isolation of the encryption and decryption processes from the cloud to a broker service that is trusted by both the cloud provider and the cloud consumer. To attain maximum security we have divided and encrypted the data with the help of extremely secured processors so that the data is protected from unfair means. Since the owner of data loses his control over his own data when he stores his data in cloud so it's a common thing that the query of security arises regarding the data. We have projected a protocol using Sobol sequence and ECC for the integrity and the security of the data available in the cloud which are far better than those of RSA and other PKC methods. The Elliptic Curve Cryptography provides nearly equal security with small keys comparable to RSA and other PKC methods. In addition of these are capable of detecting the data modification if occurred in the absence of the authenticated dealer. In this paper we also proposed a scheme of change or modify or insert or delete or reorder the data, stored in the cloud. In our design the encryption of the data is done to ensure the confidentiality and then, the computation of metadata is done over the encrypted data. This is accomplished only when the consumer demands it.

2. Security Issue:

Security, reliability, confidentiality, liability, privacy etc are the main concerns on the topic of cloud computing technique and the peak concern is security. It depends on the CSP that how they guarantee the client regarding these tribulations. The worry about security includes: (1) Problem Related to passive Attacks (2) Data location (3) Privacy (4) Data Integrity (5) Recovery (6) Freedom (7) Problem Related to Man in the Middle Attack (8) Long-term Viability. These asserted problems are endless, in our paper we have tried our best to resolve some of these problems with the aid of ECC method where the data is safe and secure from external threats. There are two types of securities threats that arise in cloud and these can be defined as:

2.1 INTERNAL THREATS:

These are caused internally in the cloud where the Cloud Service Provider can leak the information of the user or may modify it for its own purpose.

2.2 EXTERNAL THREATS:

These are caused by some external agents and outside party who can use the stored data of the user for some wrong purpose or leak or modify and delete the data to fulfil his own requirements. Now, this security challenge is also faced by this system. To resolve this matter the data is dispersed into many fractions regardless of the repository of the original data and this is accomplished through elliptical curve method and sobol sequence method. These methods compose integrity, and confidentiality, and are also highly proficient. These methods are far better than those of pseudorandom sequences.

3. Designing and Aim of this Protocol:

In this article we are confined to only two security issues and these are:

1) Integrity: It facilitates in the recognition of any alteration that has been occurred in the data stored in cloud. It refers to the protection of data from unauthorized deletion, modification or fabrication or we can say In general it refers to the security of the data from malicious parties.

2) Confidentiality: This ensures that the data has been accessed by the authenticated or authorized parties. In other words it makes sure that the official and the true ones has accessed the secured data.

3.1 Cloud Storage Model:

1) Cloud User: Cloud User is the person who uses the services of cloud.

2) Cloud Service Provider (CSP): The data to be stored or taken back is done through CSP. CSP manage the cloud server and provide a paid service to the user.

3) Third Party Auditor (TPA): It is also called a Verifier; if the user is suffering from lack of timing then the data verification is done by TPA or verifier.

3.2 Description:

With the help of following figure we can easily understand the cloud storage model and working of each type. In cloud computing the user is the one who stores his private data in cloud to prevent it from hazards and this is done with the help of CSP. Say soon after that, if the user require that information again then in order to access that particular data the user have to send an appeal to the CSP then the CSP will verify that whether the user is authenticated or not.

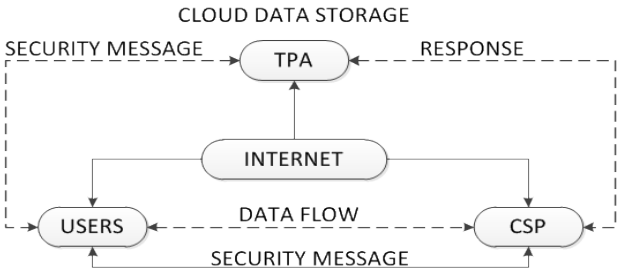

Figure 1

If authenticated then it will allow the user to access the data else not. The data accessed by the user if in encrypted form that it can be decrypted using his secrete key. And the last the TPA will keep a periodic check on the data and verify it periodically only when the user himself allows to verify the TPA. Since, in cloud computing the user loses his control over the data as soon as the data is transferred, thus the data is more prone to the errors or damage from anonymous users and attacks. The data may be lost or modified by unfair means. Thus to prevent it from these many problems an efficient and secure method is needed.

3.3 Selection of cloud deployment model:

A cloud computing based services can be deployed in three ways.

1) Community Cloud: This uses a network of computer for its purpose just like a grid.

2) Public cloud: This type of service is provided by the Google, Amazon where the data has to be secured in a private network with a confidence that it is safe there also it is based on pay basis and demands more security from a large numbers of malicious groups.

3) Private Cloud: This type of service is required in private organizations and government firms, and also where the data stored needs more care and has to be handled sensitively. These can develop their own set of rules in a cloud where only the workers of that organization can access it.

4) Hybrid Cloud: It is mixture of the two clouds defined above i.e. Private cloud and public cloud.

3.4 Simulation of elliptic curve cryptography:

This method is based on a finite field in cyclic form i.e. it stands on the field and group speculations and using these, public key is employed for high level security. The RSA and other systems also provide this key but with lesser range. This table provides a detail of ECC key size and RSA key size. And to compare the two ratios is also provided in following table 1. The following graph is formed on the basis of table 1 which describes the increase in the need of key generation as the number of users' increases. Thus the ECC require less time and less key generation as compared to RSA which is totally clear from the graph. Though in ECC, the signature generation and verification require same time as in RSA. Signature generation algorithm and key pair generation algorithm of ECDSA needs a random number to be generated. This key should be small and should be unique and should be unpredictable so that it is free from the attackers.

Type	ECC Key Size	RSA Key Size	Ratio
Type-1	110	510	1:5
Type-2	160	1020	1:5
Type-3	190	1530	1:6
Type-4	220	2050	1:9
Type-5	250	3070	1:12
Type-6	380	7680	1:20

Table 1

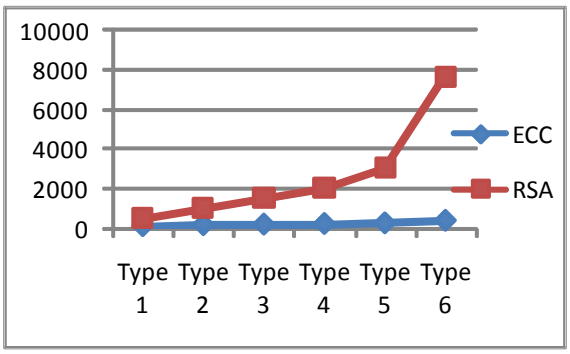

Figure 2

3.5 Representation and Definition:

Here judge subsequent parameters: F is the file to be stored in cloud which is divided into n blocks and each block have equal length m where n = [|m|/*l*] and we represent *f key*(.)- SRF (Sobol Random Function) indexed on some *key*. Defined as f : $\{0,1\}^* \times key\text{-}\{0,1\}^{\log 2\, n}$. SRP (Sobol Random Permutation) representing as π key(.)–, and its defined as $\pi_{Key}(.) : \{0,1\}^{\log 2(l)} \times key\text{–}\{0,1\}^{\log 2(l)}$.

3.6 E.C.C over a finite field and over ring:

Let K be a field of characteristic ≠ 2,3 and let a,b $\in$ K is two parameter and satisfying the equation $4a^3+27b^2 \neq 0$. An elliptical curve over K with parameter a and b is defined as the set of points (x,y) with x,y $\in$ K Satisfying the equation: $Y^2 = x^3 + ax + b$.

Consider below equation for ring *Zn, Let a, b* are the two integers in ring *Zn.*

An elliptic curve E_n (a, b) over the ring Zn is the set of points (x, y). Where (x, y)$\in Z_n \times Z_n$ satisfying the equation: y^2+ax+b, and $gcd\ (4a^3+27b^2, n) = 1$ where n is an integer

4 Efficient and Secure Storage Protocol:

Since we are confined to only ECC and sobol sequence protocol for security, integrity and confidentiality purpose so three processes will be included and these are Setup, Verification and Dynamic Data Operations.

4.1 Setup

It includes three algorithm and these are: 1) Key Production 2) Encryption 3) Metadata production.

4.1.1 Setup Explanation:

With the help of following algorithm the user can produced private key and public key pair with k inputs where k>512. Then the user is permitted to choose two large primes p and q of size k such that p≡q≡2. In step 5. N_n is a order of elliptic curve over the ring Z_n denoted by E_n *(0, b)*, and *b* is a randomly chosen integer such that *gcd(b, n)=1.* Now follow step 7, 8, 9 sequentially to achieve essential keys. With the help of this Encryption the user can ensure the confidentiality of data with *s* as inputs which are stored in F file segregated in m_i blocks , where m_i are keyed in Sobol Random Function(SRF) and secrete random parameters and produce m'_i as output as, $F = \{m_1, m_2, m_3, \ldots m_n\} = \{m_i\}_{1<i<n}$ and $F' = m'_i = m_i + f_k(s)$ where *s* is random of size *l.*

After encryption user compute the meta data over encrypted data. With the help of this Metadata production the user can verify the integrity of data by computing a metadata over encrypted data with m'i, public key and private key as inputs and T_i as data output: $T_i \leftarrow m'_i\, P(mod\ N_n))$ where $P \varepsilon\ E_n(0, b)$

After computation of metadata, the user sends metadata, public key to the TPA for later verification and sends file *F'* to cloud servers for storage.

Algorithm 1:

Step1. Procedure: KeyPrd(k) ←{ PK,PR}
Step2. Take security parameter k (k>512)
Step3. Choose two random primes p an q of size k: p≡q≡ 2
Step4. Compute n=pq
Step5. Compute Nn = lcm(p+1, q+1)
Step6. Generate random integer b<n, gcd(b, n)=1
Step7. Compute P, is a generator of En(0,b)
Step8. Private key PR={ Nn }
Step9. Public key PK={n, b, P}
Step10. Encryption(m_i , S)←m'_I
Step11. MetadataPrd(m'i ,n, b, P) ←T_i
Step12. for 1 to n
Step13. Compute $T_i \leftarrow m'_i$ P(mod Nn))
Step14. Compute $m_i' = m_i + f_k$ (S)
Step15. end for
Step16. end procedure

4.2 Verification stage:

Now arises the verification stage where the data verification is done by examining our system through various challenges and we have done it through following algorithms:

1) Challenge, 2) Proof Generation 3) Check Proof.

4.2.1 Challenge:

Here the challenge can be created even by the user by taking k_{SRF} and Q as inputs which together will produce *chal* as output where k_{SRF} and k_{SRP} are the random keys using Sobol sequence and computes random indices $1 \le i_j \le n$ (j= 1,….,c) of the set[1,n], Where, $C = \pi_{ksrp}(c)$ which prevents the server from foreseeing which blocks will be queried in each challenge. For further computation we follow, Q=rP where *r* is used to notify the server not to reuse any values from the previous challenge.

Then, verifier creates the challenge *chal* = {kSRF, j, Q}, and sends to the server. This is considered as following algorithm 2.1.

Step 1. Procedure: Challenge(j,Q, $k_{srf)}$ ←challenge
Step 2. Produce a random keys k_{SRF}, k_{SRP}
Step 3. Compute c = π_{ksrp}
Step 4. Compute Q=rPε $E_n(0,b)$
Step 5. Create Challenge ={ k_{SRF}, j, Q}
Step 6. End Procedure

4.2.2 Proof production:

As soon as the server receives a challenge then in response it will generate an integrity proof with m'i, encrypted data and chal as inputs and compute R as output by generating random numbers using Sobol random Function (SRF) i.e.

$\alpha_j = f_{kSRF}(j)$

Then compute b=$\sum_{j=1}^{c} \alpha_j m'_i$, Where $1 \le i_j \le n$

Then, Compute a response R=bQ mod n

$= \sum_{j=1}^{c} \alpha_j m'_i$, rPmodn

$=r(\sum_{j=1}^{c} \alpha_j m'_i, P \bmod n)$

Algorithm 2.2:

Step 1. Procedure: ProofPrd(m'I ,k_{SRF}, Q) ← R
Step 2. Produce "n" random numbers using k_{SRF}
Step 3. For 1 to n
Step 4. Produce $\alpha_j = f_{kSRF}(j)$
Step 5. End for
Step 6. Compute b = $\sum_{j=1}^{c} \alpha_j m'_i$,
Step 7. Compute R=bQ mod n
Step 8. End procedure

4.2.3 Check Proof:

After creating challenge and proof, now it's the turn to check proof i.e.to check the integrity using public key *pk*, challenge query *chal*, and proof production *R,* as inputs and output 0 or 1 depending on the integrity of the file verification. If the integrity of file is verified as successfully then it will generate output as 1 else 0.

The verifier re-production of random numbers using SRF i.e.

$\alpha_j = f_{kSRF} = (j)$

Then compute S= $\prod_{j=1}^{c} a T n$

S=$\prod_{j=1}^{c} \alpha_j T_i'$, mod n

R'=rS mod n

Now, verifier checks whether

R'=R,

If response is valid, then it returns 1 or else 0.

Algorithm 2.3:

Step 1. Procedure: CheckProof(T'I, r, k_{SRF}, n) ←R'
Step 2. Produce n random numbers using key k_{SRF}
Step 3. For 1 to n
Step 4. Produce $\alpha_j = f_{kSRF} = (j)$
Step 5. Compute R'=rS mod n
Step 6. Compute R'=rS mod n
Step 8. Verify if(R'=R)
Step 9. return true
Step 10. else
Step 11. return false
Step 12. End if
Step 13. End Procedure

4.3 Dynamic Data Operations

4.3.1 Prepare update:

The dynamic data operation is discussed at block level where Block Modification (BM), Block Insertion (BI) and Block Deletion (BD) can be done at any time by the server. These amendments are performed in general form ($Block_{op}$, U_b, N_b), where $Block_{op}$ indicates the block operation such as BM, BI and BD. To generate new block N_b and to update a block U_b parameters are employed by sending a request to server which in response will do the required job.

Algorithm 3: Prepare Update
Procedure:
Step 1. Prepare Update ← (BI/BM/BD,U_b, N_b)
Step 2. Select an update block mU_b
If
Step 3. (Update ==modification/insert)
Step 4. Encrypt *m*'U_b← mU_b + f_k(S)
Step 5. Compute *T*U_b ←*m*'U_b *P*mod N_n
Step 6. Update = (BM/BI, U_b, N_b)
Else if
Step 7 (update==deletion)
Step 8. Update = ((BD, U_b)
Step 9. Send update request to the server
Step 10. End if
Step 11. End procedure

4.3.2 Block Modification, Execution update and deletion.

According to above algorithm we will insert a block N_br at required spot say at r position without disconcerting the prior data/metadata in file F'. And this is possible as the metadata is not including the block index.

Using above algorithm we can even delete a block. The procedure will go reverse of above by creating a delete request (BD, r) and sending it to server for further delete operation which results in the construction of update version of the file F".

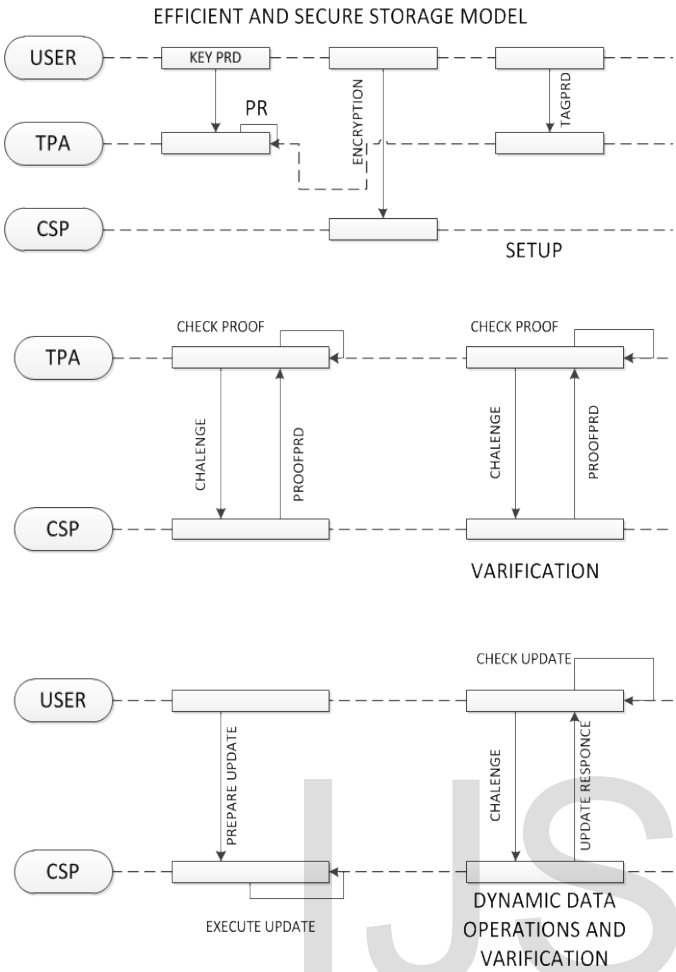

Figure 3.

(1) Block Modification (BM): This is the operation which is performed repeatedly and is accomplished using following

Algorithm 3.1

Step 1) Create a new block m_j

Step 2) Encrypt the new block using equation

$M'U_b \leftarrow mU_b + f_k(S)$

Step 3) Compute new metadata using equation

$TU_b \leftarrow m'U_b P \bmod N_n$

Step 4) Create update request (BM, U_b, N_b) and sends to the server.

Step 5) The Metadata sends to TPA for later verification Upon receiving an update request, the server replace the block N_b with $m'U_b$ and construct update version of the file F'' by running algorithm 3.2 (Execute Update).

(2) Execute Update

Algorithm 3.2, Procedure:

Step1. ExecuteUpdate← {F"}

Step2. If (updates==modification)

Step3. Replace m_i with m'_j in the file F'

Step4. Update file F"

Step5. Else if (update==insert)

Step6. Insert N_{bx} before m_i or append

Step7. Else if (update==deletion)

Step8. Delete m_i from file F'

Step9. Update the file F"

Step 10.move all blocks backward after i^{th} block

Step 11.end if

Step 12.end procedure

(3) Block Insertion and Block Deletion

The block insertion and block deletion can be put into practice using algorithm 3.2.

Algorithm 3.3 Procedure

Step 1. Create a new block N_br

Step 2. Encrypt the new block $m'N_b \leftarrow N_br + f_k(S)$

Step 3. Compute new metadata $T * U_b \leftarrow m'U_b P \bmod N_n$

Step 4. Create update request (BI, U_b, N_b) and sends to the server.

Step 5. The Metadata sends to TPA for later Verification Upon receiving the update request, the server replace the block $m'U_b$ with N_b and construct update version of the file F" by run the algorithm 3.2.

(4.3.4) Verification Update:

In cloud model for high rank security the user will confirm the integrity of metadata in above updated method as:

$Q = rP$ *i.e.* the client will initiate for proof of integrity by testing it. $RU_b \leftarrow m'U_b$ Pmod n i.e. now server will do his job of verification and then will return the result back to user. Now user will do verification by matching the result with metadata. If matched then server has been updated data successfully otherwise not.

Algorithm 3.4: Verify Update

Procedure:

Step1. Verify Update (pk, Q, R') → {1, 0}

Step2. If (update==modification/insert)

Step3. If (TU_b=RU_b)

Step4. Return 1

Step5. Else

Step6. Return 0

Step7. End if

Step8. Else if (update==deletion)

Step9. Verification directly starts from static case

Step10. End if

Step11. End procedure

5. Result and Analysis of Proposed Protocol:

5.1. Confidentiality

We rely on the rigidity of the Elliptice Curve Diffie-Hellman (ECHP) and Elliptic Curve Discrete Logarithm (ECDL) problems for confidentiality or security point of view so that the outsiders are unable to predict the way to spoil the data.

5.1.1 The proposed protocol is confidential against data leakage to attacker.

Proof:

1)The data cannot be revealed to others stored in cloud as the secret parameter is not known eavesdropper because of Elliptic Curve Diffie-Hellman (ECDH) problem. According to the ECDLP, Suppose the equation Q=rp where Q, Pε En(a, b) and r<n. it is comparatively tough to find out r given Q and P.

2) Due to this secret parameter which is chosen by the user randomly, the malicious party cannot be able to dig up anything from encrypted F'=m_i ', even if he tries some combinations of public parameters then also he will be empty handed this will happen all because of the ECDL problem.

3) Using ECDH and ECDL problems, our protocol is confidential against data leakage even from TPA too which has $T_i \leftarrow N_b \; P \; (mod \; N_n)$ because the secret key to be chosen by the user is purely confidential and purely unique and different every time thus the TPA also cannot tell or disclose the data to foes.

Integrity:

This is accomplished as:

$R' = R$

R′ = rS mod n

$S=\prod_{j=1}^{e} \alpha_j T_i$, mod n where $\alpha_j = f_k(j)$

$S=\prod_{j=1}^{e} \alpha_j m_i'$, p mod N_n) mod n

$S=\sum_{j=1}^{e} a_j m'_i$, P mod n)

R′= rS mod n

$R'= r \left(\prod_{j=1}^{e} \alpha_j m'_I, P \text{ mod} n\right)$

$R'= r^{\left(\sum_{j=1}^{e} a_j m'_i, P \text{ mod } n\right)}$

R′= R

From the above result R′=R proves this protocol is complete and valid. Here *j* is number of blocks which are chosen randomly. Now the user is probabilistically confident about the safety of data.

6. Conclusion:

In this paper we designed a protocol that solves many problems that comes in our cloud system like confidentiality and Integrity. By using this protocol we can make our cloud highly secure and efficient. We use here elliptical curve method and sobol method. We prove that the elliptical curve method generates small key size as compared to RSS method, so it can work very efficiently. Those users who have less resources and limited computing capability, they can use this method and it is most efficient method for them. It also supports public verifiability that enables TPA to verify the integrity of data without retrieving original data from the server and detects data corruptions. Our protocol is also secured at the time of Dynamic Data operation like insertion deletion and updation. The methods discussed in it are very efficient and secure and can be employed at incredibly out sided extent without fear of losing the information or disclosing the data. We also prove that the scheme is confidential and integrity and show the result, and prove this protocol is complete and valid.

References:

[1] Vic (J.R.) Winkler, "Securing the Cloud, Cloud Computer Security, Techniques and Tactics", Elsevier, 2011.

[2] H. Takabi, J.B.D. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments", Article in IEEE Security and Privacy, vol. 8, no.6, Nov- Dec. 2010, pp. 24-31.

[3] V. Miller, "Uses of elliptic curves in cryptography", advances in Cryptology, Proceedings of Crypto'85, Lecture Notes in Computer Science, 218 Springer-Verlag, pp.417-426. 1986.

[4] Z. Yang, S. Zhong, and R. Wright, "Privacy-preserving queries on encrypted data," in Proc. of the 11 European Symposium on Research In Computer Security, 2006

[5] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21(2) :120-126, 1978. Computer Science, pages 223-238. Springer, 1999.

[6] C. Wang, Q. Wang, K. Ren, N. cao and W. Lou , "Towards Secure and Dependable Storage Services in Cloud Computing", Accepted for publication in future issue of IEEE Trans. Service Computing. DOI:10.1109/TSC.2011.24.

[7] G. Caronni and M. Waldvogel, "Establishing Trust in Distributed Storage Providers", In Third IEEE P2P Conference, Linkoping 03, 2003.

[8] S. Wang, D. Agrawal, A.E. Abbadi: A Comprehensive Framework for Secure Query Processing on Relational Data in the Cloud. Secure Data Management 2011: 52-69

[9] J.Li, M. Krohn, D. Mazieres, D. Shasha. Secure untrusted data repository (SUNDR). OSDI 2004.

[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. Of CCS '07, pp. 598–609, 2007.

[11] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic."Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," Future Generation Computer Systems, vol. 25, no. 6, June 2009, pp 599–616.

[12] H.Shacham and B.Waters, "Compact Proofs of Retrievability", Proc.14th Int'l Conference Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT), LNCS 5350,2008, pp.90-107. Melborne, Austrilia.

[13] Yan Zhu, Huaixi Wang, Zexing Hu, Gail-J. Ahn, Hongxin Hu, Stephen S. Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. of the 26th ACM Symposium on Applied Computing (SAC), Tunghai University, TaiChung, Taiwan, March 21-24, 2011.

[14] L. Chen, G. Guo, "An Efficient Remote Data Possession Checking in Cloud Storage", International Journal of Digital Content Technology and its Applications. Volume 5, Number 4,April_2011.