Dhanalakshmi V, 2023, 11:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

ECC-Encrypted Secure Cloud Storage with Verifiable Data Sharing Ensures Privacy, Integrity, and Trusted Access Control

Dhanalakshmi V, Keerthana K, Saranya T, Sri Dharshini M
Department of Computer Science and Engineering
Kongunadu College of Engineering and Technology Tamilnadu,India

Abstract- Storage and IT services are expanding so quickly that more data centers and server rooms are necessary for quick processing in the allotted time. Web-based computing is the product of an enormous shift in the way digital services as well as information technologies (IT) are provided and acquired. At the moment, there has been a rise in the trend of outsourcing data to distant clouds, where users contract with cloud service providers (CSPs) who supply large storage capacities at affordable prices. As a result, users can lessen the strain and upkeep associated with local data storage. In the meantime, they lose control over their data once it is stored in the cloud, which unavoidably results in additional security threats to confidentiality and integrity. Therefore, successful and effective techniques are required to guarantee the confidentiality and integrity of rented data on trusted cloud servers. But in order to use cloud computing, businesses must have faith that a service provider's platforms are safe and offer enough data confidentiality for their clients. In order to solve these problems, we offer a secure and effective protocol in this work. Elliptic Curve Cryptography and Sobol Sequence (random sampling) form the foundation of our approach. With our approach, a third-party auditor (TPA) can routinely confirm the accuracy of the data kept at CSP without having to access the original data. By sending a modest, consistent quantity of statistics, the challenge-response protocol saves network communication. Most significantly, our protocol is private; malevolent parties never get access to the data contents. While keeping the same extent of security, the recommended approach also takes dynamic data processing at the file level to account. Our scheme is more secure and effective when compared to current schemes.

Keywords- Cloud computing, Sobol Sequence, Elliptic Curve Cryptography (ECC), data storage, integrity, confidentiality, TPA, and CSP.

I. INTRODUCTION

Cloud computing is a rapidly expanding concept that turns data centers into large-scale computing services using internet-based technology. Users can now subscribe to high-quality services from data and software on distant server facilities thanks to more affordable and robust processors and software as a service (SaaS) architecture. Storage space and adaptable computer equipment are

offered via suppliers of cloud computing such as Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2). This change, however, removes the need for physical computers to maintain data, leaving users to rely on their cloud service providers for data availability and integrity. Concerns about the confidentiality and integrity of data preserved in the cloud are brought up by this lack of control. Honesty and privacy are essential to cloud operation.

- The CSP, whose primary goal is to turn a profit and preserve its reputation, has purposefully concealed data loss and incidents that users hardly ever view.
- The malevolent CSP might perhaps erase some data or quickly acquire all the information and sell it to the company's largest rival.
- Sensitive user data and some crucial business secrets can be obtained by a criminal who stops and records the communications.
- Numerous internal and external threats might affect cloud systems.

The method known as remote data integrity testing focuses on how often and successfully we verify to see if the cloud server can reliably keep user data without retrieving it. The user creates some metadata in this protocol. He can then use the challenge-response technique to test the server's integrity of specific file blocks. The server then responds to the appropriate challenge made by the verifier, who could be the original user or an impartial third party, claiming that it still has the data in its original form. Different versions of remote data integrity checking protocols under various cryptographic systems have recently been presented by a number of researchers [6]. All of these protocols, meanwhile, concentrate on verifying static data. Dynamic data adaptability for a variety of uses is one of the design tenets of cloud storage. This indicates that in addition to being accessible by users, the data saved to the cloud is also continually updated by block operations like modification, insert, and deletion. Therefore, creating a more effective and safe method to provide dynamic audit services is essential. In [7], means of confirming current information in the cloud been suggested. Although the goal of currently available schemes is to serve as integrity verification for various data storage systems, the issue of data secrecy has not yet been entirely resolved. To guarantee the privacy and reliability of remote facts, protocols [8] have been put forward. However, because these techniques rely on pseudorandom patterns to verify the honesty of exported data, which do not include all the information while computing the integrity proof, none of them can offer customers a good security assurance.

Consequently, consumers are not guaranteed privacy of their data by statistical authentication systems that utilize pseudorandom sequences. In order to guarantee availability of information and truthfulness, Syam et al. [9] devised a distributed validation protocol leveraging the Sobol sequence; yet the technique does not tackle the problem of data secrecy. In cloud computing, it is still impossible to figure out a way to construct a safe and effective design that takes into account these two crucial elements for information retention services. With shorter keys, the ECC may offer levels of safety on par with RSA and other PKC techniques. It is made for gadgets like smartcards, PDAs, and cell phones which have little memory and/or processor power. The key strength, or the difficulty of cracking the key and getting the plain text back, is a crucial component. Our design computes information over sensitive information after the user protects the data in order to ensure confidentiality. The integrity can then be confirmed by the tester using a remote data integrity checking protocol. Any modifications to the data saved in the cloud should be detectable by the verifier. Our scheme's security depends on how difficult certain Elliptic Curve Cryptography problems are. In contrast to current methods, ours has a number of advantages:

- Since we are challenging the server for its authenticity verification using a Sobol sequence versus a pseudorandom sequence, it ought to detect any data corruption when someone removes or alters the data in cloud storage.
- Our strategy ensures the data confidentiality.

Because its key size is smaller than that of RSA-based solutions, it is computationally and storage-efficient.

The remainder of the document is structured as follows: The concepts of ECC and Sobol sequence are introduced in Section II, along with the need for CSP to implement them in order to protect data (in terms of integrity and confidentiality). The system model is introduced in Section III, together with the cloud storage model, security risks, design objectives, notations, and permutations. We give a thorough explanation of our plan in Section IV, and in Section V, we compare it to other plans and present the suggested execution. The final statement of the whole study is offered in section VI.

II. ECC, SOBOL SEQUENCE

Elliptic curve cryptography (ECC)

The algebraic structure of elliptic curves over finite fields serves as the foundation for this public-key cryptography technique. In order to provide the same level of security as non-ECC cryptographic (which is based on regular Galois fields), ECC requires smaller keys. Elliptic curves are useful for a variety of applications, including digital signatures, encryption, and pseudo-random generators. Fig. 1 illustrates its primary attributes. It is the technology of the near future for cryptography and offers stronger encryption, effective performance, and high scalability. Additionally, they are employed in a number of integer factorization methods with cryptographic applications, including Lenstra elliptic curve factorization. The major advantage that ECC guarantees is a smaller key size, which minimizes storage and transmission needs.

According to modern cryptography, an elliptic curve is a plane curve over a finite field (as opposed to real numbers), as seen in Fig. 1, which is made up of the points that satisfy the equation with a unique point at infinity, represented by the symbol ∞.

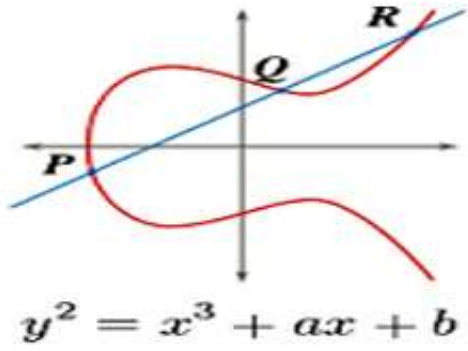

Fig.1.Elliptic Curve

With the point at infinity serving as a unique element, this collection and the group that operates on elliptic curves form an Abelian group. The group's structure is derived from the base mathematical variety's divisor group. Schemes for cryptography: A number of discrete logarithm-based protocols have been modified to work with ellipses by substituting a curve that is elliptic for the group:

- The Diffie–Hellman system serves as the basis for the elliptic curve Diffie–Hellman (ECDH) important compromise scheme.
- Alternatively referred to as the Elliptic Curve Enhanced Encryption Scheme or just the Elliptic Curve Encryption Scheme, the Elliptic Curve Integrated Encryption Scheme (ECIES).
- Electronic Signature Algorithm serves as the foundation for the Elliptic Curve Digital Signature Algorithm (ECDSA).
- Twisted Edwards curves are used by the Edwards-curve Digital Signature Algorithm (EdDSA), this is based on the Schnorr signature.
- The MQV key agreement scheme serves as the foundation for the ECMQV key agreement system.

The implicit certificate technique known as ECQV.
Whitfield Diffie and Martin Hellman created the Diffie-Hellman cryptographic key exchange technique in 1976. The "Diffie-Hellman-Merkle" approach, also referred to as the "exponential key agreement," enables participants on both ends to obtain a shared, secret key without ever exchanging it. The two parties take a similar number as well as raise it by using a different random number as a

power. After that, the outcomes are communicated to one another. The outcomes are the same for both participants when the receiving party increases the received number to the same randomised power they previously used.

The generation of domain parameters is not usually done by each participant because this involves computing the number of points on a curve which is time-consuming and troublesome to implement. As a result, several standard bodies published domain parameters of elliptic curves for several common field sizes. Such domain parameters are commonly known as "standard curves" or "named curves"; a named curve can be referenced either by name or by the unique object identifier defined in the standard documents:

- NIST, Recommended Elliptic Curves for Government Use
- SECG, SEC 2: Recommended Elliptic Curve Domain Parameters
- ECC Brainpool (RFC 5639), ECC Brainpool Standard Curves and Curve Generation

Sobol Sequence

Quasi-random low discrepancy sequences include Sobol sequences, commonly known as LPτ sequences or (t, s) sequences in base 2. These sequences reorder the location coordinates in every dimension after constructing progressively narrower uniform subdivisions of the unit interval using a base of two. This is due to the fact that sequences with minimal disparity tend to sample space "more uniformly" than random values. Such sequences may be used by algorithms with better convergence.

Related Work

The past few years have brought about an increasing focus on security issues of remote storage applications, leading to a variety of methods for designing storage verification primitives. Two primary sorts of confirmation schemes are distinguished in the literature [10]:
Probabilistic verification techniques rely on the random checking of sections of outsourced data, although deterministic evaluation schemes check the conservation of remote data in a single operation, albeit possibly more costly. The issue of guaranteeing the availability and integrity of data storage in cloud computing was covered by Wang et al. [14]. They combined data error localization and memory correctness insurance using homomorphic tokens and error correcting codes; nevertheless, their approach does not allow for an effective insert operation because of the index locations of data blocks. Because all of the current techniques use pseudorandom sequences to validate the accuracy of the data, they are unable to give users a solid sense of security. It does not compute integrity proof for all of the data. As a result, customers are not strongly guaranteed the security of their personal information by probabilistic verification systems founded on pseudorandom sequences. In order to address this issue, Syam et al. [9] suggested a homomorpic distributed verification protocol that uses Sobol Sequence—which is more homogeneous than pseudorandom sequence—instead of pseudorandom sequence in order to guarantee data storage security in cloud computing. Their plan accomplishes the availability and integrity of cloud-based data that has been outsourced, but it is comparable [14] and does not address the problem of data confidentiality.

III. SYSTEM MODEL

Cloud Data Storage Model: As displayed in the following figure, the cloud store strategy under discussion is comprised out of four main parts. (1) Cloud User: the user, who may be somebody or an organization that initially stores and retrieves data from the cloud. (2) Cloud Service Provider: the CSP is responsible for overseeing cloud servers (CSs) and offering people paid storage space on its infrastructure as a service. (3) Third Party Auditor (TPA) or Verifier: the TPA or Verifier, who checks the accuracy of data that has been transferred to the cloud on behalf of users, possesses knowledge and skills that users might not. The TPA may provide customers via a report on auditing based on the audit's findings.

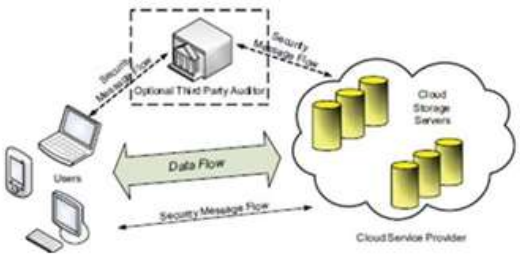

Fig.2. Cloud Data Storage Model

According to the cloud-based storage model, the user uses a company that offers cloud services to store his data in the sky. If you wishes to retrieve the information, he must submit an email to the company that provides the service and obtain the original data. If the data is encrypted, his secret key can be used to decrypt it. However, because the data is kept on unreliable storage systems, it is susceptible to attackers and could cause customers to suffer irreversible losses. Whether or not material undergoes encryption before being stored in the public internet is irrelevant, as are any preexisting relationship of confidence between the client and the server. It is necessary to reassess the current safety procedures. Therefore, it is always preferable to have a safe and effective manner for users to confirm that data is intact. The user refers this duty to a third-party auditor when they is pressed for time. On behalf of users, the auditor confirms the accuracy of the data.

The two attack types for cloud-based information storage that we are examining in this study are internal and external attacks. Internal Attacks: These are started by malevolent customers or cloud services providers (CSPs). These people are purposefully altering or erasing customer information kept within the clouds. Additionally, they have access to every one of of this data and could divulge it to third parties. Unauthorized individuals from outside the cloud launch external attacks. The external attacker can access personal data as long as it is internally consistent, which means he can remove or alter consumer data and potentially expose sensitive information. He can even take advantage of cloud servers.To achieve the following targets we have established a safe and effective storage mechanism. These objectives fall into two distinct categories: Secure Needs (Confidentiality, Authenticity) and Economic The objectives (those with computation expense and minimized transmission overhead).

Permutations and Notations

- F is a data file that will be saved on the cloud. It is divided into n equal-length blocks, m1, m2,…,mn, where n=[|m|/l].
- fkey(.)- The Sobol Random Function (SRF), defined as f: {0, 1}* ×key-{0, 1} log2 n, referenced on a key.

The definition of πkey(.)–Sobol Random Permutation (SRP) is π: {0,1}log2(1). The ×key is {0,1}log2(1).

Over the ring Zn, Elliptic Curve Cryptography: Let n be an integer and let a and b be two Zn values such that gcd(4a3 +27b2, n) = 1. An elliptic curve The set of points (x, y) ∈ Zn× Zn that satisfy the formula y2 + ax + b, along with the point at infinity known as On, is called En (a, b) over the ring Zn.

IV. PROPOSED SCHEME

We provide an effective and secure method to guarantee the safety and security of any information saved in the cloud. Our plan was created using Elliptic Curve Cryptography. [10,12] Sobol sequence design and application to arbitrarily check the integrity of stored data. Setup, Verification, and Dynamic Data Operations and Testing are the three stages of this protocol. shows the three-process model. The following is a brief presentation of how these steps were constructed: Configuration The browser prepares a document during this stage prior to a to the cloud. The following three algorithms make up the preparation step, and they are: (1) KeyGen (2) Encryption (3) Metadata Gen

Crack Operations using Dynamic Data Along with supporting block-level dynamic data operations like Block Modification (BM), Block Insertion (BI), and Branch Elimination (BD), the suggested system also preserves the same level of security [15]. The internet server carries out these operations in

fulfillment of user input in the general form (Block Op, j, m'i), where Block Op denotes the block operation (e.g., BM, BI, BD). The newly created block is denoted by m*i, and the value of parameter j identifies the specific block that needs to get modified. The user drafts a request and submits it to the website when they want to modify data in the cloud. The server carries out the specific upgrade action (modification, insert, or delete) in response to the user's request. Here, we demonstrate how our method effectively supports dynamic data operations:

Block Modification (BM): One of the most common processes in online storage of information is the modification of data.

Block Insertion (BI): In this operation, the user wishes to add a new block m* to the file F'= {m'1,..., m'n} after position j. The suggested technique can execute each block inserting operation without recalculating the the information in the metadata of all displaced blocks upon injecting the first block since the block index does not appear in the knowledge. The block insertion operation alters the logical structure of the file.

Block Deletion (BD): The opposite of an implant operations is a block deletion. All next ones advance one step after a block has been eliminated. Let's say the user wishes to remove a particular data block from the file F' at position j. To do so, they make a delete request (BD, j), submit it to the server, and ask the TPA to remove the block metadata that goes with it. The server creates an updated version pf the record F'' and delete the subsection m'j from the file in response to the user's delete request. In a similar manner, the TPA removes relevant metadata.

V. PROPOSED SCHEME IMPLEMENTATION

This part includes our protocol's evaluation of performance, results of the experiment, security analysis, and comparison to the current method.
Performance Analysis:

We evaluate our scheme's functionality in terms of computing difficulty, communication complexity, and storage capacity.

Storage Cost: In this section, we go into the storage costs that the client, TPA, and server need. User Side: The sole secrete parameter must be stored by the user. That has a storage fee of O (1).

Server Side: The entire file must be stored on the server, and doing so costs O (n) bits. TPA or Verifier: the verifier must save the public key and metadata. Since the metadata is comparatively less than the original file, O (1) is the archived cost.

Communication Cost: In this case, we take into account the cost related to communication which happens during the procedure of verification between the server and the verifier.O (1) is the request that the verifier sends to the server, and O (1) is the response that the server in turn sends to the verifier (albeit it is smaller than the original file). The whole cost associated with communication is therefore O (1).

Cost of Computation: The following is how we examine the user, verifier, and server computing costs: User: the user creates a private key and a public key with an O (1) cost during the setup phase. The user must then do integer addition, which has an O (n) cost, when trying to encrypt a file.

Finally, n-bit point multiplications with an O (1) cost are used to compute the metadata. As a result, the user's whole calculation cost is O (1).

Verifier: In order to calculate c = πkSRP (c) and Q = rP, which have an O(1) cost, the TPA or verifier must first create three random numbers, j, r, and πkSRP.
Once more, the verifier generates {aj} j= [l, c] after getting the response. The computation cost of each aj, mij is equal to the total of point multiplication of two bits. When the verifier finally calculates R', the cost of R' is O(1) for both two-point multiplications and the sum of two-bit integers plus the cost of creating random numbers. As a result, the verifier's overall computation cost is O(1).

Server Side: The server must produce n-Sobolrandom b-bit integers (ai) during the verification phase before computing (formula).

The sum of the point multiplication of two bits is what each aj mij is calculated to be. aj mij has an O(1) computation cost. The server then calculates a proof, which has an O(1) cost and is made up of point multiplications using the ProofGen method. The server's overall compute cost for producing an integrity proof (response) is O(1).

Experimental Result

Every experiment was carried out using C++ on a Windows 2007 system with a two core, 2-GHZ processor and 4GB of RAM. Instead of using RSA on 1024 bits, we use the MIRACL library version 5.4.2 in our code to accomplish superior security work on an elliptic curve with a 160-bit group order. Here, we are utilizing ECC and RSA, respectively, to evaluate the overall time for the verifier's and server's computation costs.

$$\text{RSA}-\text{ECC RSA} * 100 = \text{speedup}$$

The computing costs of our protocol for short, which involve the costs of the verifier, server, and user, are then compared to those of RSA-based remote data checking methods.

Security Analysis

We provide the formal security analysis of the suggested plan in this section. This refers to the secrecy and integrity of data stored on cloud servers. We used the Finding Order of Elliptic Curve and Elliptic Curve Discrete Logarithm Problem, or ELDL issues, as the foundation for our integrity study.

Determining the elliptic curves' order: [38,] defines the order of the elliptic curve over the ring Zn as follows: let n=pq = Nn = lcm (#Ep(a, b), #Eq(a, b)). Nn represents the curve's order, meaning that for any Pε En (a, b) and any integer k, (k Nn+1) P=P. (20)

The order of En (a, b) equals Nn if (a=0 and p≡q≡2 mod 3) or (b=0 and p≡q≡3 mod 4) are true. Nn =lcm (#Ep(a, b), #Eq(a, b)) = lcm(p+1, q+1) is the given value (21).

Factoring the appropriate number n is computationally equivalent to solving Nn.

(2) Discrete Logarithm Problem with Elliptic Curves (ECDLP). Think about the equation Q=rp, where r<n and Q, Pε En (a, b). Given Q and P, determining r is comparatively challenging. The suggested protocol is complete, according to Theorem 1.

Proof: The definition of sound and the commutative property of point multiplication in an elliptic curve [10] are used to demonstrate this theorem in this instance.

We have R' = R R′= rS mod n S = ajTij c j=1 mod n where aj = f k (j) = (ajmi ′ j c j=1 P mod Nn) mod n = ajmi ′ j c j=1 P mod n R' = r S mod n = r ((ajmi ′ j c j=1 P mod n) = r (ajmi ′ j c j=1 P mod n) = R

Equation (13) indicates if the procedure is legitimate or complete. The server's ability to securely store data is then "probabilistically" guaranteed to the verifier. The verifier essentially simply confirms that the server has the j [1, c] selective blocks, where j is selected at random.

Monte Carlo Results

Monte Carlo methods are computer algorithms that generate the numerical findings by repeatedly sampling at random. They are employed in the creation of probability distributions, numerical integration, and optimization. Zemax is a tool that can run Monte Carlo simulations and execute tolerance in different modes. Large ray datasets benefit most from sobol sampling since it speeds up convergence and cuts down on simulation time compared to truly random rays. In the source tab of the object properties, Zemax provides the choice to utilize its long-period random number generator or choose a Sobol sampling scheme.

Confidentiality: The proposed protocol is designed to prevent data leakage to malicious attackers, such as servers and TPA. The security is based on the Elliptic Curve Diffie-Hellman (ECDH) and Elliptic

Curve Discrete Logarithm (ECDL) problems. The protocol's confidentiality is proven through various attacks, including eavesdropping on communication links, obtaining a secret parameter from a user randomly chosen secrete key, and accessing data content from metadata using a TPA's secrete parameter. These challenges ensure the protocol's confidentiality against data leakage.

Comparison with Existing Schemes

Table 1: Comparisons between Proposed Protocol and selective Existing Protocols

	Q.wang [14]	Hao [17]	Syam[9]	Proposed protocol
Type of Guaranty	Prob	Deter	Prob	Prob
Integrity	Partial	Yes	Yes	Yes
Confidentiality	No	Partial	No	Yes
Public Verifiability	Yes	Yes	No	Yes
Data Dynamics	Yes	Yes	Partial	O(1)
Communication complexity	O(logn)	O(1)	O(1)	O(1)
Server Computation	O(logn)	O(n)	O(clogn)	O(1)
Verifier computation	O(logn)	O(n)	O(clogn)	O(1)
Probability Detection	$O(N^{-1/2})$	$O(N^{-1/2})$	$O(N^{-1})$	$O(N^{-1})$

Prob: Probabilistic Determination: We suggested a verification method based on ECC. The idea behind ECC is that, in contrast to RSA, it seems to provide the same level of protection with a much smaller key size, which lowers the computational overhead. While the suggested methodology uses the Sobol sequence to confirm the data's integrity, the pseudorandom sequence is not uniform (uncorrelated random values) and will take longer to identify data corruption. Since a sobol sequence covers all of the data in the file more uniformly than a pseudorandom sequence, our system should be able to detect all data corruptions with fewer blocks. Lastly, since we encrypt data before putting it in the cloud, the suggested protocol is secure against unwanted data leaks. Table 1 provides a summary of the contrast between the chosen current protocols as well as the suggested protocols

VI. CONCLUSION

One of the public key cryptography techniques is elliptic curve cryptography (ECC). Even while RSA is currently the most widely used cryptosystem scheme for web security, the proliferation of smaller devices and rising security requirements could cause ECC to surpass it. Despite the reality that there have been multiple attempts to create an ideal setting for cloud-based activities, Elliptic Curve Cryptography (ECC) offers solutions for a secure cloud environment with better battery and processing power use. It is appealing for mobile applications because of this. For the creation and implementation of secure cloud applications, ECC offered a reliable and secure methodology. The issue of cloud data storage integrity and confidentiality has been examined in this research and suggested a safe and effective methodology utilizing the Sobol sequence and ECC. Thin users with low processing power and limited resources are best suited for the suggested approach. It meets all cloud storage of data security and performance criteria. Additionally, our approach facilitates public verifiability, which allows TPA to confirm the validity of data without having to retrieve the original data through the server and likely identify data corruption. Additionally, our system allows for user-performed dynamic data operations on cloud-stored data while preserving the same level of security.

Through security research, we showed that the suggested technique is secure in terms of privacy and integrity.

The effectiveness of the suggested plan was demonstrated by performance analysis and experimental findings. We have also demonstrated that the suggested approach is more effective and safe than earlier protocols. If the experimental instruments are easily accessible, the study could possibly be extended for contrasting ECC with quantum cryptography.

REFERENCES

1. Amazon.com, "Amazon Web Services (AWS)," Online at http://aws. amazon.com, 2008.
2. N. Gohring, "Amazon's S3 down for several hours" Online at http://www.pcworld.com/businesscenter/article/142549/amazons s3 down for several hours.html, 2008.

3. Apple "ICloud" Online at http://www.apple.com/icloud/what- is.html 2010 .
4. T Mather, S Kumaraswamy, and S Latif "Cloud Security and Privacy", O'REILLY Publication, first edition, sep- 2009.
5. 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. 10, pp. 24-31, 2010.
6. Y. Deswarte, J.-J. Quisquater, and A. Saidane. "Remote integrity checking". In Proc. of Conference on Integrity and Internal Control in Information Systems (IICIS'03), Nov03, lausanne, Switzerland, 2003.
7. G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of SecureComm '08, pp. 1–10, Istanbul, Turkey, 2008.
8. M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop on Hot Topics in Operating Systems (HOTOS '07), pp. 1–6, CA, USA, 2007.
9. P. Syam Kumar, R. Subramanian, "Homomorpic Distributed Verification Ptorotocol for Ensuring Data Storage in Cloud Computing". International Journal of Information, VOL. 14, NO.10, OCT-11, pp.3465-3476, 2011.
10. N. Oualha, M. Onen, Y. Roudier, "A Security Protocol for Self Organizing Data Storage". Tech. Rep. EURECOM+2399, Institut Eurecom, France, 2008.
11. 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.
12. K. Koyama, U. Maurer, T. Okamoto, and S. Vanstone, "New Public-Key Schemes Based on Elliptic Curves over the Ring Zn", Advances in Cryptology - CRYPTO '91, Lecture Notes in Computer Science, Springer-Verlag, vol. 576, Aug 91 pp. 252 266, 1991.
13. Brately P and Fox B L "Algorithm 659: Implementing Sobol's Quasi-random Sequence Generator" ACM Trans. Math. Software 14 (1), pp. 88–100, 1988.
14. 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.
15. Z. Hao, S. Zhong, and N. Yu, "A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability", Accepted for publication in future issue of IEEE Trans. Knowledge and Data Engineering, DOI: 10.1109/TKDE.2011.62
16. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson,and D. Song, "Remote Data Checking using Provable Data Possession," ACM Trans. ACM Transactions on Information and System Security, Vol. 14, No. 1, Article 12, may, pp. 12.1–12.34, 2011. Z. Hao and N. Yu, "A multiple-replica remote data possession checking protocol with public verifiability," in Second International Symposium on Data, Privacy, and E- Commerce ,.Buffalo, Niagara Falls, 2010. [18] William Stallings, Cryptography and Network Security Principles and Practice Sixth Edition, Pearsons, ISBN10:0-13-335469-5.