2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) | 978-1-7281-5791-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICAECT49130.2021.9392476

Identity-Based Integrity Verification (IBIV) Protocol for Cloud Data Storage

Indra Kumar Sahu and Manisha J Nene

Dept. of Computer Science &Engineering, Defence Institute of Advanced Technology, Pune, India
E-mail: ikambition.s23@gmail.com mjnene@diat.ac.in

Abstract: **— With meteoric advancement in quantum computing, the traditional data integrity verifying schemes are no longer safe for cloud data storage. A large number of the current techniques are dependent on expensive Public Key Infrastructure (PKI). They cost computationally and communicationally heavy for verification which do not stand with the advantages when quantum computing techniques are applied. Hence, a quantum safe and efficient integrity verification protocol is a research hotspot. Lattice-based signature constructions involve matrix-matrix or matrix vector multiplications making computation competent, simple and resistant to quantum computer attacks. Study in this paper uses Bloom Filter which offers high efficiency in query and search operations. Further, we propose an Identity-Based Integrity Verification (IBIV) protocol for cloud storage from Lattice and Bloom filter. We focus on security against attacks from Cloud Service Provider (CSP), data privacy attacks against Third Party Auditor (TPA) and improvement in efficiency.**

Keywords—Lattice cryptography, SIS problem, IDBased cryptography, Cloud storage system, Bloom filter, Quantum Safe Cryptography.

I. INTRODUCTION

With the fast and paradigm advancement in computer and information technology, the traditional way of computing has changed drastically. Cloud computing provides a cost effective and convenient hardware, software and other services based on customer's demand. Cloud storage is one such service that has gained prime importance with the huge amount of data explosion in the environment. Cloud storage serves people saving their lot of memory space, time and efforts. Cloud users can choose to upload or retrieve their data at will by paying for the service to the cloud service provider. With such convenient and appealing features common users, companies and industry people have started utilizing these services for the data in the cloud. AWS from Amazon, Azure from Microsoft, Google Cloud Platform, Ali Baba and IBM clouds are some of the major players in the cloud storage industry.

Although there are numerous merits in the cloud storage, however, there are some challenges in the cloud storage. Amongst others, integrity has come on focus as an important challenge. Integrity of the data stored is challenged by many means such as data theft, tampering, modification or even deletion. The causes of integrity issue are due to hardware or software problems, operational and computational errors and the adversaries, which limits development of cloud storage [1]. This compels the cloud users to check the integrity of their data stored on the cloud.

Verification of integrity raises question as to how it can be checked? There are methods like downloading whole of the data stored and then carry out the process by using traditional integrity verification techniques like the message authentication code, hash function, digital signature and hashbased message authentication. But existing limitations of bandwidth and computational power for such enormous amount of data, make this idea undoable. With the explosive increase in the amount of data, it is very difficult and computationally costly affair to query and search efficiently. So, improvement in efficiency of the integrity verifying techniques is a major challenge. In 1970, Howard Bloom proposed the Bloom Filter. It is a probabilistic data structure, with space efficient, designed to check presence of an element in a set of a predefined vector in the binary form. It requires the data structure to search efficiently.

It tells us that the element either is not in the set with full guarantee or may be in the set with some false positives. In the Bloom filter, the requirement of the memory space is very less as an element needs only a few bits as compared to the traditional hash table or an array used for such operations. Bloom Filter has brought focus for research and thesis work due to continuous increase and expansion of data and various developments which need memory efficient and fast computations. Additionally, the Bloom Filter can be employed for integrity verification of the data stored on cloud. The study in this paper proposes a protocol for integrity verification comprising of the characteristics of both Lattice theory and Bloom filter.

The organization of paper is given as under. With introduction in the first section, we start the paper. The related work on the subject carried out so far is presented in the second section. Preliminaries followed by the proposed IBIV protocol are described in the third and the fourth section respectively. The security analysis of the protocol is mentioned in the fifth section. The last section concludes the proposed work.

II. Related Work

The cloud storage as one of the cloud-based services, is based on the saving time, memory space, minimal management and on the go services. The problem of integrity checking in the cloud was first addressed in 2007, when Ateniese et al. [2] proposed the first Provable Data Possession (PDP) scheme. Figure 1. shows the PDP based existing scheme. His scheme proved to be quite effective for verification of data integrity based on the RSA assumption. However, large modulo exponents and mathematical calculations could not let it be popular by restricting in terms of efficiency for the cloud data. The first of this kind, it opened new ways to find better and efficient schemes for data integrity checking from research communities and scholars. Curtmola et al. [3] proposed another PDP scheme and later proof of retrievability (PoR). In 2008, Ateniese et al [4] proposed new scheme based on hash functions and symmetric key encryptions that was a dynamic PDP scheme. Following the research work, many schemes were proposed based on the concept. Later, in the same year, Shacham et al [5] proposed PoR with additional capability to recover lost data based on homomorphic authenticators.

The existing techniques are based on the difficult problems such as complex mathematics, number theory and discrete logarithm problem for data integrity checking schemes. The quantum mechanics enables security in communication and network security systems [6] [7]. The Lattice based constructions, from one of the NIST shortlisted quantum safe techniques, is based on hard problems and got popularity due to its one way signature constructions. Wang et al. [8] proposed public verification method and dynamic storage in the cloud computing security. The scheme failed to prove safe against quantum attacks.

Identity based integrity verification received high attention from researchers and scholars due it its simple, easy and cost-effective output. Due to this, numerous schemes based on identity for cloud security have been proposed [9] [10]. Liu et al. [11] proposed a protocol, that satisfy many security requirements such as security against attacks from the CSP and TPA. Public key infrastructure (PKI) is backbone of the most of the cloud data integrity or remote data integrity verification constructions that burdens the users with the complex key management procedures such as generation of certificates, updating and revocation. These are expensive and time taking procedures [12]. To simplify certificate management, Zhao et al. [13] proposed the first identity-based PDP scheme followed by Yu et al. [14] which was based on key-homomorphic cryptographic primitive for remote data integrity checking protocol that provides security proofs in detail.

A. Contributions

In this paper an IBIV protocol using lattice based cryptographic constructions and Bloom filter is proposed. The protocol can improve efficiency, eliminate numerous certificate management procedures and reduces the risk from quantum attacks in post quantum era. SIS assumption based on hardness of the lattice cryptography is the main basis of the proposed protocol. Security against the CSP and TPA attacks is provided by our protocol.

III. Preliminaries

A. Lattice

Definition:

Let r_1 , r_2, …rn ϵ Z_n, be n linearly independent vectors, such that set R = { r_1 , r_2, …r_n}, then Lattice Λ =L (r^1 , r_2, …rn) = { Σ airi : ai ϵ Z } is generated by R. The vectors r_1 , r_2, …r_n are called the basis of the lattice. The lattice so generated can be represented in two ways as:

- Λ (R) ={ X ϵ Z_m | y= R, r mod q, r ϵ Z_n}
- $\Lambda\perp$(R)={ X ϵ Z_m | R^T X = 0 mod q}

The ranks are defined as super rank if m < n, reduced rank if m > n and full rank if m = n, where m is dimension and n is rank [15]. *B.* Lattice Problem:

The hardness of the SIS assumption in the lattice is the basis of security of the lattice-based signature scheme. Small

Integer Solution (SIS) problem can be given as:

In a matrix Rϵ Z_q^{mxn} and security parameters n, q, m, β, the SIS m,n,q,β is defined to find a nonzero integer vector r ϵ *Z*m such that ||r||$\leq$ β and Rr = 0 (mod q)[16].

C. Bloom Filter

Bloom Filter: Bloom filter is a special, unique data structure. It is space-efficient, simple and randomized data structure. A set of elements as an array data structure is used. This makes the bloom filter short and concise with set U of n elements. The filter allows to check presence of an arbitrary element in the set through it rather than directly. Very less memory space required to store bits, makes it space efficient. This may give false positives with very small and manageable probability [17].

D. Framework of the proposed Protocol

Once the cloud data owner uploads the data on cloud server, he needs to check the integrity of the data from time to time. For that he inscribes himself with the cloud service provider and manages his account with him. The data owner cannot check the integrity himself for the reasons like limited management resources, hardware limitations, lack of operational knowledge and other physical and technical limitations that brings the third

party auditor in the scene. The proposed protocol is designed with four entities, as shown in figure 1 below.

Cloud user is an individual, institute or a company with large data for storage on cloud server. He uploads the data files on the cloud server. The Third Party Auditor (TPA) is delegated with the responsibility of auditing or verification of the data integrity. He can get their own data from the cloud at any desired time. He needs not to worry about management, safety and security of the data stored on the cloud.

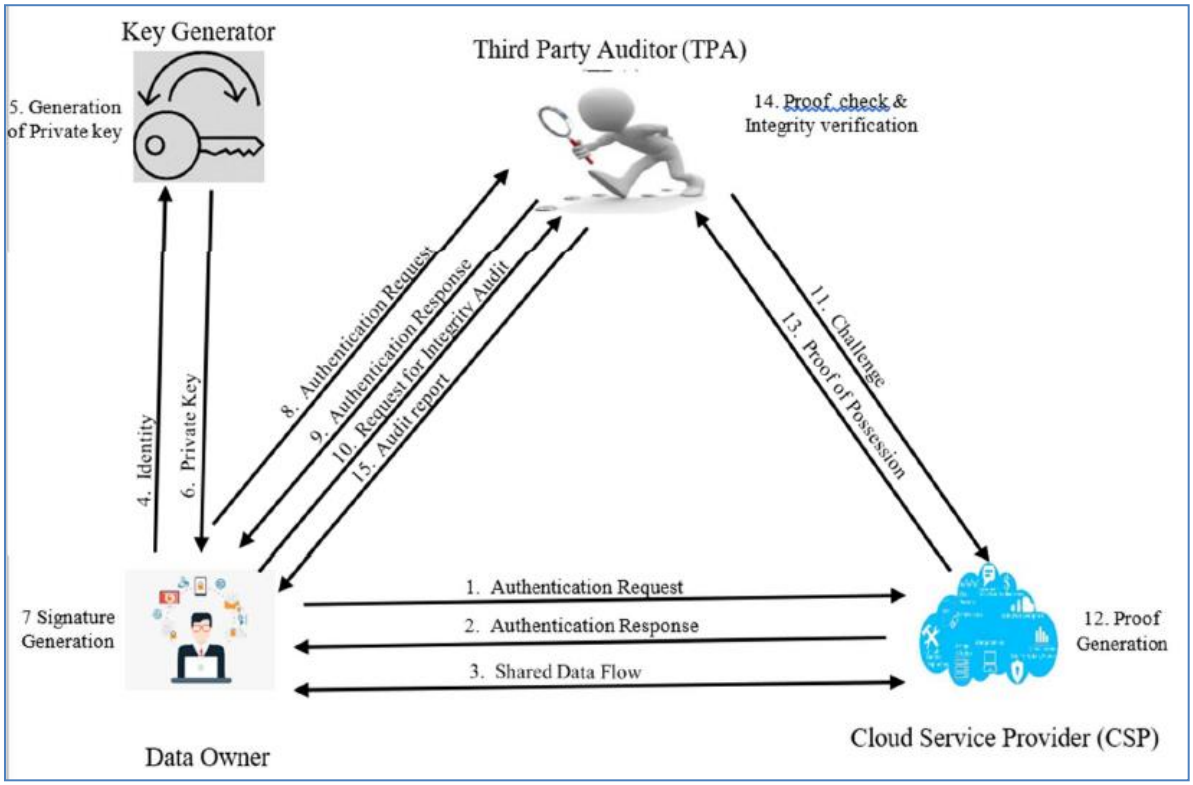

Figure 1. Identity based integrity checking protocol Framework

Key Generator is to serve the cloud data owner to generate private key based on his identity.
*Cloud Service Provider (*CSP*)* is the one who provides cloud servers for data storage, having enormous data storage facility resources and computational capabilities. He may not be reliable at times and may attack on the integrity of the data stored.

*Third Party Auditor (*TPA*)* is a reliable and trusted party to whom the responsibility of verification of integrity is delegated. He is a trusted one to assess the reliability of the cloud server.

The proposed protocol is designed to achieve the following objectives:

- Security against attacks that a CSP can launch on the stored data.
- Security against threat of attacks that TPA can launch.
- Improvement in querying and search efficiency.

The proposed protocol is having seven algorithms: (Setup, KeyGen, *Sig_Gen*, DataPre, Challenge, Gen_Proof and Verify_Proof). *Setup(1n):* The probabilistic algorithm for setup is run. The setup is carried out with input as n and outputs as the model variables modvar, master key Mk and public key Pk. *KeyGen* (modvar, msk, *id*)**:** The security parameter modvar, the master secret key Mk and Identity Idϵ {0,1}$*$ are given as input and secret key *skid* corresponding to the identity is output. This is a probabilistic algorithm run by the key generator using Basis Delegation Algorithm. *Sig_Gen* (modvar, L, skid, F, *id*): The data owner runs signature generation algorithm. With identity Id, this probabilistic algorithm, is run to generate the signature. The data file F is divided into L blocks namely, u1, u2, …uL. The algorithm takes the model variables modvar, the secret key *skid* along with data file F ϵ {0, 1}$*$ as input in this, and outputs are the signatures e1, e2 . . . , eL of each file block u1,u2, … uL. The data F along with these signatures are to be stored on the cloud server.

DataPre (φ, X[n]): Data for Bloom Filter is prepared to be fed based on the signature set φ and hash function. Any given arbitrary data array X[n] is produced as values 0/1.*Challenge* (L,F,id): The TPA runs a randomized

algorithm to generate the challenge chlg on audit request from the data owner. The model variables modvar, user's identity Id, and a level L ϵ {0, 1}∗ defined as data identifier, are inputs and challenge chlg becomes outputs. *Gen_Proof* (modvar, chlg, F, φ, id): With the system modvar, the challenge chlg, identity Id, signature set φ, the file F fed as input on the probabilistic algorithm while running by the cloud server. Its output is data proof of possession P against the challenged data blocks. *Verify_Proof* (modvar, P, chlg, id, L): In this algorithm, the third party auditor runs the deterministic algorithm rather than probabilistic one. This is to verify the proof received from CSP against the challenge. The algorithm process or computes the proof of possessions by comparing with the existing data on bloom filter and other specified conditions. The modvar, chlg, Id, proof of possession P become input, and the result in the form of 0 or 1 become output that indicate the integrity status of the data block challenged.

IV. IBIV PROTOCOL

The proposed Identity Based Integrity Verification (IBIV) protocol for cloud data storage using Lattice and Bloom filter is based on a homomorphic signature scheme from lattices and bloom filter. The role of the trusted third party auditor (TPA) is of expert in verification job who ensures his availability at user's request for integrity verification is very important. The proposed protocol is described in details in the following paras: In the Setup step, Trap Door Generation (TDGen) function generates Mk. In the key generation, the secret key Sk is extracted from user's identity Id by running the basis delegation algorithm. In the Sig_Gen step, SIS based homomorphic signature is generated by data owner. In DataPre, in accordance with the signature set, the TPA builds MHT and maps the corresponding values to vector in Bloom Filter. TPA generates chlg in challenge step and sends it to the service provider for proof of possession. In the Gen_Proof phase, post receiving the chlg, the service provider obtains the proof from the data blocks, signature and randomly selected vector. Then he sends it back to the TPA. In Verify_Proof, integrity of the data block is verified. Data privacy-preserving is ensured at the time of construction. TPA is never allowed to get any information about the user's data stored on cloud. The details of protocol are as follows:

Setup(1n): TDGen function is used to generate master secret key Mk for the data owner. The security parameters n,m, q and β are chosen in such a way that integer $m \geq 2n \log q$, $q \geq \beta\omega(\log n)$ is a large prime and a real number β= poly(n). The protocol uses following hash functions:

The Data owner runs TDGen function on the security parameters (n, m, q) to generate a matrix $\mathbf{D} \in Z_q^{m\times n}$ where its basis $\mathbf{T_D}$ is from $\in Z_q^{m\times m}$. Similarly, CSP runs TDGen on the security parameters (n, m, q) to generate a $\mathbf{S} \in Z_q^{m\times n}$ where its basis **Ts** is from $\in Z_q^{m\times m}$. Post above actions, the model variable *modvar* is declared as (**D**, **S**, H, H_1, H_2, H_3). The Data owner's master key is $M_k = T_D$.

KeyGen (modvar, msk, id): Basis Delegation Algorithm is run to generate private secret key sk in accordance with owner's Id by the key_gen. Based on input compute the following:

- Calculate M = H(Id) $\in Z_q^{m\times m}$, where M is an invertible matrix.
- Run the BasisDel (**D**,**M**, T_A, s) algorithm to generate *Skid* = $\mathbf{T_{Id}}$ which becomes owner's private key.

Sig_Gen (modvar, L, sk_{id},F, Id):): Signature is generated for data owner's data file using SIS difficult problem. The data file F,is divided into L blocks ($\mathbf{u}_1$, $\mathbf{u}_2$,….. , $\mathbf{u}_L$) where $u_i \in Z_q^m$ blocks before it is uploaded to store in the cloud server. Given the Data owner's private key T_{Id},

- Calculate $\mathbf{M} = H(Id)$, and $\mathbf{B} = \mathbf{DM}^{-1}$, and calculate α_j (j ≤ n), $\alpha_j = H_3$ (**B**||L ||j) $\in Z_q^m$.For each $\mathbf{u}_i$, $1 \leq i \leq L$, the Data owner computes $\lambda_i \in Z_q^m$: $\lambda_i = H_1$ (L ||i) + $\mathbf{Su}_i$.
- Compute the inner products $h_{ij} = \langle \lambda_i , \alpha_j \rangle$, $1 \leq i \leq L$, $1 \leq j \leq n$. $\mathbf{h}_i = (h_{i1} , h_{i2} , \ldots , h_{in})$. Let $\mathbf{C} = (\alpha_1 , \ldots , \alpha_j)$, $\mathbf{h}_i = \mathbf{C}\lambda_i$
- Cloud Data owner runs SamplePre algorithm to gain signature of block, e_i = SamplePre (**B**, T_{Id}, h_i , σ).
- Let φ = (e_1 , . . . , e_L). The Data owner uploads the file F = ($\mathbf{u}_1$, $\mathbf{u}_2$, ….. $\mathbf{u}_L$) where $u_i \in Z_q^m$ and corresponding signatures e_1 , . . . , e_L to CSP, signatures to the TPA. After that, he deletes the original data at his end.

DataPre (φ, $X_{[n]}$): Data for Bloom Filter is prepared to be fed based on the signature set φ and hash functions. Any given arbitrary data array X[n] is produced with calculated values 0/1. This is used in the verifying step of the protocol.

Challenge (L,F): TPA runs a randomized algorithm to generate challenge chlg to check the integrity of F = (u_1 , u_2 ⋯ ⋯ , u_L) on Data owner ' s request for auditing. Subsequently, the TPA defines the subset of set [1, L] to be v= {s }(1 ⩽ j ⩽ θ) and s1 ⩽ . . . ⩽ s θ . For each element i ∈V, the TPA chooses a random $c_i \leftarrow$ Zq using L, F and i ∈V, the TPA chooses a random $c_i \leftarrow$ Zq using L, F and generates chlg = {L, i, c},i ∈ v as the challenge. TPA forward chlg to the CSP for checking integrity. The model variables modvar, user's identity Id, and a level L ϵ {0, 1}* defined as data identifier, are inputs and challenge chlg becomes output.

Gen_Proof (modvar, chlg,F, φ, Id): This step is carried out by CSP. Upon receiving chlg {L, i, $\mathbf{c}_i$ }i ϵ v (v being the sub block from L data blocks), the data blocks {u_i}i ϵ v and the {e_i}i ϵ v are chosen. Then, The CSP calculates $\mathbf{u_c} = c_i \mathbf{u_i}$,

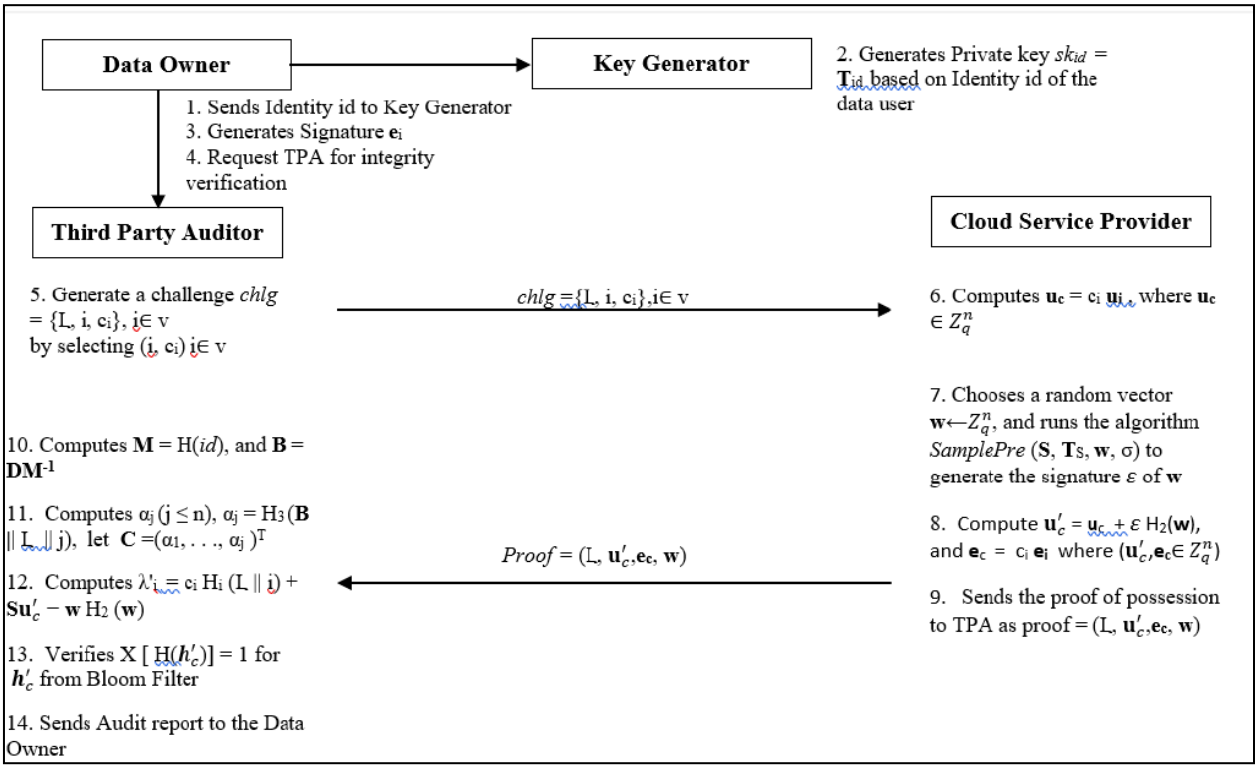

Figure 2. IBIV protocol from Lattice and Bloom Filter

chooses a random vector $\mathbf{w} \leftarrow Z_q^n$, to generate the unique signature ε of **w** runs the pre image sampling algo on (**S**, T_S,**w**, σ). Then he calculates $\mathbf{u}'_c = u_c + \varepsilon H_2(\mathbf{w})$, and $\mathbf{e}_c = c_i \mathbf{e}_i$ where $(\mathbf{u}'_c, \mathbf{e}_c \in Z_q^m)$ and sends proof = (L, $\mathbf{u}'_c$,$\mathbf{e}_c$, **w)** to TPA the proof information.

Verify_Proof (modvar, P, chlg, id, L): Once the proof = (L, $\mathbf{u}'_c$,$\mathbf{e}_c$, **w)** from CSP received , the TPA computes the following:

- Calculate **M** = H(Id), and **B** = $\mathbf{DM}^{-1}$
- Calculate α_j (j ≤ n), $\alpha_j = H_3$ (**B** ‖ L ‖ j), let **C** = $(\alpha_1, \ldots, \alpha_j)^T$
- Calculate $\lambda'_i = c_i H_i (L \| i) + \mathbf{Q}u'_c - \mathbf{w} H_2(\mathbf{w})$
- Calculate $\boldsymbol{h}'_c = \mathbf{C}\, \lambda'_c$

Finally, TPA compares $\boldsymbol{h}'_c$ mod q with the pre-fed values in the Bloom Filter, and $\|e_c\| \leq \theta\sigma\sqrt{m}$. If both the conditions are met integrity is maintained else Not.

V. Security Analysis

The proposed IBIV protocol is lattice based cryptographic constructions and uses Bloom filter. Analysis of the protocol exhibits that it improves efficiency, eliminates certificate management and is safe against quantum computer attacks. The proposed protocol is based on the hardness of SIS assumption and provides protection against the attacks from CSP and the TPA attacks. The efficiency of in querying and searching of data file for verification of integrity and the cloud storage space utilization are enhanced as the operation is in the vector space for Lattice and Bloom Filter. The protocol demonstrates its merit to suit for data owners to verify the integrity of the cloud data.

A. Data Privacy Against TPA Attacks

TPA cannot retrieve any data block during verification of proof from CSP thus guarantees the privacy. The TPA reads the proof = $(L, \mathbf{u}'_c, \mathbf{e}_c, \mathbf{w})$ from the CSP, where retrieval of any block of the data file F stored in cloud servers is not possible.

Let us consider that the TPA launches attacks in the cloud storage server and tries to recover any block in F = (u_1, u_2 .., u_L). $SIS_{q,m,n,\beta}$ problem assumptions from lattice and GPV signature scheme makes it impossible to retrieve data due to combined $\mathbf{u}'_c = u_c + \varepsilon H_2(\mathbf{w})$, and $\mathbf{e}_c = \Sigma\, c_i \mathbf{e}_i$ where $(\mathbf{u}'_c, \mathbf{e}_c \in Z_q^m)$ and randomly selected vector $\mathbf{w} \leftarrow Z_q^n$.

B. Security Against the CSP

The problem of the SIS based assumptions are not solvable in polynomial time if the trapdoor is not available. So, to gain signature, there is no such probability polynomial algorithm. Having digital signature scheme as non-forgeable in the random oracle model and the hard SIS problem in lattices, it is almost impossible for an adversary to break the security of the proposed ID-Based remote data integrity checking protocol except with very negligible probability. The data

file is divided into smaller blocks and separate signature is generated for each data block independently at the time of signature generation. It can check the known plaintext or ciphertext attack. So, it still cannot obtain sensitive information even if the attacker obtains some of the ciphertext information in the signature generation process. To prevent attacks from CSP, each data block is incorporated with corresponding authenticating factor. In case any of the data block is tampered or modified, the corresponding signature generated earlier will expose the changes. The integrity checks through TPA, can effectively resist the plaintext/ ciphertext attacks.

C. Improvement in Efficiency

Bloom Filter having excellent features and merits in verifying integrity of data files with high efficiency, improves overall efficiency. The generation vectors and auxiliary update vectors are generated for each data block during operation. They lead to the constant time complexity for querying and searching a large number of data files F. It is very efficient with manageable false positives. The less memory space requirements make it better than the other data structures like arrays. With the use of hash function unidirectionality, the Bloom Filter shows certain probability of fault tolerance. The characteristics of the bloom filter combined with lattice signature ensures the protection of the data in this protocol.

VI. CONCLUSION

Identity based integrity verification (IBIV) protocol using lattice based cryptographic constructions and Bloom filter is proposed in this paper. Analysis of the protocol exhibits that it improves efficiency, eliminates numerous certificate management procedures and is safe against quantum computer attacks. The hardness of SIS assumption became the basis of the proposed protocol. It is secure against the attacks from cloud server and the protects data privacy against the third party auditor if he tries to look into it. Vector space operations in lattices and bloom filter ensure the efficiency of in querying and searching of data file for verification of integrity. The space utilization for cloud servers is enhanced. The protocol demonstrates its merit to suit for data owners for the verification of the integrity of the data on cloud servers. Further, research and study may be taken up for the proof of concept using lattice-based signature constructions and bloom filter techniques to enhance the process of integrity verification.

REFERENCES

[1] Armbrust, M., Fox, A., Griffith, R: A view of cloud computing. Commun. ACM 53(4), 50C58 (2010).

[2] Ateniese, G., Burns, R., Curtmola, R.: Provable data possession at untrusted stores. In: Proceedings of CCS, pp. 598C609 (2007).

[3] Curtmola, R., Khan, O., Burns, R., Ateniese, G.: MR-PDP:multiplereplica provable data possession. In: Proceedings of ICDCS, pp. 411C420 (2008).

[4] Ateniese, G., Pietro, R.D., Mancini, L.V., Tsudik, G.: Scalable and efficient provable data possession. In: Proceedings of SecureComm, pp. 1C10 (2008).

[5] H. Shacham and B. Waters, “Compact proofs of retrievability”, in Proc. Adv. Cryptol. ASIACRYPT, 2008, pp. 90C107.

[6] Sharma, Mohit Kr, and Manisha J. Nene. "Two-factor authentication using biometric based quantum operations." Security and Privacy 3.3 (2020): e102.

[7] Upadhyay, Gaurav, and Manisha J. Nene. "One time pad generation using quantum superposition states." 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, 2016.

[8] Wang Q , Wang C , Ren K , et al. Enabling public auditability and data dynam- ics for storage security in cloud computing. IEEE Trans Parallel Distrib Syst 2011;22(5):847–59.

[9] C. Gentry, C. Peikert, V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions”, Proceedings of the 40th annual ACM Symposium on Theory of Computing (STOC 2008), Victoria, British Columbia, Canada, pp.197C206, 2008.

[10] D. Boneh, D.M. Freeman, Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures, Proceedings of PKC 2011, LNCS 6571, Berlin: Springer-Verlag, pp.1C16, 2011.

[11] H. Liu and W. Cao, Public proof of cloud storage from lattice assumption, Chinese Journal of Electronics, Vol.23, No.1, pp.186C190, 2014.

[12] Liu, Zhangyun, et al. "Identity-based remote data integrity checking of cloud storage from lattices." 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM). IEEE, 2017.

[13] J. Zhao, C. Xu, F. Li, and W. Zhang, Identity-based public verification with privacy-preserving for data storage security in cloud computing, IEICE Trans. Fundam. Electron., Commun. Comput. Sci., vol. E96-A, pp. 2709C2716, Dec. 2013.

[14] Zhao X, Wang X, Xu H , et al. Cloud data integrity checking protocol from lattice. Int J High Perform Comput Networking 2015;8(2):167.

[15] Wang, FengHe, YuPu Hu, and BaoCang Wang. "Lattice-based linearly homomorphic signature scheme over binary field." Science China Information Sciences 56, no. 11 (2013): 1-9.

[16] Wang C, Wang Q, Ren K, et al. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. INFOCOM, 2010 Proceedings IEEE, 2010, 62(2):525–533.

[17] Zhiguang Qin, Shikun Wu, and Hu Xiong.: Strongly Secure and Cost- Effective Certificateless Proxy Re-encryption Scheme for Data Sharing in Cloud Computing. BigCom 2015, LNCS 9196, pp. 205C216, 2015.

[18] Yan, Yunxue, et al. "A dynamic integrity verification scheme of cloud storage data based on lattice and Bloom filter." Journal of information security and applications 39 (2018): 10-18.

[19] Desai, S. Sundeep, and Manisha J. Nene. "Trust Based Security in Battlefield-of-Things." 2019 International Conference on Wireless Networks and Mobile Communications (WINCOM). IEEE, 2019.