IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Remote Integrity Auditing Scheme (RIAS) based on Luhn's Approach

V. Hema[1*], M. Ganaga Durga[2]

[1] Bharathiar University, Coimbatore, Dept. of Computer, Agurchand Manmull Jain College, Chennai, India
[2]Bharathiar University, Coimbatore, Dept. of Computer Applications, Sri Meenakshi Govt. Arts College for Women, Madurai

Corresponding Author: vhema23@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i5.16021607 | Available online at: www.ijcseonline.org

Accepted: 19/May/2019, Published: 31/May/2019

Abstract— Cloud technology has gained fabulous popularity in recent years. By outsourcing the confidential resources to the public providers and paying for the provision used, the users can bliss upon the advantages of this new paradigm. However, the archive which backups the user's sensitive data may not be fully trustworthy and introduces new challenges from the perspectives of data correctness and security. The users may also concern much about data intactness. Bountiful attempts have been espoused and many technological implementations have been established to remove insecurities. This paper aims to enhance the importance of the data integrity scheme and proposes a remote data possession checking based on the Luhn's approach. The main idea is to design the tags computed from cipher blocks can be used to check the integrity of the resources in deposited in the archive. The security and performance analysis illustrates the computational, storage and communication efficiency of this scheme. Finally, it performs unbounded data possession checking which provides confidentiality of archived sensitive data.

Keywords— Cloud storage, remote data possession checking, provable data possession, proof of retrievability.

I. INTRODUCTION

Cloud computing is pondered as a major Information Technology shift and achieves the dream of getting the computing resources and other applications in pay per use scheme. The increased number of users can deploy their resources in the cloud storage without any local backups [1]. It has the copious advantages such as scalability, ease of use, plug – and – play service, abridged infrastructure planning, and so on. However, the emerging use of cloud storage has led to the problem of verifying that the storage server indeed stored the data [2]. The main need of the remote data possession scheme is to check the cloud storage regularly, efficiently, and securely the outsourced data without retrieving it. The storage server may not be fully trustworthy in terms of both security and reliability. Auditing scheme can be of two types, namely provable data possession (PDP) and proof of retrievability (POR). The difference between both the schemes is that POR audits the possession and recover the data in case of a failure. Nowadays, a PDP can be converted to a POR by adding error-correcting schemes.

An auditing scheme designed must satisfy the following requirements to be of practical use in cloud backup. They include acceptable computation and communication overhead, low storage cost and no need of the original file for verification purposes. This paper proposes a new auditing scheme in combination with novel hill cipher techniques and Luhn's algorithm. The user's sensitive data is first encrypted using the novel hill cipher [3] and computes the meta tags from cipher text blocks using Luhn algorithm, and sends the encrypted blocks to the cloud server.

When the file owner wants to verify, challenge the storage server with the block number and random number with the help of TPA. The server performs a simple calculation and returns the block number, newly generated random number from the existing one, data tags computed for the specified cipher text blocks to the client. Then, the TPA can valid the response against the storage details to know the file's integrity.

The proposed scheme which almost fulfils all the requirements noted above. First, it is proficient in terms of computation and communication overhead between the client and server. Second, it allows verification without the server cheating. . Third, the data transmission for challenge and response sequence was very small. And owner needs to store a small amount of metadata with corresponding block numbers. Finally, it provides data possession checking at the same time it provides confidentiality of data. The efficiency of the scheme makes it ideally suited for use in cloud store.

The complete organization of the article is arranged as follows: Section I contains the introduction to the cloud and auditing scheme. Section II contains the related work. Section III presents preliminary notations, system and threat model. Section IV contains the detailed remote auditing scheme using luhn and LCG. Section V presents the results and discussion of the proposed scheme. Section VI concludes research work with future directions.

II. RELATED WORK

Auditing is a periodic event to evaluate the quality of confidential data for a specific purpose like to assess the confidentiality and integrity. It could be a principal source for guarding sensitive data resources against potential risk and loss. Various remote integrity checking protocols have been proposed by various researchers in recent years. This entire proposed scheme can be categorized into two classes. Data integrity auditing on all the data blocks and on selected data blocks in storage.

The first type of system provide deterministic guarantee of data by checking all the data blocks. The other type uses the probabilistic auditing techniques by using random blocks to ensure the integrity of the outsourced file the protocols of these two classes are as follows.

Deswarte and Quisquater [4] use hash function based on RSA to hash the file at every challenge. To access the file segment, the server has to compute the exponentiation function over the outsourced file F'. Ateniese et al. has designed a PDP system using the homomorphic verifiable tags and uses public key cryptosystem [5][6].

In [7], using symmetric key cryptography, token based secure PDP scheme was proposed. These tokens can be stored in the client side as plain form and encrypted form in server. Xiao et al. proposes a system which was based on symmetric key system. Their contribution which includes a challenge-update mechanism for audit the dynamic content stored in the storage server.

In [8], Secure multiple-replica PDP system which allows, the user to store t mock-ups of a file and audit those files held on the server which is safe or not. The system proposed by Filho and Baretto [10] prevents the data sleaze in transfer.

Juels and Kaliski introduced [11] the auditing technique which use disguised blocks called sentinels, concealed among the file blocks that the server cannot differentiate from cipher blocks. In [12], Sebe et al. proposed the RDPC technique that allows an unlimited number of auditing and the maximum running time can be selected at the setup time and traded off against storage at the auditor.

In [13], Schwarz and Miller designed the XOR-based technique which is used to yield the n shares of a file that are outsourced at multiple locations. Using algebraic signatures, they provide a very brilliant way to audit whether every storage has stored intactly each other's resources.

For security reasons, most of the proposed schemes are based on public-key cryptosystem. They are always expensive when the total data handled is large. The computational and communication overheads are also high. The auditing technique [5], [7], [12], [14], [15] generates probabilistic proofs of possession by audits the random blocks from the server, which significantly reduces the transaction costs. The challenge/response protocol transmits a low, constant amount of data which minimize both the computation and communication costs.

Different from the above schemes, we propose a Luhn's algorithm based RDPC scheme that is probabilistically secure but apt for auditing the cloud archive. It also detects the single error in the encrypted data stored in the storage server.

This is simple and efficient scheme to verify the confidentiality and integrity of the user's data. An experiment shows the efficiency and performance which enables it preferably suited for use in cloud archive.

III. PROBLEM STATEMENT

In this section, we first delineate the system model and threat model. Then, we provide definitions of the notations that will be used in our scheme.

A. System Model

Cloud data storage model consists of three entities such as the cloud, third party auditor (TPA) and the cloud users as depicted in fig. 1. The cloud entity is managed by the Cloud Service Provider (CSP), which offers the data outsourcing and significant computation resources to the users. TPA is an authentic entity in a cloud environment required for launching secure interactions between cloud users and server.

TPA can greatly diminish the burden of the user's auditing task. The cloud user hosts their sensitive data to the cloud archive. The cloud user has only the limited resources, so they do not keep their local copy of data. The auditor on behalf of owner, checks the integrity of the outsourced data. The outsourced data file is split into blocks and encrypted using the novel Hill cipher technique [3], [16]. During the process, the users can update or delete the blocks stored in the backup. The auditor with the help of owner, update their table to reflect the change in the blocks in the outsourced server.

Table (1) shows the comparison of different schemes

	Type	Data	Set up	Enc	PA	BV	UQ	DR	PP	DD	IMPLEMENTATION	REMARKS
[6]	Pb	Static	C	SK	N	Y	N	N	N	Y	Homomorphic tags	CSP cheat possible – lack of randomness
[17]	Pb	Static	C	AK	Y	Y	Y	N	N	N	RSA Homomorphic tags	Not fit for multi-cloud
[10]	Pb	Static	C	FEC	N	Y	Y	N	N	N	Pseudo Random function	Not fit for multi-cloud
[8]	Pb	Static	C	AK	Y	Y	Y	Y	N	N	BLS Signature & Pseudo Random function	Unable to prevent data leakage by verification protocol
[18]	Pb	Static	C	-	N	Y	Y	Y	N	Y	ECC	Rank based authenticated skip list
[19]	Pb	Static	MC	SK	Y	Y	Y	Y	N	Y	Signature based, Merkle Hash Tree & BLS	Privacy Leakage possible
[21]	Pb	Dyna	C	AK	Y	Y	Y	Y	Y	Y	Homomorphic Authenticator, Random mask technique	Overhead in cases of failed auditing response
[15]	Pb	Static	C	SK	Y	Y	Y	Y	N	N	Merkle hash tree	drawback of this scheme lies in its probabilistic security.
[20]	Pb	Dyna	C	AK	N	Y	Y	Y	N	Y	Range based 2-3 trees & ECC	Low performance because of Error correcting codes
OURS	Pb	Static & partial dyna	C	SK	Y	Y	Y	P	Y	Y	RCLT based. Luhn's approach Avoid CSP cheat using LCG	Suitable to find the single or transposition errors only. Unable to find the Twin errors
BV(Block less verification), UQ(Unbounded Querying),DR(Data Recovery), PP(Privacy Protection), DD(Dynamic Data Operation),PA(Public Auditing),AK(Asymmetric Key),SK (Symmetric Key),Pb(probabilistic), C(Cloud), MC(Multi-Cloud)												

B. Threat Model

In general, the integrity of the stored confidential data is threatened by two kinds of threats. They are internal and external threats. In internal threats, the provider may damage the archived data due to the system failure or human errors and may try to hide the data loss incidents for reputation. In external threats, corruption of outsourced data and prevents the users from accessing their data correctly may occur. These threats can be solved by providing proper access control mechanisms. The following fig (1) shows the threat model.

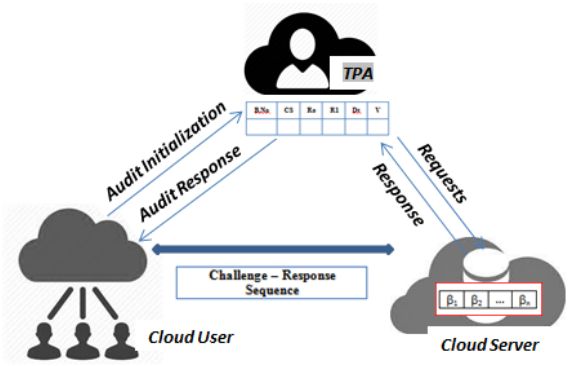


Fig (1) Threat Model with 3 components

In this paper, we mainly focus on the internal threats [23] threatening the user to use the data outsourcing facility effectively. Since, the data stored in the cloud's archive is untrusted; we need a method to audit the integrity of the data to guarantee that the outsourced sensitive data is intact. The proposed integrity auditing scheme is devised to protect the encrypted data stored in the storage. The trusted third party auditor can utilize this scheme to audit the storage and sent the status to the cloud user. This scheme also supports little error detection scheme.

C. Preliminaries and Definitions

In this section, we present some definitions used in the proposed scheme.

1. An improved Novel Hill Cipher using RCLT

An Improved Novel Hill cipher (INHC) [1], [5], [8] is a symmetric key technique capable of encrypting the confidential data by the user before outsourced to the cloud server and decrypt the disguised data by the receiver after transmission through the cloud archive. INHC was designed to reduce the chosen plaintext and cipher text attack in the unsecure channel.

The file f is divided into β blocks (i.e.) $f = \{fb_1, fb_2 \ldots fb_\beta\}$ where $1 \le i \le \beta$. The enciphering and deciphering the file blocks is done by

$$\gamma : (M + \kappa) \bmod 87 \rightarrow C \quad (1)$$

$$\psi : (C - \kappa) \bmod 87 \rightarrow M \quad (2)$$

Where κ is the key matrix generated using

$$\kappa_{11} = (\mathcal{H} * \varepsilon(1,1)) \bmod x$$

$$\kappa_{12} = (\mathcal{H} * \varepsilon(1,2)) \bmod x$$

$$\kappa_{13} = (\mathcal{H} * \varepsilon(1,3)) \bmod x \ldots. \quad (3)$$

So γ' is the final cipher block stored in the archive after performing the in-place rotation by 90^o clock wise?

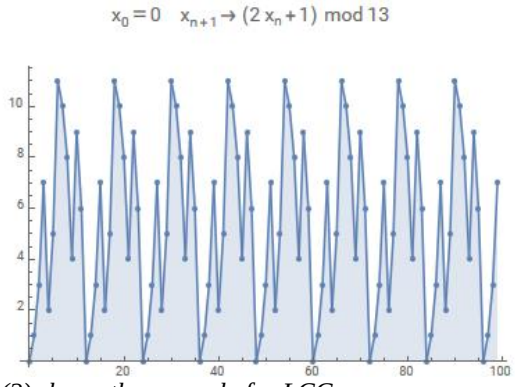
2. *Luhn's Algorithm*

The Luhn algorithm was developed by Hans Peter Luhn in 1954. It computes simple checksum used to authenticate numbers such as credit card numbers, PAN numbers and IMEI numbers. It also used with cryptographic technique to safeguard against accidental single bit error as well as transpositions of adjacent digits. The formula to calculate the check digit is $(\sum_{i=0}^{m} 2n_{2i} + \sum_{i=0}^{m} n_{2i+1}) \, mod \, 10$ where n $\in$ Z_{10} and i, m$\in$Z but if ∋ q ≥ 10, sum the digits of the products together to get the single digit.

If q=a*b then calculate a+b=e before the calculation of the check digits mod 10. If $(\sum_{i=0}^{m} 2n_{2i} + \sum_{i=0}^{m} n_{2i+1}) \, mod \, 10$ is equal to 0 then the number is valid or otherwise it is considered as invalid. Let $d_1 d_2 \ldots . d_n$ is the digits of the number. The number is represented as $d_1 d_2 \ldots . d_n x$. Every even positioned digits is multiplied by 2 from the rightmost direction to obtain $q_1 q_2 \ldots . q_n x$. If ∋ *qi* ≥ 10, then the check digit is $x = 10 - (\sum_{i=1}^{n} qi) \, mod \, 10$. If the number is valid, if the check digit obtained is 0 otherwise the number is invalid.

3. *Linear Congruential Generator*

A **linear congruential generator** (**LCG**) is one of the oldest and best-known pseudo random number generator algorithms. This method produces a series of integers such as $r_1, r_2, \ldots . r_n$ based on the following mathematical formula $X_{i+1} = (aX_i + c) mod \, m \; \forall \, i = 0,1,2..$ where X_0 is the seed, a is constant multiplier, c is the increment and m is the modulus. The following fig (1) shows the example for the LCG.



Fig(2) shows the example for LCG

IV. DATA POSSESSION CHECKING BASED ON LUHN

Data outsourcing and backup has led to the difficulty of inspecting the data in the archive and this must often be done in the absence of the knowledge of what the data should be. In this paper, we describe a system which uses the cryptographic technique along with the Luhn's Algorithm and LCG to safeguard and show the data possession in the cloud storage. In this section, we first state some definition applied in the developed of the auditing scheme. Then, we describe the algorithm and detailed phases of the auditing scheme for cloud storage.

A. Scheme Description

A remote data possession auditing scheme generally has one or more phases. They include the following main five stages namely, prelim, TagBlock, Challenge, ProofGen and Proofcheck phases. In prelim phase, it generates some initialization parameters required for auditing process, challenge and response activities. The random numbers are generated using LCG used to avoid server fraud.

In TagBlock stage, it selects *c* file blocks randomly to compute the checksum using the Luhn's algorithm and seed using LCG to check the integrity of the file *f* 'stored in the archive. In pre-processing stage, it computes *t tags* along with seed and stored in the table used by the TPA for auditing. In the Challenge stage, TPA generates the query string (b_i,r_i) which contains the block number to be checked and a random number, which is sent to the server in response.

In the ProofGen stage, the storage server computes the locations of the requested block b_i, and computes the checksum (τ) of the cipher block, generate the random number based on the seed *r* and then returns the $\langle b_i, \tau, rs \rangle$ to the TPA . In the ProofCheck stage, the TPA verifies the response against information stored in the table and sends the status to the user. If yes, it indicates that the file is not destroyed. The details of the scheme is depicted in the fig (2)

Prelim: *Owner initializes the following parameters (a, i, c and m) required for LCG to generate the seed and checksum generation.*

TagBlock :

For $0 < i \leq m$

$X_{i+1} = (aX_i + c) mod \, m \; \forall \, i = 0,1,2..$

$r_i = X_{i+1}$

Next i

For $b_{i \, cipher}$ *blocks* $0 < i \leq n$

Computes $\tau = (\sum_{i=0}^{m} 2n_{2i} + \sum_{i=0}^{m} n_{2i+1}) \, mod \, 10$

Owner sends (b, r, τ*) to the TPA*

Challenge: *for the* i^{th} *challenge, user generate the query string* (b_i,r_i) *send to the server*

ProofGen :

$d_1 d_2 \ldots . d_n$ *is the digits or bits of the block.*

It performs the following:

Digits

Test num

*1) n=n*2 for every 2nd n from right*

*2) if n*2>9 then add digits*

*3) else n or n*2*

4) cs=sum of the digits

Server sends (cs, b, rs) to the TPA

ProofCheck :

TPA checks the cs and rs against the value stored in the table along with seed generated using LCG

Checks if c =τ then send the status "Valid" to the owner.

The drawback of the existing scheme includes server cheating the client challenge, extra space required to store data along with tag and computation overhead. To overcome these drawbacks, we propose a new protocol based on number theory concepts which have the amalgamation of cryptography and auditing technique. The table (1) above shows the comparison of existing scheme with our scheme. In our scheme, the storage cost of client, cipher text attack, plain text attack and server fraud is greatly reduced. In order to ensure the integrity, TPA uses the challenge-response technique and it chooses n file blocks randomly from b_n' blocks each time. This also reduces the workload on the server, while still achieving detection of server misbehaviour with high probability. It uses the stored table of information to check the integrity of the outsourced file. This scheme also handles single and transposition errors in the stored block.

4.1 Data Dynamics

Data dynamics means that the users can accomplish data insertion, updating and deletion of sensitive data at any time while the auditor can still performing the auditing tasks. To support data dynamics, the auditor and the cloud need to maintain an index hash table to record the indexes of the blocks, version of the block, date last modified and their signature generated for the encrypted blocks.

B. Updation

Data update is one of the most frequently performed operations. In their local storage, client updates the content directly on the data file. In the cloud, the data files are stored in the cloud, directly modifying is not possible. Therefore, the data update in the cloud storage scenario actually means replacing the old file block in the cloud with a new block modified by the user. Assume the user wants to update the i-th file block β_i of the outsourced data with the modified file block; the data updating process are as follows:

- Check and update the *cs* and random value *r* for the modified block. Then user computes the τ'_i for the modified block.
- The user sends the updated information to the TPA. Then also append the updated information in the index table.
- The cloud replaces the old file block of index i with the new file block on receiving the data updating message. Also change the last modified date and increment the version number as v=v+1.

C. Deletion

To delete the desired block, perform the following operations:

- To delete the ith block the outsourced file f', generate and send the delete request to the TPA and server.
- The server and TPA deleted the block with the index of i from the table. After performing the deletion operation, the user could send the audit request to the TPA to check the integrity of the outsourced file f'.

D. Insertion

To insert a new block, first encrypt it using NHC technique. The encrypted block say f'_{new} is inserted at the specified position of index i with the newly generated tag for that block. To the insert the new block, perform the following steps:

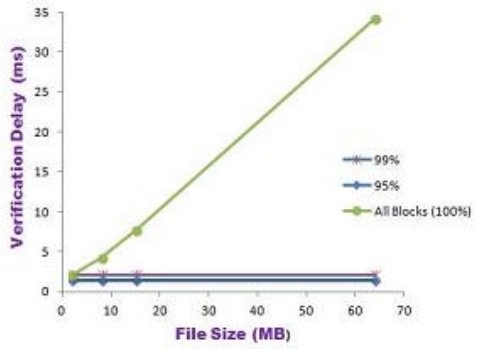
- The owner calculates the checksum for the new block say, τ_{new}
- The owner forwards the block appending message {f'_{new} , b_i, date, v} to the TPA. And also send the encrypted new block to the archive.
- The server insert the new encrypted block at the specified position and modified the index table for further auditing process.

V. RESULTS AND DISCUSSION

The proposed cryptosystem utilizes the random lookup table, random seed based key generation and number theory concept is used to heal the security downside in the existing algorithms and it can be implemented to provide the data confidentiality. We discuss the performance of the scheme in the point of auditing (i.e.) how frequently and efficiently the client audit the storage server. In our scheme, only constant amount of data is required for challenge the server. The total number of verification and the blocks required for each challenge can be made lithe according to user's need.

In public auditing for collaborative scheme, the TPA performs efficient checking and also preserves confidentiality of the information outsourced to the cloud. Proposed algorithm which is designed to improve the storage, computation and communication overheads of the existing algorithms. This scheme supports accelerating encryption and also provides the stateless verification, which incurs less overhead to accomplish challenging response activities. Public verification using TPA for reconstructing code based distributed repository achieves protection of outsourced data against exploitation and external attacks to the repository. This proposed mechanism relieves clients from online burden and supports partial error detection scheme.

The fig (3) shows the verification overhead for detecting missing block on the server. The confidence the verification overhead of our scheme is less than 2.20 ms for any file.



Fig(3) shows the verification overhead in accorandance with file size

CONCLUSION AND FUTURE SCOPE

In this paper, we propose an efficient approach for data integrity auditing with TPA. This scheme uses Luhn's algorithm and LCG for tag generation and for data integrity verification. Benefits of this scheme greatly reduce the computational and client overhead. Efficiency of this approach makes it utterly suited for cloud archive. By using randomness in client challenge, this scheme greatly reduce the server cheat.

Fortuitously, when the server erases a fraction of encrypted file, the owner with the help of TPA can detect server misconduct with great probability by challenge a constant amount of file blocks. This scheme supports error detection and data dynamic in partial manner with some high overhead. In future, we try to design scheme that supports full-fledged dynamic data operations.

REFERENCES

[1] Cachin, C., Keidar, I., Shraer, A, *"Trusting the Cloud",* ACM SIGACT News 40(2), 81-86 (2009).

[2] K Ren, C Wang, Q Wang, *"Security Challenges for the Public Cloud"*, IEEE Internet Computing,2012,16(1):69-73.

[3] Hema. V, and Dr M. Ganaga Durga. "*An Improved Novel Hill Cipher Using RCLT* ", International Journal of Engineering & Technology,vol.7,no.3.3,2018,p.209.

[4] Deswarte Y, Quisquater J J. *"Remote integrity checking"*. In: Proc. of IICIS '03, 2003, pp.1–11.

[5] Ateniese G, Kamara S, Katz J. "*Proofs of storage from homomorphic identification protocols"*. In: Proc. of ASIACRYPT '09, 2009, pp. 319–333.

[6] Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., & Song, D. (2007). *"Provable data possession at untrusted stores",* Proceedings of the 14th ACM Conference on Computer and Communications Security - CCS 07. doi:10.1145/1315245.1315318

[7] Ateniese G, Pietro R D, Mancini L V, Tsudik G. *"Scalable and efficient provable data possession",* In: Proc. of SecureComm '08, 2008, pp.1–10.

[8] Curtmola R, Khan O, Burns R, Ateniese G. *"MR-PDP: multiple-replica provable data possession*", In: Proc. of ICDCS '08, 2008, pp.411–420.

[9] Xiao D, Shu J, Chen K, Zheng W. *"A practical data possession checking scheme for networked archival storage"*, Journal of Computer Research and Development 2009; 46(10):1660–1668.

[10] Filho D L G, Baretto P S L M. *"Demonstrating data possession and uncheatable data transfer"*, IACR ePrint archive,2006. Report 2006/150, http://eprint.iacr.org/2006/150.

[11] Juels A, Kaliski B S. PORs: *"Proofs of retrievability for large files"*, In: Proc. of ACM-CCS '07, 2007, pp.584–597.

[12] Sebe F, Domingo J F, Martinez A B, Deswarte Y, Quisquater J. *"Efficient remote data possession checking in critical information infrastructures"*, IEEE Trans. on Knowl. and Data Eng., 2007, pp.1034–1038.S.

[13] Schwarz T J E, Miller E L. *"Store, forget, and check: using algebraic signatures to check remotely administered storage",* In: Proc. of ICDCS '06, 2006, pp.12.

[14] Erway, Chris, et al. *"Dynamic Provable Data Possession." Proceedings of the 16th ACM Conference on Computer and Communications Security - CCS 09*, 2009, doi:10.1145/1653662.1653688.

[15] Chen, Lanxiang. *"Using Algebraic Signatures to Check Data Possession in Cloud Storage." Future Generation Computer Systems*, vol. 29, no. 7, 2013, pp. 1709–1715., doi:10.1016/j.future.2012.01.004.

[16] Hema. V, and Dr M. Ganaga Durga. *"A Novel Hill Cipher (NHC) based on Galois Field "* ,International Journal of Pure and Applied MathematicsVolume 118 No. 7 2018, 641-645

[17] S. Nepal, S. Chen, J. Yao and D. Thilakanathan, *"DIaaS: Data Integrity as a Service in the Cloud," 2011 IEEE 4th International Conference on Cloud Computing*, Washington, DC,2011,pp.308-315. doi: 10.1109/CLOUD.2011.35

[18] Dodis, Yevgeniy, et al. *"Proofs of Retrievability via Hardness Amplification."* Theory of Cryptography Lecture Notes in Computer Science, 2009, pp. 109–127., doi:10.1007/978-3-642-00457-5_8.

[19] Bowers, Kevin D., et al. "*Hail"*, Proceedings of the 16th ACM Conference on Computer and Communications Security - *CCS 09*, 2009, doi:10.1145/1653662.1653686.

[20] Popa, Raluca Ada, et al. *"Enabling Security in Cloud Storage SLAs with CloudProof."*, *USENIX Annual Technical Conference*. Vol. 242. 2011.

[21] Wang, Qian, et al. *"Enabling public verifiability and data dynamics for storage security in cloud computing."*, European symposium on research in computer security. Springer, Berlin, Heidelberg, 2009.

[22] Nitesh Jain, Pradeep Sharma, "*A Security Key Management Model for Cloud Environment*", International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.1, pp.45-48, 2017.

[23] M. Arora, S. Sharma, "*Synthesis of Cryptography and Security Attacks*", International Journal of Scientific Research in Network Security and Communication, Vol.5, Issue.5, pp.1-5, 2017.

