International Journal of Engineering & Technology, 7 (2.7) (2018) 565-569

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

A Comprehensive Survey on Public Auditing for Secure Cloud Storage

Smita Chaudhari[1]*, Siva Kumar Pathuri[2]

[1]Research Scholar, K L Deemed to be university, Vijaywada, India
[2]Associate Professor, K L Deemed to be university, Vijaywada, India
**Corresponding author E-mail: smita.m.c@gmail.com*

Abstract

Cloud computing is the most popular paradigms used today by Industries & individual users to store data. This outsourcing releases user from capital as well as maintenance cost to own their data. But it brings new security challenges such as data integrity & privacy since user has no power of control on his own data. Advances in encryption & authentication techniques has improved security of data at cloud server(CS) but still it is not providing any certificate or assurance about cloud data to user. Most of the times users are not aware of the different controls employed by CS to protect integrity of data. Due to this lack of transparency in system, the user may lose trust on CS. Hence it is a need of user to check integrity of his data at regular intervals. Most of the researchers have given solutions to this problem with the help of cryptography techniques. External parties such as Third-Party Auditors (TPA) are performing audit to verify this remote data on behalf of user. This paper surveys different cryptography mechanisms proposed by different researchers to check integrity of remote data. Finally, we address future research challenges that need to be resolved by researchers to make system more transparent.

Keywords: *Aggregate Signature; Cloud Computing; Data dynamics; Homomorphic Authenticators; Public Auditing; Ring Signature; Shared data*

1. Introduction

Mell and Grance [12] defined the cloud computing as “A model of 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 minimum management effort or service provider interaction”. One most important benefit of this model is that outsourced data is centralized to the CS. With respect to the user, either individual or IT enterprise, storing data remotely brings profit such as relief from the burden of storage management at local site and escaping from capital expenditure on infrastructure and personal maintenance [2].

Although this model proved to be a gifted service paradigm for the Internet, it has come up with many new complicated design issues which may concern the security and performance of the whole system. The major security concern is of data integrity proof at unreliable CS. For example, the CS practices involuntary failures some time or may choose to cover the data errors from the cloud user to maintain their reputation. Rarely accessed data files of ordinary users might be deleted intentionally by CS to create space for other user. Consider that the large size of electronic data is outsourced by cloud user on the cloud data storage. The cloud user may need to ensure the integrity of his data periodically which may create overhead on him of downloading that file from CS because of restricted resource capability [1].

As the file owners are not having physical possession of the data outsourced to CS, conventional cryptographic techniques cannot be used directly to make sure the accuracy of data. Simply downloading the full data to check for integrity is costly and not efficient because of higher expenses of I/O and communication cost. With reference to user’s view, the data owner must be capable to use cloud data storage as if its own home drive, without caring regarding the necessity to verify the correctness of data stored in the cloud. To verify data integrity, to save the owner’s computational assets with an online load of communicational resources, data owners generally delegate these responsibilities to TPA. The TPA, in support of the data owner checks the integrity of stored data periodically. The TPA, who is expert and having capability than ordinary users, generally verify the integrity of the outsourced data in the cloud from time to time. This delegation offers a much simpler and inexpensive approach for the file owners to make certain that the integrity of their outsourced data is not compromised.

With respect to the verification activity, the auditing scheme is categorized as: private auditability and public auditability. Private audit generally refers the work done by an organization’s own employees. It concentrates primarily on optimization and risk management. But with public auditing, anybody can claim the cloud server for confirming the integrity of cloud data storage.

2. Cloud Storage Architecture

A cloud data storage auditing system consists of three distinct components, as shown in Figure1: Cloud User, CS and TPA. The cloud user is the file owner who outsources large quantity of data files on the CS. The CS is controlled by the cloud service provider (CSP) consisting of vast storage space and computation assets to offer data storage services to the user. The TPA is an expert and having potential that ordinary cloud users are not having. TPA is a reliable component who checks the correctness of cloud storage for cloud user upon his request or periodically. Cloud users rely on the CS for storage functions and preservation of outsourced data. They may also access and modify their stored data interacting with the CS any time.

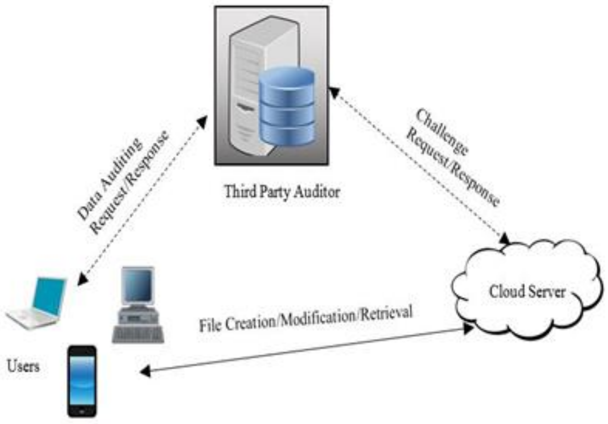
Third Party Auditor
Data Auditing Request/Response
Challenge Request/Response
Cloud Server
File Creation/Modification/Retrieval
Users

Fig.1: Architecture of Cloud Stoarge Auditing System

As users are not having physical ownership of their outsourced data, it is very essential for owners to make sure that their data are being properly stored and preserved. Periodic checking of outsourced data may create burden on cloud user in terms of computational and network resources. To save these resources, data owners may invoke the TPA to make sure the storage correctness of their farm out data but TPA may infer some information while auditing. So it is necessary to maintain privacy of their data from TPA.

CS is considered as a semi-trusted entity, since usually it behaves properly according to the protocol. Integrity of the owner's data may be affected by both internal and external attacks on CS e.g. hardware failures, software errors, bugs in the network path, economically enforced hackers, malicious or unintentional management errors, etc. However, CS may intentionally create harm to some ordinary user's data, which is not used or referenced frequently by deleting it or not keeping. It can reclaim that space to other user for its own benefit. To maintain the reputation, CS might not inform these data corruption events to users. By including trusted third parties, users can keep trust on a cloud server. TPA is one of the reliable and independent entities in the model. However, it may create harm to the privacy of the data owner if the TPA could gain knowledge of the outsourced data during the audit.

3. Goals of Public Auditing

1. Public Auditing: - TPA can check the accuracy of cloud data without downloading the full data file and minimizing online burden on data owner.
2. Privacy Preserving: - During the audit process, user's data content or identity must not be revealed to TPA.
3. Data Dynamics Support: - Data owners most of the time update their data for different application tasks. Auditing procedures should support such type of dynamic updates on data.
4. Batch Auditing: - TPA must have capability to tackle several auditing requests from a different number of users concurrently.
5. Lightweight: - TPA as well as Data Owner must be capable to carry out auditing work with smallest communication and resource cost.
6. Accountability: - Up till now we have considered cloud server as an un-trusted entity but sometime cloud user or TPA may be untrustworthy. Accountability helps to address such issues.

4. Approaches for Auditing

To perform public auditing on cloud storage, the system has to work in two stages: Setup and Audit. In the setup phase, the metadata is generated for the data file to be outsourced. The public and secret parameters are initialized by data owner to pre-process the file. Verification metadata is produced by pre-processing the data file F. The owner stores the data file F and the pre-processed metadata at the CS. Data file F may be modified by including additional information to be stored on CS while pre-processing. In audit phase, the TPA issue a challenge to a CS to confirm the correctness of data file F. CS then derive a response based on metadata and send it to TPA [2]. Many researchers have used different cryptographic techniques to audit the cloud storage. Table 1 lists different schemes used by different researchers to satisfy requirements while auditing.

4.1. Traditional Methods

A well-known method to check data integrity is Message Authentication Codes (MAC) [13] in cryptography. Data owner calculates the MAC for the file to be outsourced on cloud storage. Whenever user requests to access the file, he can check integrity by re-computing the MAC and compare it with pre-computed MAC. The problem with this method is that it gives assurance about correctness of data which is currently referenced but no assurance about that data which is not referenced. Again another problem is that data owner has to retrieve all data to check integrity which is an impractical way to perform auditing. Even to delegate this responsibility to TPA, data has to be transferred to TPA which may create online communication burden on the system.

A simple enhancement to this solution is that the data owner selects a set of arbitrary MAC keys; pre-computes the MACs for the complete data file and put out metadata to TPA. The TPA every time asks Cloud Server to calculate the fresh MAC for comparison. But the major problem of this method is its failure to support data dynamic since every time new MACs have to generate.

4.2. Sentinel Embedding

The Cloud User produces Sentinels to secure its own data at CS. These are generally one-way functions and Predefined numbers of sentinels are appended to the encoded file. The Cloud User can verify the integrity of data by giving challenge to CS by checking fixed number of sentinels. Since CS is not having any idea of the location of sentinels, it cannot modify or delete the data. Juels and Kaliski[9] used same technique to get Proof of Retrievability(POR) for stored files. The main problem with this method is the restricted number of challenges once the file is uploaded.

4.3. Random Sampling Authenticator

Before uploading a file to CS, authenticators are produced for each file by Data owner. The owner stores only some part of metadata related to file with him. Authenticators are generated using cryptographic algorithms such as RSA, ECC. Ateniese et al.[10] suggested Provable Data Possession(PDP) which uses the Homomorphic Verifiable Tags based on RSA to ensure possession of data at un-trusted store.

4.3.1. Homomorphic Authenticators

Homomorphic Authenticators [15] are one type of metadata associated with each data block. These are unforgeable authenticators.

Table1: Public Auditing Method Comparison

	Public Auditing	**Privacy Preserving**	**Data Dynamics**	**Batch Auditing**	**Shared Data**	**Identity Privacy (Privacy from TPA)**
Q. Wang et. al. [1]	RSA based Homomorphic Authenticators	X	Merkle Hash Tree (MHT)	Bilinear Aggregate Signature	X	X
C. Wang et. al. [2]	Homomorphic Linear Authenticators (HLA)	HLA + Random Masking	Merkle Hash Tree (MHT)	Modified Bilinear Aggregate Signature	X	X
B. Wang et. al. [3]	Homomorphic Authenticators	Random Masking	Index Hash Table (IHT)	Bilinear Maps	Ring Signature	Homomorphic Authenticable Ring Signature (HARS)
B. Wang et. al. [4]	Homomorphic Authenticators	X	Index Hash Table (IHT)	X	Proxy Re-Signature with User Revocation	X
H. Tian et. al. [5]	Homomorphic Verifiable Authenticators (HVA)	BLS* based Signature	Dynamic Hash Table (DHT)	Aggregate BLS Signature	X	X
T. Chakraborty et. al. [6]	ECC based Homomorphic authenticators	X	Merkle Hash Tree (MHT)	X	X	X
Sumathi D. et. al. [7]	Elliptical Curve Digital Signature Algorithm (ECDSA)	X	Modified EC-DSA	X	X	X

By verifying only aggregated authenticators, a verifier gets assurance that a sequential grouping of blocks is properly computed. By using this technique, before outsourcing the data user need additional information encoded along with the data. To compute the metadata, a data file is partitioned into blocks m_i (i=1,…,n). Homomorphic authenticator σ_i is computed on each block. This authenticator is metadata to ensure the integrity of the file. To ensure that CS stores this data honestly, the authenticators should be verified by data owner or TPA. A challenge $ch = \{(i, v)\}$ for a set of arbitrarily chosen blocks for random weights $\{v_i\}$ is submitted to CS. Because of the feature of the homomorphic authenticator, the server has to calculate a reply based on a linear arrangement of the sampled data blocks $\mu= \sum_i v_i.m_i$, and an aggregated authenticator $\sigma= \prod_i \sigma_i^{v_i}$ which is calculated from $\{m_i, \sigma_i, v_i\}$ $i \in ch$. The reply of μ and σ are checked by TPA. It gives assurance of data correctness on huge portion of cloud data. The problem with Homogeneous Authenticators is that the linear combination of blocks may disclose some information to TPA.

Many researchers [1,2,3,4,5] have used the RSA based Homomorphic Autheticators to ensure the correctness of cloud storage. Public auditing system using RSA based Homomorpihc Authenticators consists of mainly 4 functions *Key_Gen(), Sig_Gen(), Gen_Proof(), Verify_Proof().*

Key_Gen ():

The Data Owner C partitions the file F into blocks $m_1,\ldots.m_n$

- Key Generation Centre(KGC) compute C's private key S_{IDc} & public key P_{IDc}
- A signing key pair (ssk, spk) is generated by C.
- A private key sk: = (S_{IDC}, ssk) and public key pk: = (P_{IDc}, spk) is formed.

*** Boneh, Lynn,Shacham Short Signature**

Sig_Gen(sk,F):

- An arbitrary element name for the file F is chosen by data owner C.
- A file label t=name||sig_{ssk}(name) is computed with a signature on it.
- By taking arbitrary element for each block of F, Data Owner C generates the signature and transfers it to CS.

Gen_Proof (F,spk):

In the auditing protocol, TPA get backs the label t and verifies it by using spk. The procedure is finished if the test fails.

- TPA put a random challenge ch= $\{(i,v_i)\}_{i \in I}$.
- After receiving random challenge ch, CS computes $\mu=\sum_{i\in I} v_i$, m_i
- CS randomly picks r $\leftarrow \mathbb{Z}_p$, computes $R=e(u,v)^r$andγ=h(R)and send it to TPA.

Verify_Proof (μ, R):

- γ=h(R) is computed and then check $m_{i.}$

The problem with RSA based Homomorphic Authenticators is that the key size and cipher text produced is large in size which increases the computation and processing time. Some researchers have proposed Elliptical Curve Cryptography (ECC) based Homomorphic Authenticators [6, 7] for public auditing of cloud data storage.

To accomplish privacy-preserving with public auditing, Homogeneous authenticators are included with random masking technique [2]. In this technique, server's reply, a linear combination of sam pled blocks are masked randomly. So TPA has no source of information to infer a correct user data contents. In the Homogeneous Authenticator scheme, index information is added every time to prevent the CS from using the same authenticator to achieve a verification proof for a diverse block. But any insertion or deletion operation alter these indices and need to recompute and pass to CS again which lead to incapability to support data dynamics feature of public auditing.

4.4. Merkle Hash Tree

In cloud storage system, data owners may update data dynamically any time. Block level functions such as block alteration, block addition, and block removal are performed to maintain dynamic updates. But for auditing purpose, this is the most critical tasks in cloud data storage system. The homomorphic authenticator scheme uses the index information which is to be used in the authenticator calculation. By including this index information CS is prevented from using the same authenticator to achieve the verification evidence for a different block. So any update such as insertion or deletion operation will modify the indices of all the blocks. This modification leads to the recalculation of all equivalent authenticators.

If we are avoiding using such index information in authenticator calculation, such type of re-calculation will be avoided. Instead of index, tag of file can be used to improve security model. The Merkle Hash Tree (MHT) [1] is used to achieve data dynamics during auditing. With MHT, a set of components is intact and unchanged. The leaves of the MHT are considered as the file blocks m_i. A publicly certified root value R and the Auxiliary

Authentication Information (AAI) of every leaf is used to verify the block m. For the path joining from the leaf to the root, AAI includes all the siblings of the nodes. Some researchers [1, 2] have used MHT to support auditing on dynamic data.

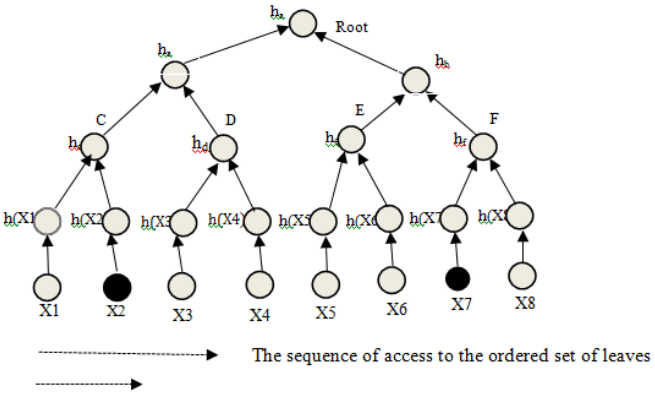

Fig. 2: MHT Authentication

Figure 2 shows example of authentication by MHT. Verifier gives authentication request h_r for {x2}. The prover provide authentication information AAI $\Omega_2 =<h(x_1),h_d>$. The verifier then check by computing, $h_c=h(h(x_1)||h(x_2))$, $h_a=h(h(x_c)||h(x_d))$, and $h_r=h(h(x_a)||h(x_b))$.

For auditing shared data, some researchers [3, 4] have proposed Index Hash Table (IHT) which uses virtual indexes (Version Number). These indices make sure that all blocks are in right order. Dynamic Hash Table (DHT) [5] is also used for auditing. DHT is a two dimensional data structure consisting two elements, file elements and block elements.

4.5. Bilinear Aggregate Signature

To perform the batch auditing, CS may handle multiple verification requests from different users. Most of the times CS prefer to aggregate K signatures of K distinct data files and verify it once to reduce the communication cost. It also gives efficient proof for the legitimacy of all the messages. Bilinear aggregate signature [14] can be used to perform batch auditing.

It has the following property:

For any u1, u2 ∈ G,

$$e\ (u1\ u2,\ v) = e\ (u1,\ v).e\ (u2,\ v)$$

and for any u,v ∈ G,

$$e\ (\Psi\ (u),\ v) = e\ (\Psi\ (v),\ u).$$

4.6. Homogeneous Authenticable Ring Signature

The most widely used functionality of cloud storage is to share data among different users. Users can share different documents with others in a team using the cloud storage services. Drop Box and Google Docs offers such type of features. To access or modify the shared data, the individual has to sign the block. During auditing, TPA may infer the history of all users who have signed one particular block, understanding the importance of that block in group. This compromises the identity privacy of users. To preserve it from TPA during auditing is one of the major problems in shared data.

For example, Alice and Bob share a file to one group on the cloud. Different users on the cloud, sign the file blocks individually. As shown in Figure 3, after performing several audits, TPA can reveal some information that the majority of the blocks are signed by Alice, who plays a very important role in the company. As well as the block no. 8 is modified most of the time, which gives information to the TPA that this block contains some important information.

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
Audit 1	A	A	A	A	A	A	B	A	B	B
Audit 2	A	A	A	A	A	A	A	B	B	B
Audit 3	A	A	A	B	B	A	A	A	B	B

A- Block signed by Alice
B- Block Signed by Bob

Fig. 3: Alice and Bob share a file in the cloud.

Using Ring signatures approach, a ring or group is formed of which different users are the members. User can sign a message or block to perform different operation so that a ring of possible signers is identified. The formation of ring does not reveal exactly which member of that ring actually generated the signature. With ring signature, the verifier just determines that the signature is generated by some member in the ring but it can't reveal any information about actual user who has generated the signature. Since traditional ring signature does not support block-less verification, the whole data file has to be downloaded by TPA during auditing process to check the identity of shared data. This may create online communication burden on the system. So to maintain identity privacy of each user and block less verification, Homomorphic authenticators with Ring signature [HARS] technique is given in [3] to audit the shared data.

A user may leave the group or revoke because of some misbehavior. The revoked user must not be allowed to access/change the shared data because the signature created by him is no longer applicable to the group. This revoked user must have signed some blocks earlier, which should be re-signed by present user. Before re-signing of existing user, he has to check the integrity of these blocks[16-19]. To do this, existing user has a burden to download all the blocks, verify correctness and resign the data. This may create an extra burden in the form of cost and time for existing user by allocating a large number of communication and computation resources. Instead of giving burden to existing user, using proxy re-signature scheme [4], CS can re-sign the block in support of existing user during revocation process. Proxy re-signature [11], use a partially reliable proxy who work as a converter for signatures among two users e. g. Alice and Bob. Proxy is one entity who can translate a signature done by Alice into the signature of Bob on the equivalent block (which is signed by Alice before revocation) without knowing the private key of Bob. Generally CS may work as a proxy and translate signatures of users during revocation.

5. Issues and Challenges

Different solutions are provided by researchers for the auditing of cloud storage. One solution is using external entity such as TPA, which check the accuracy of data on behalf of the cloud user. Because of the increasing popularity of cloud paradigm, many users are delegating auditing tasks to these TPAs. This delegation may overload TPA to handle multiple requests. Some of the researchers already had given solution of multiple TPAs [8] to audit the stored data. But still many issues such as auditing of shared data, user revocation are not resolved with this technique, which may be the research direction for future researchers.

Many schemes for public auditing given by researchers have used bilinear map technology. Even though this technology is efficient, it is costly in terms of computation time & economy. Expansion of cryptographic techniques can give solutions to such types of problems. Some of the researchers already given solution to these problems by modern cryptography techniques such as AES, Elliptic curve Cryptography. Modern new cryptographic primitives such as program obfuscation and structure preserving cryptography has powerful functionalities which previously not used by researchers. These can provide cloud users good and rich cloud services.

The auditing task again increases the burden on client because of pre-processing. The task of pre-processing can be delegated to the auditor so that the client should get relief in terms of infrastructure & cost. Communication cost is one of the major drawbacks of the previously proposed auditing schemes. If client's metadata is placed at CSP & secured by some software memory locks, then there is no need for data transfer between TPA & Cloud user. This may raise privacy issues of data, which may be the potential research area for future researchers.

6. Conclusion

Cloud Computing provides two basic service models, computing and storage. To meet user demands, all cloud computing services need high-performance cloud storage. So cloud storage is the significant components of cloud computing system. This paper primarily focuses on cloud storage security. The data in cloud storage is shared and accessed by many users which may create security and privacy issues. Cloud server is also one of the semi-trusted entities which can hide some unintentional failures and data errors to maintain their reputation. So the user must have some way to ensure the correctness of his data outsourced on cloud storage.

The user can delegate this responsibility to external parties such as TPA who will verify the integrity of cloud storage in support of the user using audit process. The system model for this auditing is presented which shows different components and its roles. Auditing is the process which determines the accuracy of data without retrieving the entire data. We then suggest different goals that must be achieved while auditing. Different approaches for auditing are summarized which can focus on techniques used for auditing. Through the in-depth analysis, all pros and cons of every approach are summarized in this paper. New challenges with regard to cloud computing in terms of auditing are also identified so that new research areas can be identified by researchers.

References

[1] Q. Wang, C. Wang, K. Ren, W. Lou & J. Li (May 2011). Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing. *IEEE Trans. Parallel and Distributed Systems*, 22(5),847-859,

[2] C. Wang, Sherman S. –M. Chow, Q. Wang, K. Ren &W. Lou (Feb 2013). PrivacyPreserving Public Auditing for Secure Cloud Storage" *IEEETrans. On Computer*, 62(2).

[3] Boyang Wang, Baochun Li &Hui Li. (March 2014).Oruta:Privacy-Preserving Public Auditing for Shared Data in the Cloud. *IEEE Trans. On cloud Computing*,2(1).

[4] Boyang Wang, Baochun Li &Hui Li.(February 2015).Panda:Public Auditing for Shared data with Efficient User Revocation in the Cloud.*IEEE trans. ServicesComputing*, 8(1).

[5] Hui Tian, Y. Chen, C. Chang, H. Jiang, Y. Huang, Y. Chen, Jin Liu, "Dynamic-Hash-Table Based Public Auditing for Secure Cloud Strage", IEEE Transactins on Service Computing, Vol.10, PP. 701-714, December 2015

[6] T. Chakraborty,Anil Dhami, P. Bansal, Tripti Singh, "Enhanced Public Auditability & Secure Data Storage in cloud computing", IACC IEEE Inter. Conf., 2013

[7] Sumathi D., R. V. Pujeri, "A Modified Elliiptic Curve Digital Signature Algorithm for Public Verifiability With Data Dynamics in Cloud Computing", Journal of computer Science, Vol. 10,PP.2077-2087, 2014

[8] Salah H. Abbdal, Hai Jin, Ali Yassin, Zaid A, M. Abdulridha H.,Zaid Alaa H., Deqing Zou, "An Efficient Public Verifiablity and Data Integrity using Multiple TPAs in Cloud Data Storage", IEEE 2nd International Conference on Big Data Security-HPSC-IDS, 2016

[9] A. Juels, B. Kaliski, Jr., " Pors; Proofs of Retrievability for Large Files", Proc, 14th ACM Conf. Computer and Comm. Security(CCS'07), PP.584-597, 2007

[10] G. Ateniese R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores", Proc, 14th ACM Conf. Computer and Comm. Security(CCS'07), PP.598-609, 2007

[11] M Blaze, G. Bleumer & M.Strauss. (1998). *Divertible Protocols and Atomic Proxy Cryptography*.Proc. Int'l Conf. Theory and application of Cryptographic Techniques (EUROCRYPT'98).

[12] P. Mell &T. Grance, (2009). Draft NIST Working Definition of Cloud Computing.

[13] C. Wang, K. Ren, W. Lou &Jin Li (August 2010). Toward Publicly Auditable Secure Cloud Data Storage Services.*IEEE Network*.

[14] D. Boneh, C. Gentry, B. Lynnand& H. Shacham. *Aggregate and Verifiablly Encrypted Signatures from Bilinear Maps*.Proc. 22nd Int'l Conf. Theory and application of Cryptographic Techniques (EUROCRYPT'03).

[15] Boyang Wang, Sherman S. M., Ming Li &Hui Li. (July 2013).*Storing Shared Data on the cloud via Security-Mediator*.Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS'13).

[16] ANNABATTULA, J., KOTESWARA RAO, S., SAMPATH DAKSHINA MURTHY, A., SRIKANTH, K.S. and DAS, R.P., 2015. Underwater passive target tracking in constrained environment. Indian Journal of Science and Technology, 8(35), pp. 1-4.

[17] HUSSAIN, S.N. and KISHORE, K.H., 2016. Computational Optimization of Placement and Routing using Genetic Algorithm. Indian Journal of Science and Technology, 9(47),.

[18] Vudatha, C.P., Nalliboena, S., Jammalamadaka, S.K.R., Duvvuri, B.K.K., Reddy, L.S.S., Automated generation of test cases from output domain of an embedded system using Genetic algorithms, ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology 5,5941989, pp. 216-220"

[19] Sastry, J.K.R., Suresh, A., Bhanu, S.J., Building heterogeneous distributed embedded systems through rs485 communication protocol, ARPN Journal of Engineering and Applied Sciences, 2015, 10(16), pp. 6793-6803