International Journal of Research in Advent Technology, Vol.6, No.5, May 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

Certificateless Public Auditing Protocol in Cloud

S. Mahaboob Basha[1], C. Shoba Bindu[2], C. Sasikala[3], P. Dileep Kumar Reddy[4]
Department of Computer Science and Engineering, jntuacea, Anantapuramu [1,2,3,4]
Email:basha.shaikatp06@gmail.com[1], *shobabindhu@gmail.com*[2],
sasikalareddy27@gmail.com[3], *dileepreddy503@gmail.com*[4]

Abstract-Cloud computing provides storage as a service where users can store data and access whenever necessary. In this model users may loss his physical control over data. Data can be lost, damaged or corrupted due to the presence of attackers or server failures either from inside or outside security threats in the cloud. Data auditing is essential for implementing secure cloud storage. It offers data owners checking their integrity of data without downloading the entire data. Most of the existing protocols are designed based on Public Key Infrastructure (PKI), hereto initiate data auditing the Third Party Auditor must validate the public key of the data owner. But the major problem with PKI is certificate management. For addressing this issue, in this paper we proposed a Certificateless public auditing protocol basing on Lattices. In this approach, we design public data auditing protocol by using data owner's identity which guarantees the right public key for performing data auditing, so that we can avoid the certificate management problem in PKI based auditing protocols by using our Certificateless public auditing protocol. Finally, our security analysis shows that Certificateless public auditing protocol is secure and efficient and it also reduces the computation cost over the public verifier.

Keywords- Public Auditing Protocol; Certificateless Signatures; Lattices; Key Generation Centre.

1. INTRODUCTION

Cloud Storage offers the users to outsource their data to the remote servers, by outsourcing their data to the cloud they may get benefits such as data access with independent of geographical locations, and eases from the problem of storage management and capital expenditure is avoided on software, hardware, and maintenances, etc. In fact, the cloud resources are very powerful and reliable than that of users own resources, but the data on the cloud is still vulnerable to many threats due to loss of control over data. These threats will compromise Integrity, Confidentiality and Availability of data. Example, for monetary reasons an untrusted CSP may reclaim storage space by removing the data that has not been used or rarely accessed and he may hide data loss incident to maintain prominence. Besides cloud server could not be trusted. Hence it is necessary for a user to check frequently whether the data is stored properly at the remote servers.

To efficiently perform the data auditing at untrusted cloud servers many techniques have been proposed, most of the Auditing protocols [3][7][8] depend on Public Key Infrastructure. In this scenario, prior to performing Auditing, the Third Party Auditor (TPA) has to validate the certificate of data owner to know the authenticity of the public key. Here it faces key management problem that the public key has to be managed by the data owner. Certificate generation, updation, validation and revocation causes a burden on Third Party Auditor (TPA) and also brings high computation cost. So, for the source constrained cloud users, this type of Auditing protocols became a major problem. To overcome this, authors have suggested Identity-based Auditing protocols [1][2][10][11] which uses Identity-Based Signatures (IBS) to avoid certificate management problem in PKI based Auditing protocols. But, the drawback of IBS is, it suffers from key escrow problem due to its complete dependency on KGC (Key Generation center) to generate the private keys.

By making use of Certificateless Signatures (CLS) [4][5], a verifier can be able to check the integrity of the outsourced data without suffering from complex certificate management problem in PKI and key escrow problem in IBS. Here in this CLS, the private keys are generated by combining the partial private key generated by KGC with the data owner's secret information, and the public key itself is simply his identity such as her name or email address. So that, there is no need to manage the certificates to guarantee the genuine public key of the owner which initiates the data integrity checking process in the cloud.

2. SYSTEM MODEL

The system model of Certificateless public auditing protocol in the cloud is shown in below Figure1. It

includes the four entities:i) Data owner: It is an entity that has a large number of files for outsourcing to the cloud server. ii) Cloud Server: It has significant storage space for the client to store their data in the cloud. iii) TPA: It is fully trusted and it is having expertise capabilities in performing data auditing on behalf of data owner iv) KGC: It is responsible for generating the partial private key of a data owner based on her identity information.

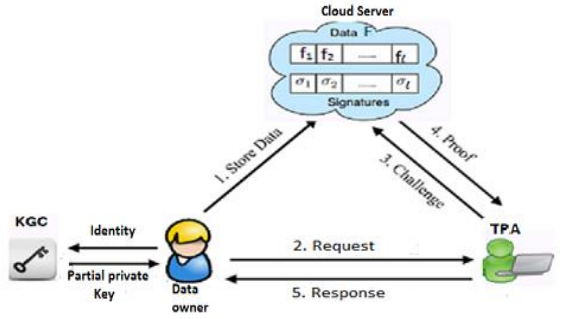

Figure 1. The system model for Certificateless public Auditing protocol in cloud

2.1. Framework

- **Setup (n)**:- This algorithm is run with the help of Key Generation Center, which takes the input of parameter n and produces the output of master secret key *msk.*
- **Extract Partial Private key (PParams,msk,ID) :-** This algorithm is run by KGC, takes the *PParams*, master key *msk*, and the data owner's identity *ID* as input and outputs the partial private key p_{ID}.
- **Set secret value(PParams,ID):-** This algorithm is run by data owner, takes *PParams,ID* as input and chooses a random matrix as secret value and set his secrete value as s_{ID}.
- **Set-Privatekey**(**PParams, p_{ID}, s_{ID}**):-This algorithm is run by data owner, takes the *PParams*, partial-private key p_{ID} and the sectret value s_{ID} as input then it combines the partial private key p_{ID} with secret value and outputs the full private key *sk.*
- **Set-Publickey(PParams, s_{ID}):-** This algorithm is run by data owner, takes the *PParams*, and s_{ID}as input and outputs public key *PK.*
- **Sign(PParams, ID,sk,F, id):-** This algorithm is run by data owner, takes *PParams*, owner's identity *ID*, private key *Sk* , File *F* and File identity *id* as input and outputs the signature σ_i.
- **Challenge**(**id,F**):- This algorithm is run by TPA, takes File *F* and File identity *id* as input and outputs the challenge message as chal={id, i ,v_i} and uploads it to server.
- **ProofGen(PParams,F, ϕ,chal,id**):-This algorithm is run by cloud server, takes the *PParams*, File *F*, signature set ϕ, challenge, and File identity *id* as input and it outputs the proof of possession *P.*
- **Proofcheck**(**PParams, ID, id, Proof, Chal**):-This algorithm is run by TPA, takes the *PParams*, owner's identity *ID*, file identity *id*, chal, and the proof *P* as input, and verifies whether file is intact or not.

3. PRELIMINARIES

A lattice L with m-dimensions[6][12] in R^mEuclidean space and aset of the integral combinations of n linearly independent vectors b_1,b2,b3, ..., b_n. The L Lattice was represented by

$$L(b_1, ..., b_n) = \{ \sum_{i=1}^{n} x_i b_i, x_i \in Z\}$$

Where B=(b_1, ..., b_n) is called the basis for the lattice, integer n is a number of vectors in a basis and m is the dimensions of the lattice. Let A $\in Z_q^{n\times m}$ and n and q are two positive integers we define the q-array lattices as follows:

$$L^{\perp}(A) = \{ v \in Z^m : Av = 0 \bmod q\}$$

$$L(A^T) = \{ v \in Z^m : \exists\ s \in Z_q^n \text{ , such that } v = A^T s \bmod q \}$$

For any u $\in Z_q^n$, an integral solution to AX=u mod q represents the shifted lattice or coset of the lattice.

$$L_u^{\perp}(A) = \{ v \in Z^m = Av = u \bmod q\} = L^{\perp}(A) + X$$

3.1. Discrete Gussians on lattices

For any vector c $\in R^m$, and positive s>0 then the Gaussian function on R^m centered at c with parameter s is define as:

$$\forall\ x \in R^m\ ,\ \rho_{s,c}(X) = \exp(-\pi \| x - c \| / s^2)$$

The discrete Gussian distribution over m-dimensional lattice L is defined as:

$$\forall\ x \in L\ ,\ D_{L,s,c}(X) = \frac{\rho_{s,c}(X)}{\rho_{s,c}(L)},\ \text{where } \rho_{s,c}(L) = \sum_{x \in A} \rho_{s,c}(X).$$

3.2. Trapdoor and Basis Delegation functions

One-way trapdoor function and basis delegation functions which are used in our signature generation algorithm is defined as follows.

TrapGen (n,m,q): For integers n , m, q with q≥ 2 and m ≥ 5nlgq, it generates a matrix A $\in Z_q^{n\times m}$ and T_A as

the short basic of the lattice $L_q^{\perp}$ (A) and $\| T_A^{\sim} \| \leq$ m. $\omega.\sqrt{\log m}$

SampleBasis[6] : Given the integers n ,m ,q with q≥ 2 and m ≥ 5nlgq and on input of A= (A_1, A_2,...,A_k) $\in Z_q^{n\times km}$, on the set S⊆ [k] , a trapdoor basis T_s of $L_q^{\perp}(A_S)$ and an integer I ≥ $\| T_S^{\sim} \| . \sqrt{km} .\omega.\sqrt{\log km}$, by taking these parameters as input SampleBasis (A,T_s, S, I) generates a matrix B as a basis for the lattice $L_q^{\perp}$ (A) with $\| B^{\sim} \| \leq$ I.

SamplePre[6]: For integers n , m, q with q≥ 2 and m ≥ 5nlgq, on the input of a matrix A $\in Z_q^{n\times m}$ and trapdoor basis T_A of the lattice $L_q^{\perp}$ (A), a vector v $\in Z_q^n$ and an integer i ≥ $\| T_A^{\sim} \| . \omega.\sqrt{\log m}$, the PPT Samplepre (A,$T_A$, v, i) generates a vector y such that distribution of y is within the negligible statistical distance of D L_q^v (A), i.

3.3. SIS Hard Problem in Lattices

Small Integer Solution (SIS) problem [12] is defined as for an integer q, real β and a matrix A$\in Z_q^{n\times m}$, find a nonzero integer vector v $\in Z^m$ such that Av = 0 (mod q) where $\| v \| \leq \beta$ is hard.

4.CERTIFICATELESS PUBLIC AUDITING PROTOCOL IN CLOUD

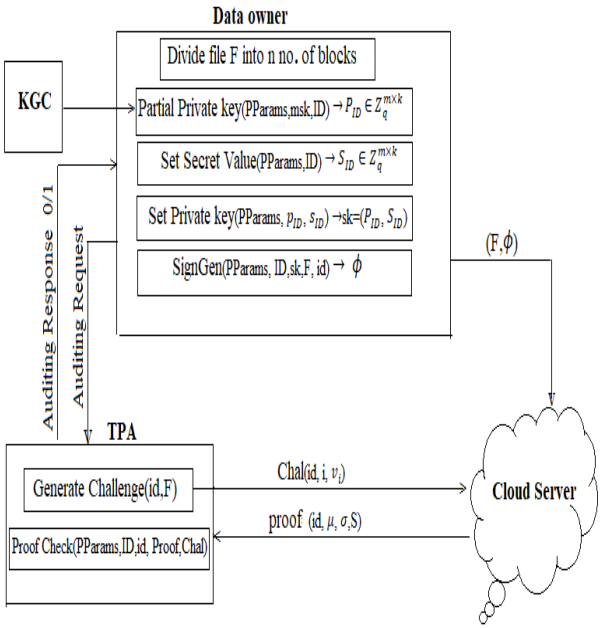

Figure 2. Certificateless Public Auditing Protocol in Cloud

The description of the protocol is as follows: Let n be a security parameter, m > cn log q for a fixed constant c > 0 and q ≥ $\delta\omega$(log n) be a large prime for δ= poly(n).Let s = $\Omega\sqrt{n\ log q}$ be a Gaussian parameter.

Setup(n): Given n as the security parameter, initially the PKG runs TrapGen(n,m,q) algorithm to get a matrix A $\in Z_q^{n\times m}$ along with a short trapdoor basis $\mathbf{T}_A$ of the lattice $L_q^{\perp}$(A).The cloud server also runs the TrapGen(n,m,q) to generate a matrix B$\in Z_q^{n\times m}$ with a short trapdoor basis T_B.Then PKG chooses three secure hash functions, H_1 : $\{0,1\}^* \rightarrow Z_q^m$, H_2: $\{0,1\}^* \times Z_q^{n\times m} \rightarrow Z_q^m$and H_3: $\{0,1\}^* \times Z_q^{n\times m} \rightarrow Z_q^n$ .The system public parameters are PParams={A,B, H_1,H_2,H_3 } and keeps $\mathbf{T}_A$ as the master secrete key i.e. msk= $\mathbf{T}_A$.

Extract-Partial-Privatekey(PParams,msk, ID): On input the system public parameters PParams, the master secrete key and the user ID, PKG executes the Samplebasis(A, $\mathbf{T}_A$, s, H_1(ID)) algorithm to get a matrix $P_{ID} \in Z_q^{m\times k}$andsend it to the data owner and then he sets his partial private key as p_{ID}=P_{ID}.

Set-Secrete-value (PParams, ID): Given PParams and user ID, the user chooses a random matrix $S_{ID} \in Z_q^{m\times k}$ (which must satisfy $\| S_{ID} \| \leq$ b, where b is a positive integer) and set it as his secrete value s_{ID} = S_{ID}.

Set-Privatekey(PParams,p_{ID},s_{ID}):It takes the public parameters and users partial privatekey and secrete value as the input and the data owner computes his full private key sk=(P_{ID},S_{ID})

Set-Publickey(PParams,s_{ID}):Given the public parameters and user secrete value as input the user computes his public key Pk=AS_{ID}.

SignGen(PParams, ID,sk,F, id): To store the data file F in the cloud, data owner divides F into l blocks f_1,f_2,……,f_l , where $f_i \in Z_q^m$ and id $\in\{0,1\}^*$ is the identity of the file F.Then the user runs the following steps to generate the signature for each block of the file.

Step 1: Caluclates $\alpha_j \leftarrow H_2$(ID‖id‖ j) $\in Z_q^m$, where j ≤ n.

Step 2: For each f_i , 1≤ i ≤ l , the user computes $\beta_i \leftarrow H_3$ (id ‖ i) ,where $\beta_i \in Z_q^n$and β_i is the row vector for i=1,2,….l , so the matrix representation of those vectors is , C =(α_1 ,α_2 ,……,α_l) $^T \in Z_q^{l\times n}$

Step 3: Computes the inner products h_{ij}=<β_i, α_j>, for 1≤ i ≤ l and 1≤ j ≤ n. where h_i =(h_{i1}, h_{12},….,h_{in}). So h_i =Cβ_i.

Step 4: To generate the signature for each block user runs sample preimage algorithm. i.e.

σ_i = Samplepre(A,sk,h_i,s).

Step 5: Let us assume that $\phi = \{\sigma_1, \sigma_2, \ldots\ldots, \sigma_l\}$, now the cloud user sends the File F along with the signatures set ϕ to the cloud server and deletes a file locally.

Challenge(id, F): To check the auditing for the file F, the data owner sends the auditing request to the TPA. Then TPA chooses a subset I of the set [1, l] to be I=$\{c_i\}$ where 1≤ i ≤ r. For each i∈ I, a random value $v_i \in Z_q$is choosen by the TPA and then he will compute the challenge message chal={id, i ,v_i} where i∈ I, and is forwarded to the server.

ProofGen(PParams, F, ϕ,chal,id): Upon receiving the challenge , the cloud server chooses the data blocks and coreesponding signatures in it. Now cloud server will compute aggregation of both signatures and data blocks as follows.

Step 1: The data blocks and signatures aggregation is done by the cloud server as follows:

$$\mu^1 = \sum_{i=c_1}^{c_r} v_i f_i \mod q \text{ ,where } \mu^1 \in Z_q^m$$

$$\sigma = \sum_{i=c_1}^{c_r} v_i \sigma_i \mod q \text{ ,where } \sigma \in Z_q^m$$

Step 2: Then, cloud server chooses a random vector w∈ Z_q^m and verifies wether ‖ w ‖ ≤ β or not, if it does not holds again the cloud server chooses it until ‖ w ‖ ≤ β.

Step 3: The cloud server computes S= Bw mod q and $\gamma = H_4$ (S)

Step 4: Then, the linear combination of μ^1 and r can be written as $\mu = \gamma\, \mu^1$ + w mod q.

Step 5: Finally, cloud server sends proof=(id, μ, σ,S) to the TPA as a proof.

Proofcheck(PParams, ID, id, Proof, Chal): Upon receiving the proof TPA calculates $\gamma = H_4$(S) and verifies whether the equation γ . A σ +S = B. μmod q holds or not. If it holds TPA accepts the proof otherwise proof is invalid.

5.SECURITY ANALYSIS

Here we discuss the security concerning to the correctness for our proposed protocol.

5.1.Correctness:

If both cloud server and TPA runs the protocol honestly, then the proof of the cloud server must pass the verification successfully. The correctness of the protocol can be described as follows.

$$\gamma.\text{A}\,\sigma\text{+S} = \gamma.\text{A}(\sum_{i=c_1}^{c_r} v_i f_i)\text{+S}$$

$$= \gamma.(\sum_{i=c_1}^{c_r} v_i A\,\sigma_i)\text{+S}$$

$$= \gamma(\sum_{i=c_1}^{c_r} v_i B\, f_i)\text{+Bw mod q}$$

$$= \gamma\ (\text{B}\ \mu^1)\text{+Bw mod q}$$

$$= \text{B}\ (\gamma\ \mu^1 + \text{w})$$

$$= \text{B}\ \mu \text{mod q}$$

6.PERFORMANCE EVALUATION

6.1.Computation Cost

To evaluate the performance of our work, we used NTRU-CRYPTO library tested on windows 3230M system with 2.60 GHz Intel I3 processor with 4 GB RAM. We assume that the file F of size 4 GB is divided into

l =2,00,000 blocks, | n|=40 bits and |q| = 80 bits.

Our Protocol is designed using matrix-matrix or matrix-vector multiplications only, where the existing protocols have been designed using pairing and exponential operations. The comparison of our protocol with some existing protocols are as shown in below table1.

Table 1 Comparision of Computation cost (in milliseconds)

Signature Type / Property	PKI based Signatures	Identity-Based Signatures	Certificateless Signatures
SignGen	l [h]+ l[e]	3l[e]+ l[h]	n[h]+ l n[mv]+ l [spf]
ProofGen	[h]+ r [e]	2[h]+2[p]+r [e]	[h]+[mv]
ProofVerify	2[p]+ r[e]	3[e]+r[h]	[h]+ 2[mv]

The Comparision for SignGen is shown in Figure 3, and the Comparision for ProofGen is shown in Figure 4, and the Comparision for ProofVerify is shown in Figure 5 respectively.

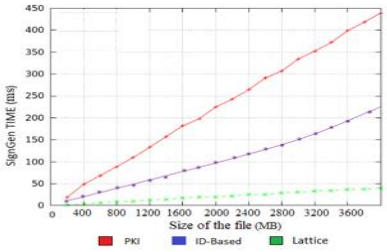

Figure 3: Computation cost comparision for SigGen among PKI vs ID-based vs Lattice-based protocols

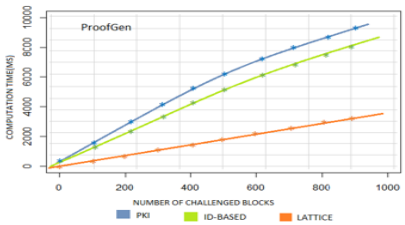

Figure 4: Computation cost comparison for ProofGen among PKI vs ID-based vs Lattice-based protocols

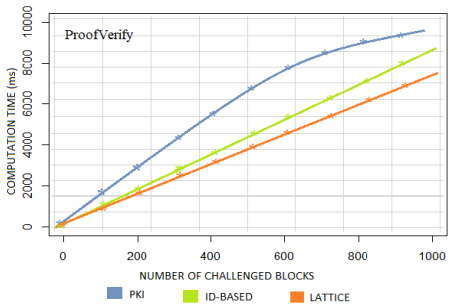

Figure 5: Computation cost comparison for Proofverify among PKI vs ID-based vs Lattice-based protocols

6.2.Communication cost

Initially data owner makes a auditing request to the verifier, then verifier submits the challenge information challenge ={id, i ,v_i} to the server. Then the server responses the proof=(id, μ, σ,S) to the verifier. The verifier validates and will forward the auditing report to the data owner. Thus the communication cost is $3lq$. Here $lq$is the number of bits required to represent element Zq.The communiation cost for our protocol is same as that of previous protocols.

7. CONCLUSION

This paper proposes the certificateless data auditing protocol by using Lattices. In this public verifier is able to check the data integrity in the cloud by eradicating the certificate management problem with PKI based protocols in the previous works.We also proved the security of our protocol using SIS assumption in Lattices and our experimental results show that the protocol is efficient.

REFERENCES

[1] Yong Yu, Man Ho Au*, Giuseppe Ateniese, Xinyi Huang, illy S, Yuanshun D, Geyong "Identity based remote data integrity checking with perfect data privacy for cloud storage" IEEE Transactions on Information Forensics and security

[2] Wang C, Wang Q, Ren K, Lou WJ."Privacy-preserving public auditing for data storage security in cloud computing". In: Proceedings of IEEE INFOCOM.San Diego, CA: 2010.P.1-9.

[3] Yang K, Jia X. "Data storage auditing service in cloud computing: challenges, methods and opportunities". World Wide Web 2012a.

[4] Boyang wang, Boachun Li, Hui Li , Fenghua Li "Certificateless public Auditing for cloud storage", 2013 IEEE Conference on Communications and Network Security (CNS).

[5] Al-Riyami, S.S. and Paterson, K. G. (2003) 'Certificateless Public Key Cryptography', in the Proceedings of ASIACRYPT 2003. Springer-Verlag, pp. 452–473.

[6] David, C., Hofheinz, and Kiltz, E. (2009)'How to Delegate a Lattice Basis, [J]. IACR Cryptology ePrint Archive, pp: 351-362.

[7] Wang H. (2015) 'Identity-based distributed provable data possession in multi-cloud storage'IEEE Trans. on Service Computing, Vol.8, no.2,pp. 328–340.

[8] Erway, C. , Kupcu, A., Papamanthou, C. and Tamassia,R. (2009) 'Dynamic Provable Data Possession',in the Proceedings of ACM CCS, pp. 213–222.

[9] Wang,B., Li,H. and Li ,M. (2013) 'Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics', in the Proceedings of IEEE ICC 2013, pp.62-74.

[10] Zhang, J. and Dong, Q. (2016) 'Efficient id-based public auditing for the outsourced data in cloud storage', Information Sciences pp.

[11] Wang,H.,Wu, Q.,Qin, B. and Domingo-Ferrer J.(2014) 'Identity-based remote data possession checking in public clouds'. IETInf.Security, Vol.8 no.2, pp.114-121.

[12] Gentry, C., Peikert, C. and Vaikuntanathan V. (2008) 'Trapdoors for hard lattices and new cryptographic constructions', Proceedings of the 40th annual ACM symposium on Theory of computing. ACM, pp: 197-206.