Privacy Preserving and Public Auditable Integrity Checking on Dynamic Cloud Data

Surmila Thokchom[1] and Dilip Kr. Saikia[2]
(Corresponding author: Surmila Thokchom)

Department of Computer Science and Engineering, National Institute of Technology Meghalaya[1]
Bijni Complex, Laitumkhrah, Shillong, India
Department of Computer Science and Engineering, Tezpur University[2]
Tezpur, Assam, India
(Email: surmila.thokchom@nitm.ac.in)

(Received Aug. 9, 2017; revised and accepted Nov. 18, 2017)

Abstract

The Cloud storage service allows data owner to outsource data storage at the Cloud which introduces security challenges requiring an auditor to check the integrity of stored data. This paper proposes an efficient auditing scheme for checking the integrity of dynamic data outsourced at untrusted Cloud storage. This scheme based on the Boneh and Boyen signature enables a third-party auditor to audit the client's data while preserving the privacy of the data. The scheme is found to be secure in the standard model. Complexity analysis shows the proposed scheme is efficient when compared to existing schemes.

Keywords: Batch Auditing; Cloud Computing; Dynamic Data; Privacy-Preserving Auditing

1 Introduction

Cloud computing introduces many attractive services. One such service is the cloud storage where the Cloud hosts the data and software of the client [21, 24]. Due to the economic advantage it offers and its elastic nature the service is very attractive for enterprises as well as individuals. However, the service introduces security concerns for the clients. The client loses control over their own data since it is stored in the Cloud servers. The data is put on risk from various threats in terms of privacy and integrity, both from within and outside the Cloud service provider. To overcome these threats the client cannot remain reliant on the assurance of the service provider alone. There is need for additional checks. To take care of the privacy issue the data can be transmitted and stored in encrypted form [1]. Provisions need to be made for verification of integrity of the data, preferably by an independent auditor without compromising on privacy of the data to the Cloud as well as the Auditor.

Various integrity checking schemes have been proposed over the years to check the integrity of the stored data in the Cloud. Integrity checking in these schemes is either performed by the data owner or by a third-party auditor who is employed by the data owner for verifying the integrity of the data on their behalf [16].

This paper presents a scheme for storage of dynamic Cloud data in the standard security model based on Boneh-Boyen signature [4, 5]. The scheme allows public auditability preserving data privacy. The scheme extends the Boneh-Boyen signature scheme that uses bilinear mapping to allow integrity checking of encrypted data so that the data privacy is not compromised while auditing. It is shown that the proposed scheme is efficient and secure through complexity analysis and simulation studies. The proposed scheme is designed to minimize the computational load on the auditor in the verification process. A mechanism is also presented for the scheme for batch auditing in multi-owner and multi-Cloud environment.

The rest of the paper is organized as follows. Section 2 discusses the existing related works. Section 3 presents the system model used and Section 4 presents the theoretical preliminaries. The proposed auditing protocol is presented in Section 5. Section 6 discusses the dynamic data operations with data integrity assurance support and Section 7 analyses the security issues of the model. The performance analysis is done in Section 8. Section 9 presents the multi-owner and multi-Cloud batch auditing support and Section 10 concludes the paper.

2 Related Works

Ateniese *et al.* [2] were the first one to propose provably secure schemes using RSA-based Homomorphic Verifiable Tag (HVT) to verify the integrity of stored data in the cloud without retrieving the data from the cloud. Their schemes have several drawbacks such as high overhead on server computation and communication cost and fails to provide fully secure data possession. Their scheme only supports static data. Zhu *et al.* [46] propose the Cooperative PDP (CPDP) scheme which is based on homomorphic verifiable response and hash index hierarchy. Their scheme emphasize on checking the integrity of client's data stored at multiple cloud service provider. Their scheme considers only static data. Hanser *et al.* [14] proposed a provable data possession based on elliptic curves cryptosystem which allows the data owner and the third party auditor to simultaneously audit the data outsource at the cloud storage. Their scheme only provides probabilistic guarantee of data possession and supports only static data.

Juels *et al.* [18] proposed the first Proof of Retrievability method where the sentinels are hidden among the regular data before outsourcing the data to the cloud storage. The verifier checks the data on the basis of the sentinels hidden among the data. Any changes in the data will affect the sentinels. The number of challenge is restricted by the number of embedded sentinels. Their scheme also supports only static data. Shacham *et al.* [28,29] proposed two compact PoR schemes. One is a public verifiable PoR scheme built using BLS signature [6] and the other one is a private verifiable PoR schemeusing the pseudo-random function. Schwarz *et al.* [27] propose a challenge-response scheme the data store remotely based on the algebraic signature properties in the peer-to-peer networks. Chen [10] extended the algebraic signatures for checking the possession of data in cloud storage. Their scheme has less overhead on the server and the client as compare to homomorphic based scheme. Their scheme also supports only static data. Wang [36] proposed a proxy provable data possession scheme using bilinear pairing technique. Their scheme uses a proxy for checking the integrity of the outsource data. Their scheme is also meant for static data only.

Ateniese *et al.* [3] updated the static Provable Data Possession methods called scalable Provable Data Possession based on symmetric key cryptography to make the scheme dynamic and to increase the efficiency and scalability. Their scheme supports data modification, data appending and data deletion operations. Their scheme restricts the number of updates and challenges and does not support data insertion.

The schemes proposed by Wang *et al.* [34,35] also supports only partial dynamic data operations. Erway *et al.* [12,13] proposed Dynamic Provable Data Possession which support full dynamic data operations such as insertion, deletion, modification and appending. The authors presented two varieties of Dynamic PDO scheme. The first scheme utilizes rank based authenticated skip list for supporting fully dynamic data operations and the second scheme uses rank based RSA trees. Both the schemes use homomorphic block tag. The first scheme does not support privacy. Second scheme has high probability of possession guarantee but incurs high computation as compared to the first scheme and it lacks flexibility for data updates. Wang *et al.* [37] proposed a scheme which enables public auditing and supports full data dynamic operations. Their scheme uses Merkle Hash Tree and bilinear aggregate signature. Their scheme does not consider privacy of the data from the auditor and incurs high computation at the auditor side. The scheme proposed by Hu *et al.* in [15], support dynamic data. However the privacy of the data might be compromised as the server needs to send linear combinations of the data as proof to the auditor. Wang *et al.* [32, 33] proposed a scheme which solves data leakage problem to auditor by using homomorphic linear authenticator (HLA) and random masking before sending the proof to the auditor. Their scheme suffers from large storage overhead at the server side due to large number of data tags. The scheme is found to be vulnerable to attacks from a malicious CSP and an outside attacker. The vulnerability of this scheme is because of the inappropriate definition and the use of private/public parameters during signature generation. Worku *et al.* [38] proposed a scheme which is more efficient than the protocol in [33]. However, Liu *et al.* [22] has shown that a malicious Cloud service provider can still produce a proof to the challenge given by an auditor without being caught even after deleting all files of a data owner.

Li *et al.* [19] proposed a light weight integrity checking scheme meant for low performance end devices. The scheme is a privacy preserving public auditable and supports dynamic data operation. Since the scheme is for low end devices data uploading is only for very small set. Zhang *et al.* [45] propose a scheme taking into consideration if the auditor colludes with the cloud server. The scheme utilizes the Bitcoin to generate the challenge random blocks and a check log file is maintained. The data owner will check the check log file to confirm that the auditor has done a fair integrity checking on the data. This adds a computation overhead for the data owner.

The scheme proposed by Yang *et al.*. [39] supports dynamic data, batch auditing, and preserves data privacy in random oracle model. The scheme uses Bilinearity property of Bilinear pairing. The server presents the proof of the data possession to auditor in an encrypted form which auditor can only verify. The scheme uses index table which will incur storage overhead proportional to the file size on the auditor.

Chattopadhyay *et al.* [8], proposed a scheme using simple low cost Boolean based encryption and decryption for image-files only. The encrypted data files will be shared on the Cloud. A threshold (t, n)-secret sharing scheme is used for obtaining the symmetric key. Liu *et al.* [23] proposed a public auditing scheme for the regenerating-code-based Cloud storage. In the absence of the data owner a proxy which has a priviledge is used to regenerate athenticators thus allowing the data owner not to be online all the time. Privacy of the data is preserved by randomizing the encode coefficients with a pseudorandom function. Chen *et al.* [11] proposed a remote data possession checking (RDPC) scheme. Their scheme is based on homomorphic hashing and the Merkle hash tree is used for enabling the data dynamics operations. Yu *et al.* stated in their paper [43] that [11] schemes is vulnerable to forgery attack and replace attack launched by a malicious server and their proposed scheme has shown improvement to overcome this vulnerability.

Lin *et al.* [20] proposed a scheme for mobile provable data possession. The scheme uses a hash tree data structure and a Boneh-Lynn-Shacham [6] short signature scheme. To reduce the computation overhead of the mobile data owner for generating the tag of the data block, trusted third party is utilized. Many communications take place between the trusted third party and the data owner while generating the data tag and that leads to high communication overhead between the third party and the data owner. Yi et al. [40] proposed a multi-copy Provable Data Possession. The scheme considers that the data owners stores multiple copies of sensitive data in the cloud. The scheme gives assurance that multiple copies of data are consistent with the latest version.

The scheme by Chen *et al.* [9] is based on homomorphic network coding signature scheme. It does not support dynamic data. Ma *et al.* [26] proposes an efficient privacy preserving scheme based on homomorphic network coding and RSA for the standard model and supports dynamic data. The scheme does not consider batch auditing.

Various Cloud storage auditing exists [7,41,42] which deals in key exposer problems. There are many integrity checking schemes for shared data among group of users [17,25,30,31,44]. This paper does not go into details of the shared data schemes since shared data is not considered in the proposed scheme.

3 The System Model

The proposed auditing scheme considers the cloud storage architecture as illustrated in Figure 1. This storage architecture comprises three entities; The client or the data owner, the Cloud servers and the third-party auditor. The client creates data and stores it at the Cloud storage. Upon requirement the client can retrieve and update the data. The Cloud server stores and maintains the client data and gives access the data to the client. The auditor is a neutral trusted entity who has the expertise and resources to perform integrity checking on large data sets. The auditor periodically or upon request will challenge the Cloud server to provide proof of integrity of the outsourced data. Based on the proof provided by the Cloud server on the challenge sets, the third-party auditor will deliver unbiased audit reports to the client.

The data integrity threats of the client data from the server may be non-malicious or malicious. At any time, if the integrity of the client data is compromised, the Cloud servers may try to hide it so as to maintain its reputation. Therefore, the Cloud server is considered untrusted. The dynamic data stored on the cloud may face the following attack [39]:

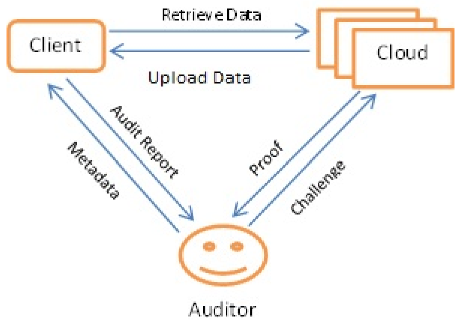

Figure 1: System model

1) *Replay attack*: The client's data may not be updated correctly on the server and to make up the mistake, the server may use the previous uncorrupted pair of data block and data tag, to replace the challenge pair of data and data tag so that the auditing will passed.

2) *Forging attack*: If the Cloud server has the information required for generating the data tags, it can forge the data when the client updates his data to a new version. The forging by the server may go undetected by the auditor if suitable provisions are not made.

On the other hand, the integrity of the auditor is not questioned. However unprotected data may lead to loss of privacy. Hence the data is to be transmitted and stored in encrypted form and the auditor should be able to verify the integrity of the stored data with zero knowledge of the content.

4 Preliminaries

The scheme proposed is primarily based on Bilinear Map and Boneh-Boyen signature scheme. These are briefly discussed below

4.1 Bilinear Maps

Let G_1, G_2 and G_T be multiplicative cyclic groups of order p. Let g_1 be the generator of G_1 and g_2 be the generator of G_2. A map $e : G_1 \times G_2 \rightarrow G_T$ will be a bilinear map if it satisfies the following properties:

- *Computable*: an efficient algorithm exists for computing map e;
- *Bilinear*: $e(u^a, v^b) = e(u, v)^{ab}$ for all $u \in G_1$, $v \in G_2$ and $a, b \in Z_p$;
- *Non-degeneracy*:$e(g_1, g_2)$ does not equal to 1.

4.2 Boneh-Boyen Signature Scheme

The Boneh-Boyen signature scheme is based on Bilinear Mapping and it comprises the three functions of *Key Generation*, *Signing* and *Verification*. These are defined as follows.

KeyGeneration: Generate the key pair *(PK, SK)* as follows. Choose random integers $x, y \rightarrow Z_p^*$ and compute, $A = g_2^x \in G_2$, $B = g_2^y \in G_2$, $z \leftarrow e(g_1, g_2)$ The public key is $PK{=}(g_1, g_2, A, B, z)$ and the private key is $SK{=}(g_1, x, y)$.

Signing: Given a message $m \in Z_p$ and a private key $SK{=}(g_1, x, y)$, select a random value $r \in Z_p$ and compute $S = g_1^{\frac{1}{(x+m+yr)}}$ with the inverse $\frac{1}{(x+m+yr)}$ computed modulo p. The signature is $\sigma = (S, r)$.

Verification: Given a message $m \in Z_p$ and a public key $PK{=}(g_1, g_2, A, B, z)$ and signature $\sigma = (S, r)$, it is verified by checking $e(\sigma, A, g^m, B^r) = z$ if is true.

Table 1: Notation used

A, B	Public Keys
α, x, y	Private Keys
σ	Tag
s	Intermediate value of Tag
r, h, t	Random values
n	Number of blocks
k	Number of Challenge blocks
C_{info}	Meta data

5 Proposed Cloud Storage Auditing Protocol

The proposed Cloud storage auditing protocol assumes that a data file F is split into n number of data blocks $(b_1, b_2, b_3 \ldots\ldots b_n)$ and these data blocks are encrypted individually with a suitable encryption algorithm to produce the encrypted data file $C{=}(c_1, c_2, c_3 \ldots\ldots c_n)$ before they are uploaded. The scheme uses the following five functions for the auditing as illustrated in Figure 2.

1) $GenKey(k) \rightarrow K$: This function is executed at the client side taking security parameter k as input and produces a secret key and public key pair, *K=(SK,PK)*;

2) $GenTag(C, SK, h, t) \rightarrow S$: This function takes an encrypted data file C and the secret key SK, random values h and t to compute a set of the data tags S, one for each of the data blocks in C. The encrypted data blocks in C and their corresponding tags in$S$ and t are uploaded to the Cloud storage and corresponding C_{info} which comprises the tuple $\langle$*Index, BlockName, Version number, h value for the data block uploaded in the cloud storage* $\rangle$. Here h is treated as a secret value between data owner and the auditor;

3) $GenChall(C_{info}) \rightarrow Q$: The auditor executes this function with the meta data,C_{info} as input to generate a challenge Q, to be sent to the Cloud server;

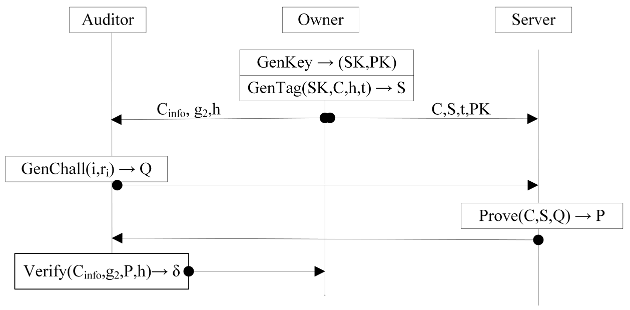

Figure 2: Work flow of the auditing scheme

4) $Prove(Q, C, S, PK) \rightarrow P$: The cloud server executes this function taking the challenge Q received from the auditor, the stored data file and its set of random t value *(C,t)* and the public key PK as inputs to produce a proof P;

5) $Verify(P, Q, C_{info}) \rightarrow V$: The auditor executes this function with inputs - the proof P provided by the server, the challenge Q, the metadata C_{info} and to produce the output value of *V =1*, if the proof is correct, otherwise produce.

5.1 Theoretical Basis

The original Boneh-Boyen signature scheme is adopted for the proposed scheme as follows:

- Let $(\alpha, x, y) \in Z_p$ be a randomly generated value used as private keys *SK=(α, x, y)* and *PK= (A,B)* the public keys computed as- $A = g_2^{\frac{y}{(\alpha+x)}}$, $B = g_2^{\frac{1}{(\alpha+x)}}$

- Let $\Sigma = (\sigma_1, \sigma_2 \ldots \sigma_n)$ be the set of n tags with $\sigma_i = (t_i, s_i)$ generated for the encrypted data block c_i in, $C{=}(c_1, c_2 \ldots c_n)$ where t_i and h_i are two random values and $S_i \in G_2$ computed as-

$$s_i = h_i^{\frac{\alpha+x}{yt_i+c_i}} \tag{1}$$

- Let $Q = \{i, r_i\}, i = 1, 2 \cdots k$ be a challenge set generated with random values $r_i \in Z_p^*$, one for each of the chosen k number of data blocks in set D.

- Let Z be a quantity computed using h value from the metadata C_{info}, for each of the k chosen data blocks in D.

$$Z = \prod_{d=1}^{k} e\left(h_d, g_2\right)^{r_d^2} \tag{2}$$

- Let P be the proof generated for each of the chosen data blocks as-

$$P = \prod_{d=1}^{k} e\left(s_d^{r_d}, A^{r_d t_d}, B^{C_d r_d}\right) \tag{3}$$

Theorem: As per the bilinearity property $P = Z$. i.e.,

$$\prod_{d=1}^{k} e(s_d^{r_d}, A^{r_d t_d}, B^{C_d r_d}) = \prod_{d=1}^{k} e\left(h_d, g_2\right)^{r_d^2} \tag{4}$$

Proof.

$$P = \prod_{d=1}^{k} e\left(s_d^{r_d}, A^{r_d t_d}, B^{C_d r_d}\right)$$

$$= \prod_{d=1}^{k} e\left(h_d^{\frac{r_d(\alpha+x)}{yt_d+C_d}}, g_2^{\frac{r_d y t_d}{\alpha+x}}, g_2^{\frac{r_d C_d}{\alpha+x}}\right)$$

$$= \prod_{d=1}^{k} e\left(h_d^{\frac{\alpha+x}{yt_d+C_d}}, g_2^{\frac{yt_d+C_d}{\alpha+x}}\right)^{r_d r_d}$$

By Bilinearity Property we can rewrite the expression as

$$= \prod_{d=1}^{k} e(h_d, g_2)^{\frac{(\alpha+x)(yt_d+C_d)r_d^2}{(yt_d+C_d)(\alpha+x)}}$$

$$= \prod_{d=1}^{k} e(h_d, g_2)^{r_d^2}$$

□

6 Data Integrity Assurance Support for Dynamic Data Operation

The auditor and the client will be maintaining metadata which consists of block, version, h value for each of the blocks. The index number is the current block number. For tag generation of each of the blocks the h value is used. There are three types of operations the client can perform to update their data.

1) *Data block modification*: Consider the operation ofthe client modifyingthe data block, *Block-b to Block-b'*. First of all, the client will download this block from the cloud server and make the required modification on the data block. The client will compute a new tag for the modified block. To compute new tag, the client will generate a random value for h and also a random value for t and compute the new tag σ using Equation (1). The client will then update *version* and h value in the metadata table. The client will then upload the modified block and new tag value and the new t value to the cloud server. The client will communicate the auditor the new h value and the *version* number of the modified block.

2) *Data block insertion*: Now consider the operation of inserting a newdata block, *Block-0* after the k^{th} block. The client will generate a random value h and t for computing the tag of the block. The client will update the metadata table by moving down all items following the k^{th} block entry in the table by one block. The client will then upload the new block and its corresponding tag value and t value to the cloud server. The client will communicate the auditor the insertion of the new block and its h value.

3) *Data block deletion*: If the client wants to delete the k^{th} block, it will send a request to the cloud server for removing the block and shall communicate the same to the auditor. In the metadata table, both the auditor and the client will delete the k^{th} entry from the table and shall move up the following entries in the table by one slot each.

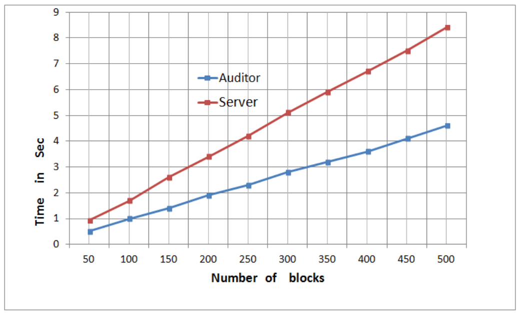

Figure 3: Computation costs of the auditor and the server

7 Security Analysis of the Model

As discussed in Section 2, in a cloud storage system, when dynamic data are stored, the cloud server may carry out replay attack and forging attack. In the proposed scheme, h value is used while computing the tag for each block and as stated it is unique for each version of each block. Hence a replay attack will never pass the audit check.

On the other hand, if the server could forge the data tag, it can pass the audit using any data and its forge data tag. Forging a data tag in our scheme requires the server should be able to predict the value of h, which is a randomly selected unique value for each block. If any block is modified, this modified block will have a new h value. A forging attack by the server will therefore get detected in audit.

8 Performance Analysis

The Communication and Computation cost of the proposed scheme can be computed as follows.

- *Communication Cost*: The challenge and the proof parts will give the communication cost between the auditor and the server. In the challenge part, the cost depends on the number of blocks d which the auditor sent for audit. In the proof, it is the only the proof result. So the communication cost will be O(d).

Table 2: Comparison with different cloud data integrity auditing schemes

Schemes	Computation cost		Communication cost	TPA	Privacy	Dynamic	Model	Batch
	Server	Auditor						
[37]	O(d log n)	O(d log n)	O(d log n)	Yes	Yes	Yes	ROM	Yes
[33]	O(d log n)	O(d log n)	O(d log n)	Yes	Yes	Yes	ROM	Yes
[2]	O(d)	O(d)	O(d)	No	No	No	ROM	No
[39]	O(d)	O(d)	O(d)	Yes	Yes	Yes	ROM	Yes
[12]	O(d log n)	O(d log n)	O(d log n)	No	No	No	Standard	No
[9]	O(d)	O(d)	O(d)	Yes	Yes	No	Standard	No
[26]	O(d)	O(d)	O(d)	Yes	Yes	Yes	Standard	No
Our scheme	O(d)	O(d)	O(d)	Yes	Yes	Yes	Standard	Yes

- *Computation Cost*: The scheme in this paper involves three computations cost, namely at the owner side, the server side and the auditor. The simulation of scheme has been done on a Windows system with an Intel(R) Core(TM) i3 CPU at 3.60 GHz and 4.00GB RAM. The code for simulation of the scheme uses the pairing-based cryptography library version 0.5.12. An elliptic curve of MNT d159, with a base field size of 159 bits and embedding degree 6 is chosen. The d159 curve has a 160-bit group order, which means prime p is 160-bits long. The simulation is run multiple times and averaged to obtain stable results. Computation cost of the auditor and server versus the number of data blocks are compared in Figure 3. As shown in the Figure 3, data blocks are taken up to 500 blocks and block size is 2 KB. For 500 blocks (1000 KB) of data the Server requires around 8 seconds for providing the proof and the auditor requires around 4 seconds for verification.

  The graph in Figure 3, shows computation cost of the Auditor and the Server. The computation cost for the auditor consists of the time for auditing and verifying the data. Auditing time is just the generation of random numbers for the queried number of data blocks. The verifying time will compute Equation (3), which comprises mapping of each of the queried data blocks and then multiplying each of them. Each of the terms in Equation (3), takes *O(1)* time to be computed and the expression consists of product of challenged number of blocks. So time taken to compute will be *O(d)*, where d is the number of challenge blocks. The computation cost for the server is the time for proving the possession of the data given as a challenge. The server computes the expression in Equation (2) for providing the proof, which again takes *O(d)*, as each term in this equation takes *O(1)* to compute. From the given result, it is apparent that in the integrity checking protocol most of the computations are done at the server side for computation of the proof and thus minimizing the load on the auditor in the verification.

  Table 2 presents a comparison of the proposed scheme with other existing integrity schemes [2, 9, 12, 26, 33, 37, 39] in terms of computation complexity of the server and the verifier computation, communication cost, support for third party audit (TPA), preserving privacy, support of dynamic data [26, 39]. In the table, d represents the number of challenge blocks, n is the total number of blocks stored in the cloud.

9 Multiowner and Multicloud Batch Auditing Support

The proposed scheme can be extended to support batch auditing for multicloud and multiowner. Let n be the number of owner and l be the number of cloud service provider. Steps to be followed are as follows:

1) *Initialization*: Each of the owner will run the *GenKey(k)* algorithm to generate a pair of private and public keys. Each of the data owner can generate different private and public keys for different cloud servers. They will then run the *GenTag(c,sk)* algorithm to generate tag $\sigma_i = (t_i, S_i)$for each of their data blocks using Equation (1) as in *GenTag* algorithm in Section 4. The clients then uploads the data blocks and the corresponding tags to the chosencloud servers. Each owners then sends meta data to the auditor. The meta data consists of the cloud server name, block number and the h value for each of the datablocks.

2) *ChallengeBatchwise*: The auditor executes this function to generate challenge to l number of cloud severs for n number of data owners. It takes C_{info} as input to generate a challenge set Q, which consists of total m number of data blocks, to be sent to the cloud server. Out of m number of data blocks, each owner has k number of data blocks. The auditor generates a random number for each of the selected data blocks and sends these block numbersalong with the respective random numbers to the corresponding cloud servers.

3) *ProveBatch*: As a proof each cloud server will send the proof value to the auditor as:

$$P = \prod_{o=1}^{n} \prod_{d=1}^{k} e\left(s_d^{r_d}, A^{r_d t_d}, B^{C_d r_d}\right)$$

4) *VerifyBatch*: The auditor will first compute the product of all the proofs provided by individual cloud servers (cs).

$$\prod_{cs=1}^{l} P_{cs}$$

The auditor then computes the following product for the entire set of data blocks of all the owners in different cloud servers.

$$\prod_{d=1}^{m} e(h_d, g_2)^{r_d^2}$$

The auditor then verifies the integrity of the data by checking for the following equality:

$$\prod_{cs=1}^{l} P_{cs} = \prod_{d=1}^{m} e(h_d, g_2)^{r_d^2} \quad (5)$$

10 Conclusion

The paper has presented an integrity checking scheme in standard model that supports dynamic data operations and public verifiability while preserving data privacy. The scheme is extended to support batch auditing for multi-owner and multi-cloud servers to increase the efficiency of the auditing scheme. It is shown that the scheme is resistant to replay and forge attacks by the server. In the scheme, the computation cost of verifying the data integrity is kept low for the auditor. The proposed scheme is shown to comply with the best existing schemes in terms of computational and communication complexity. The scheme also has the advantages of using the standard security model and supporting batch auditing in the multi-owner and multi-cloud environment.

The proposed scheme however suffers a small storage overhead at the auditor side in terms of requires storing the tag value h for each block to be stored at the auditor side thus creating some storage overhead at the auditor side. Future efforts to overcome this overhead and incorporating provision for sharing of the data by multiple clients may be worthwhile.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40-48, 2018.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Commmunication Security (CCS'07)*, pp. 598–609, Oct. 2007.

[3] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *ACM Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, pp. 1–10, Sep. 2008.

[4] D. Boneh and X. Boyen, "Short signatures without random oracles," in *In Advances in Cryptology (EUROCRYPT'04)*, pp. 56–73, May 2004.

[5] D. Boneh and X. Boyen, "Short signatures without random oracles and the sdh assumption in bilinear groups," *Journal of Cryptology*, vol. 21, no. 2, pp. 149–177, 2008.

[6] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Advances in Cryptology (ASIACRYPT'01)*, pp. 514–532, Dec. 2001.

[7] Z. Cao, L. Liu, and O. Markowitch, "Analysis of one scheme for enabling cloud storage auditing with verifiable outsourcing of key updates," *International Journal of Network Security*, vol. 19, no. 6, pp. 912–921, 2016.

[8] A. K. Chattopadhyay, A. Nag, and K. Majumder, "Secure data outsourcing on cloud using secret sharing scheme," *International Journal of Network Security*, vol. 19, no. 6, pp. 912–921, 2016.

[9] F. Chen, T. Xiang, Y. Yang, and S. S. M. Chow, "Secure cloud storage meets with secure network coding," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM'14)*, pp. 673–681, Apr. 2014.

[10] L. Chen, "Using algebraic signatures to check data possession in cloud storage," *Future Generation Computer System*, vol. 29, no. 7, pp. 1709–1715, 2013.

[11] L. Chen, S. Zhou, X. Huang, and L. Xu, "Data dynamics for remote data possession checking in cloud storage," *Computers and Electrical Engineering*, vol. 39, no. -, pp. 2413–2424, 2013.

[12] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proceedings of ACM Conference on Computer and Commmunication Security*, pp. 213–222, Nov. 2009.

[13] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information System Security*, vol. 17, no. 4, pp. 1–29, 2015.

[14] C. Hanser and D. Slamanig, "Efficient simultaneous privately and publicly verifiable robust provable data possession from elliptic curves," in *Proceedings of the 10th International Conference on Security and Cryptoghraphy*, pp. 1–10, July 2013.

[15] H. Hu and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in *Proceedings of ACM Symposium on Applied Computing*, pp. 1550–1557, Mar. 2011.

[16] M. S. Hwang, T. H. Sun, C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *Journal of Circuits, Systems, and Computers*, vol. 26, no. 5, 2017.

[17] T. Jiang, X. Chen, and J. Ma, "Public integrity auditing for shared dynamic cloud data with group user revocation," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2363–2373, 2016.

[18] A. Juels, J. Burton, and S. Kaliski, "Pors: Proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Commmunication Security (CCS'07)*, pp. 584–597, Oct. 2007.

[19] J. Li, L. Zhang, J. K. Liu, H. Qian, and Z. Dong, "Privacy-preserving public auditing protocol for low-performance end devices in cloud," *IEEE Transaction on Information Forensics And Security*, vol. 11, no. 11, pp. 2572–2583, 2016.

[20] C. Lin, Z. Shen, Q. Chen, and F. T. Sheldon, "A data integrity verification scheme in mobile cloud computing," *Journal of Network and Computer Applications*, vol. 77, no. -, pp. 146–151, 2017.

[21] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing", *International Journal of Network Security*, vol. 18, no. 4, pp. 650–666, 2016.

[22] H. Liu, L. Chen, Z. Davar, and M. Pour, "Insecurity of an efficient privacy-preserving public auditing scheme for cloud data storage," *Journal of Universal Computer Science*, vol. 21, no. 3, pp. 473–482, 2015.

[23] J. Liu, K. Huang, H. Rong, H. Wang, and M. Xian, "Privacy-preserving public auditing for regenerating-code-based cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1513–1528, 2015.

[24] L. Liu, Z. Cao, C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 29–35, 2018.

[25] X. Liu, Y. Zhang, B. Wang, and J.Yan, "Mona: Secure multi-owner data sharing for dynamic groups in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1182–1191, 2013.

[26] M. Ma, J. Weber, and J. Berg, "Secure public-auditing cloud storage enabling data dynamics in the standard model," in *Third International Conference on Digital Information Processing Data Mining and Wireless Communications*, pp. 170–175, July 2016.

[27] T. S. J. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, pp. 12–22, July 2006.

[28] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Advances in Cryptology (ASIACRYPT'08)*, pp. 90–107, Dec. 2008.

[29] H. Shacham and B. Waters, "Compact proofs of retrievability," *Journal of Cryptology*, vol. 26, no. 3, pp. 442–483, 2013.

[30] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," *IEEE Cloud Computing*, vol. 2, no. 1, pp. 43–56, 2014.

[31] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," *IEEE Trans Services Computing*, vol. 8, no. 1, pp. 92–106, 2015.

[32] C. Wang, S. chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, pp. 1–9, Mar. 2010.

[33] C. Wang, S. chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage in cloud computing," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[34] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," *IEEE Network*, vol. 24, no. 4, pp. 19–24, 2010.

[35] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards secure and dependable storage services in cloud computing," *IEEE Transactions on Service Computing*, vol. 5, no. 2, pp. 220–232, 2012.

[36] H. Wang, "Proxy provable data possession in public clouds," *IEEE Transaction on Services Computing*, vol. 6, no. 4, pp. 551–559, 2013.

[37] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[38] S. Worku, C. Xu, J. Zhao, and X. He, "Secure and efficient privacy-preserving public auditing scheme for cloud storage," *Computers and Electrical Engineering*, vol. 40, no. 5, pp. 1703–1713, 2014.

[39] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2013.

[40] M. Yi, J. Wei, and L. Song, "Efficient integrity verification of replicated data in cloud computing system," *Computers and Security*, vol. 65, pp. 202–212, 2017.

[41] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Transactions on Information Forensics Security*, vol. 10, no. 6, pp. 1167–1179, 2015.

[42] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with verifiable outsourcing of key updates," *IEEE Transactions on Information Forensics Security*, vol. 11, no. 6, pp. 1362–1375, 2016.

[43] Y. Yu, J. Ni, M. H. Au, H. Liu, H. Wang, and C. Xu, "Improved security of a dynamic remote data possession checking protocol for cloud storage," *Expert*

Systems with Applications, vol. 41, pp. 7789–7796, 2014.

[44] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1717–1726, 2015.

[45] Y. Zhang, C. Xu, H. Li, and X. Liang, "Cryptographic public verification of data integrity for cloud storage systems," *IEEE Cloud Computing*, vol. 3, no. 5, pp. 44–52, 2016.

[46] Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multi-cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2231–2244, 2012.

Biography

Surmila Thokchom is an Assistant Professor of Computer Science and Engineering at the National Institute of Technology Meghalaya, India. Her research interest includes cryptography, information security, cloud computing.

Dr. Dilip Kr. Saikia is a Professor of Computer Science and Engineering at Tezpur University, India. Current areas of his research interest include Software Defined Networks, Network Function Virtualization, Wireless Sensor Networks and Network Security.