International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-01 May 2024
https://ijgst.com.2024.v13.i1.pp116-124

A CONCRETE AND PROVABLE APPROACH FOR SECURING CLOUD DATA USING IDENTITY-BASED DATA POSSESSION MODEL

M VENKATA RAMANA [1] K.ANUSHA [2] CHALLA VINOD [3] Dr.S .GOPI KRISHNA [4]
[1]ASSOC.PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
[2] ASST. PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
[3]ASST. PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
[4] PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
[1,2,3,4] SRI MITTAPALLI COLLEGE OF ENGINEERING

Abstract— With the advances in cloud computing, many business organizations are interested in public cloud to store their data. When one organization acquires a part of business from another organization, then the acquiring organization possess the corresponding data. Here, we have to address two questions regarding outsourcing of data to the organization who had purchased a part of business from acquired organization. Firstly, what about the integrity of remote data purchased by acquiring organization. Secondly, how much cost can be incurred on transferring data to the acquiring organization? Hence outsourcing of data had motivated us to propose a new concept Identity-Based Data Possession Model (IBDPM). By availing the benefits of IBDPM, three security considerations which are to be addressed are: (a) The data which is not purchased by the acquiring organization must be secured. (b) The integrity of the data purchased by the acquiring organization must be checked. (c) The data transfer operations can be maintained by Cloud service provider. Firstly, we will present the architecture of IBDPM and later we will present how security policies are implemented in IBDPM using bilinear pairings. We analyzed that security of the proposed method is really strong, agile and scalable and cannot be breached by unauthorized access. We also analyzed that the proposed method can be used in multitenant architecture. The proposed method can also be implemented for hybrid clouds also.

Index Terms—Identity-Based Data Possession Model, bilinear pairings, multitenant architecture, multi cloud.

I. INTRODUCTION

The delivery of computing services to the client is referred to as cloud computing. The clients can use internet to access the cloud resources like applications, interconnected systems, servers, storage, and services.

The key aspect which is to be considered when outsourcing the data to cloud servers is security. When the business organizations store their huge data on public cloud server (PCS), they have to pay their storage.

Sometimes, a small fault would result in large damage to the stored data say for *e.g.*, technology failure, accidental deletion, overwriting data, *etc*. These unexpected actions may cause severe damage to the clients' business. With the advances in cloud computing, many firms prefer remote public cloud servers to store their data. By sharing cloud resources through internet, business organizations can save money incurred on the data center components like computing resources, network infrastructure, storage infrastructure, data center security, site maintenance, personnel maintenances, etc.

The firms' data which is stored on the cloud might be valuable, sensitive and

confidential. The remote data is not under the control of business organization. These businesses would face significant losses if any data is lost.

Any malicious cloud computing servers will cover their faults to prevent compensating the harmed businesses. It is unfair to businesses. In storing data on untrusted servers, for example, public cloud servers, verifying the secrecy of remote data has become a critical problem.

II RELATED WORK

Many models were proposed to check consistency of cloud data.

Pooja Natu et al., [1] given an overview of current PDP versions by defining models, operations, benefits and disadvantages.

Zhou F et al., [2] suggested an identity-based scheme to remove certificate administration. Their test results demonstrated that the scheme is absolutely correct and strong based on bilinear pairings and Computational Diffie-Hellman problem hardness assumption.

Peng S et al., [3] suggested the first identity-based scheme to significantly minimise computing costs for many owners and many clouds simultaneously.

Prasad B. S. V et al., [4] proposed an alternate halfway resolved data transmission and remote data accuracy inspection paradigm of character-based open key cryptography: character-based intermediate arranged data transmitted what's more, remote data respetability testing transparently cloud (ID-ICBP) with Tate pairings that is better than Diffie Hellman. They also indicated the scheme is potentially secure due to the computational Diffie–Hellman issue's hardness.
schemes.

Shi Y et al., [5] proposed an effective attribute-based encryption scheme that relies on bilinear pairing over Barreto and Naehrig curves. They also built a mixed ABE for lower ciphertext expansion rate when plaintext is massive.

The Tat-pairing algorithm based on a multibase number representation scheme (TP-MBNR-PH) that uses base 1/2, 3, and 5 for a low costs of bilineary pairing, with the TP-MBNR-PH protocol used in decentralised KP-ABE for equating computational effort of encryption to existing dynamic remote data integrity, was proposed by Chandrasekaran B et al[6].
In order to provide external vertification of outsourced knowledge with several replicates without IT, Peng S, et al.[7] suggested a novel multi-replica proven data keeping method (IDPMR-PDP). Peng They also implemented a structured identity-based protection model for multiple PDP identity systems and proved that the IDPMR-PDP is protected against malicious cloud servers and safeguarding privacy against curious verifiers under this model.

Li, C et al.,[8] suggested a complex efficient PDP data integrity authentication system.

The dynamic data integrity protection system suggested by Wadhwa D et al, [9] offers a proof that right data ownership may be made on request from a server. They followed a dynamic approach to data integrity, using the RSA encryption mechanism as a public crypto system tool.

Keerthana, Aetal. [10] suggested that the allocation of data ownership in multi-cloud storage should be based on a scalable identity.

Y. Yu et al.,[11] introduced a new ID-based RDIC protocol by using the key homogeneous primitive encryption to reduce the device complexity and costs associated with the establishment and maintenance of the key mechanism of PKI-based RDIC authentication.

They made the RDIC ID and its security model legitimate, including the

https://ijgst.com.2024.v13.i1.pp116-124

defense of a malicious cloud service and zero information privacy of a third party control system. They also provided a tangible construction of an RDIC ID-based framework that does not leak any information to the verifier during the RDIC process from stored files.

III PROPOSED METHOD

Bilinear Pairing

A bilinear pairing maps a pair of group elements to another group element. Specifically, let G_1 , G_2 be two cyclic groups of order p. g_1 and g_2 denote generators of G_1 and G_2 respectively.

A function $e : G_1 \times G_1 \rightarrow G_2$ is called a bilinear pairing if it has the following properties:

Bilinearity. For all u, v ∈ G_1 and x, y ∈ Z_p, $e(u^x , v^y) = e(u, v)^{xy}$ holds.

Non-Degeneracy. $e(g,g) \neq 1_{G2}$ where 1_{G2} is the identity element of G_2 .

Efficient Computation. e(u,v) can be computed efficiently (in polynomial time) for all u, v ∈ G_1.

Data owners themselves will typically track the security of their cloud data by executing a two-party protocol for remote data integration control. The audit outcome, however may be considerable in a two-party situation either as the data owner or the cloud server.

The public-verifiable remote data integrity management protocols allow any person to check the integrity of the outsourced data. We presume that a third-party auditor (TPA), whose experience and capacity to perform inspection work exits, would identify the publicly verifiable Remote Data Integrity Checking protocols. In this case, the ID based paradigm of data ownership is seen in Figure 1.

In the framework are participating four separate organizations namely the trustworthy Key Distributed Center (KGC), the cloud customer, the cloud server and the TPA. The KGC provides hidden keys according to identities for all users. The Cloud customer requires vast amounts of files to store without needing a local copy on the cloud, and the cloud server has large storage and processing capacity and provides cloud customers with data storage facilities.

On behalf of cloud utilisers, TPA has specialist know-how and skills and is secure in testing the integrity of the cloud data on request. Each organisation has its own responsibilities and advantages. The cloud server can be self-interested and the cloud server can also decide to cover corruption events for cloud customers for its own benefits such as keeping a positive image.

We presume, however, that because of regulations and financial rewards the cloud server has none to disclose the TPA hosting details. The TPA's role is to conduct data integrity controls for the cloud customer, but TPA is also curious that during the data integrity management process it is able to learn any user data information.

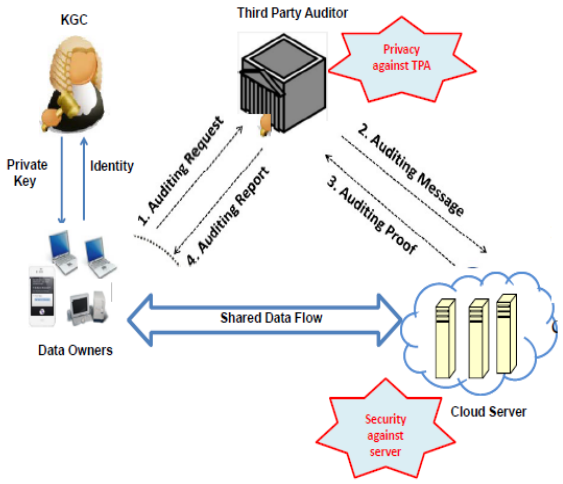

Figure 1: The system model of identity-based Data Possession Model

System Components and its Security

https://ijgst.com.2024.v13.i1.pp116-124

Six algorithms namely Setup, Extract, TagGen, Challenge, ProofGen and ProofCheck are involved in an identity-based RDIC system.

Setup(1^k) is an algorithm of probabilism run by the KGC. The protection parameter 'k' is used to input and exit 'param' function parameters and 'msk.'

Extract(param, msk, ID) This is a probabilistic KGC algorithm. It takes system parameters 'param,' 'msk' and user ID {0, 1}* as input, the hidden key skid, referring to the identity id, output. The secret key is the ID of the user.

TagGen(param, F, sk_{ID}) This is a probabilistic algorithm operated by the ID data user. The systems 'param' parameters, the hidden user key, file file F {0, 1}*, and the tags − = (4-01,.., μn) of each block file mi that will be stored in the cloud along with file F are needed for the saving of the input file..

Challenge(param, F_n, ID) is a TPA running randomised algorithm. It takes 'Param' device parameters, the ID of the data holder and a special Fn file name as input to release the 'chaal' challenge for the file called Fn on behalf of the user ID.

ProofGen(param, ID, chal, F, σ) This is a public cloud maximum likelihood process. The 'P' involves comparing 'Param,' the 'chal' challenge and the information keeping ID, tag¨, file f' and Fn title are being used to demonstrate the 'P' data property evidence to the frames in query.

ProofCheck(param, ID, chal, P, F_n) Is a TPA deterministic algorithm. 'param' system parameters for outputs1 and 0 shall be required to signify if File F remains intact, as well as 'chal' difficulties (chal, data owner ID), the name Fn and the alleged proof of input data ownership of P.

Anyone who has access to the signatory identification will search the symbol's signature in an ID-based signature system. Similarly, someone who knows the identity of a cloud customer is able to verify data integrity on their behalf in ID-based RDIC protocols, a third-party auditor (TPA) says.

Therefore, public verification in RDIC, particularly for resources-limited cloud users, is more preferable than private verification. Public verification. In this case, for data secrecy in ID-based RDIC protocols the property of zero-known privacy is extremely significant.

Our first contribution is to formalise, in ID-based RDIC protocols, the security paradigm of zero information data protection. We fill the void by the fact that to date there has been no new and stable RDIC ID based scheme. Specifically, by using the concept of a new primitive agreement known as the asysmmetric group key, we are suggesting a specific ID-based RDic protocol, a new construction separate from previous ones. More precisely, our Challenge-Access protocol is a two-part central arrangement between the TPA and the cloud server that will use the challenged blocks in the creation of a cloud-shared key to respond to a TPA challenge.

The new protocol contains comprehensive safety proof, including consistency and zero-knowledgement of the stored data. Our protection evidence is conducted in the standardised community model. This is the first accurate ID-based RDIC protocol security proof. The new protection evidence approach may therefore be self-interesting. By designing a sample implementation of the protocol, we demonstrate how realistic the idea is.

The details of the proposed protocol are as follows.

Setup. On input a security parameter sp, the KGC chooses two cyclic multiplicative

https://ijgst.com.2024.v13.i1.pp116-124

groups G_1 and G_2 with prime order q, where g is a generator of G_1. There exists a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. The KGC picks a random $\alpha \in \boldsymbol{Z_q^*}$ as the master secret key and sets $P_{pub} = g^{\alpha}$. Finally, the KGC chooses three hash functions H_1 , $H_2 :\{0,1\}^* \rightarrow G_1$ and $H_3 : G_2 \rightarrow :\{0,1\}^l$. The system parameter is (G_1, G_2, e, g , P_{pub}, H_1, H_2, H_3, l).

Extract. On input the master secret key α and a user's identity ID ∈ {0, 1}*, this algorithm outputs the private key of this user as $s = H_1 (ID)^{\alpha}$.

TagGen. Given a file 'M' named '*fname*', the data owner firstly divides it into 'n' blocks $m_1,\ldots\ldots,m_n$, where $m_i \in Z_q$ and then picks a random $\eta \in \boldsymbol{Z_q^*}$ and computes $r = g^{\eta}$. For each block m_i, the data owner computes

$$\sigma_i = s^{m_i} H_2(fname||i)^{\eta}.$$

The tag for m_i is σ_i. The data owner stores the file 'M' together with (r, $\{\sigma_i\}$, IDS(r||fname)), to the cloud, where IDS(r||fname) is an identity-based signature from the data owner on the value r||fname.

Challenge. The verifier selects a random c-element subset I of the set for testing the integrity of M [1, n] and a random element $v_i \in \boldsymbol{Z_q^*}$ for each i ∈ I do I. Let Q be the set $\{(i,v_i)\}$. To generate a challenge, the verifier picks a random $\rho \in \boldsymbol{Z_q^*}$, computes $Z = e(H_1 (ID), P_{pub})$ and does the following.

1) Compute $c_1 = g^{\rho}$, $c_2 = Z^{\rho}$)
2) Generate a proof that :
   pf = POK{(ρ): $c_1 = g^{\rho} \wedge c_2 = Z^{\rho}$ *}*

The verifier sends the challenge chal = (c_1, c_2, Q, *pf*) to the server.

GenProof. Upon receiving chal = (c_1, c_2, Q, *pf*) the server first computes $Z = e(H_1 (ID), P_{pub})$. Then, it verifies the proof *pf*. If it is invalid, the auditing aborts. Otherwise, the server computes $\mu = \sum_{i \in \mathbf{I}} v_i m_i$, $\sigma = \prod_{i \in \mathbf{I}} \sigma_i^{v_i}$ and $m' = H_3(e(\sigma,c_1).c_2^{-\mu})$ and returns (m', r, IDS(r||fname)) as the response to the verifier as shown in the figure 2.

CheckProof. Upon receiving m' from the server, the verifier checks if IDS(r||fname) is a valid identity-based signature from the data owner on the message r||fname. If not, the roof is invalid. Otherwise, the verifier checks if

$$m' = H_3\left(\prod_{i \in I} e(H_2(fname||i)^{v_i}, r^{\rho})\right).$$

If the equality holds, the verifier accepts the proof; Otherwise, the proof is invalid.

If both the data owner and the cloud server are honest, for each valid tag σ_i and a random challenge, the cloud server can always pass the verification. The completeness of the protocol can be elaborated as follows.

$$\begin{aligned}
m' &= H_3\left(e(\sigma, c_1) \cdot c_2^{-\mu}\right) \\
&= H_3\left(\frac{e(\sigma, c_1)}{e(H(ID), P_{pub})^{\rho \sum_{i \in I} m_i v_i}}\right) \\
&= H_3\left(\frac{e(\prod_{i \in I} \sigma_i^{v_i}, c_1)}{e(s, g)^{\rho \sum_{i \in I} m_i v_i}}\right) \\
&= H_3\left(\frac{\prod_{i \in I} e(\sigma_i^{v_i}, c_1)}{\prod_{i \in I} e(s, c_1)^{m_i v_i}}\right) \\
&= H_3\left(\prod_{i \in I} e(\frac{\sigma_i}{s^{m_i}}, g^{\rho v_i})\right) \\
&= H_3\left(\prod_{i \in I} e(H_2(fname||i)^{\eta}, g^{\rho v_i})\right) \\
&= H_3\left(\prod_{i \in I} e(H_2(fname||i)^{v_i}, g^{\rho \eta})\right) \\
&= H_3\left(\prod_{i \in I} e(H_2(fname||i)^{v_i}, r^{\rho})\right).
\end{aligned}$$

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-01 May 2024
https://ijgst.com.2024.v13.i1.pp116-124

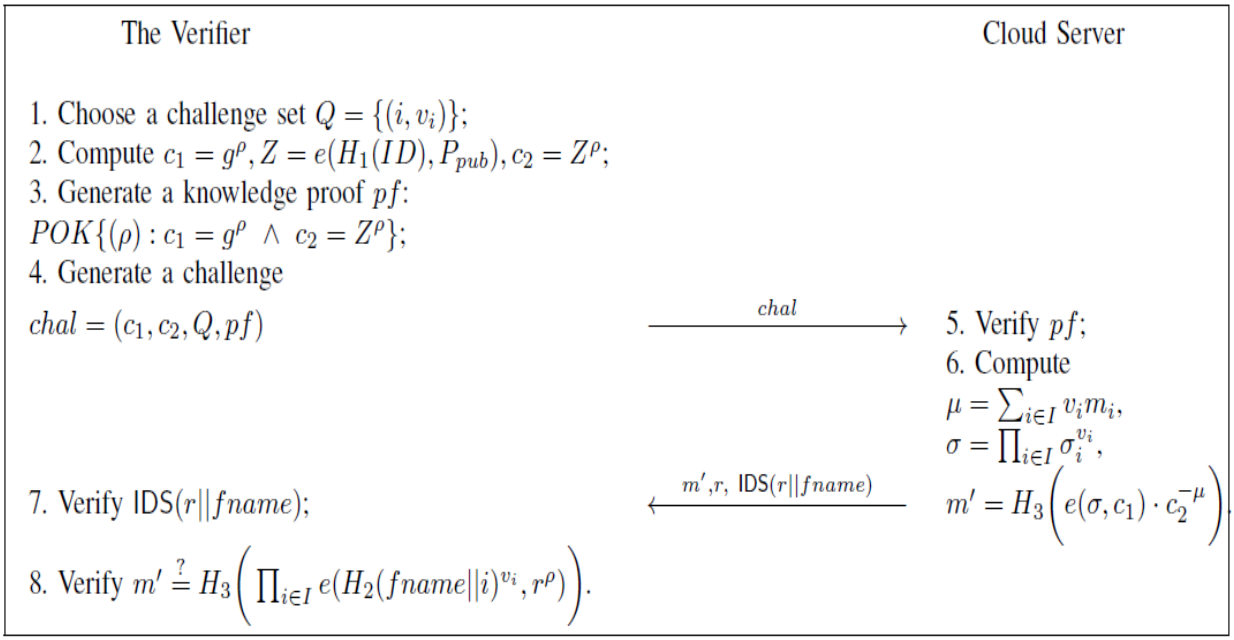

Fig. 2. Identity-based remote data integrity checking protocol

IV RESULTS AND DISCUSSION

In this section, we first numerically analyze the cost of the new protocol, and then report its experimental results.

Numerical Analysis

We provide a numerical analysis of costs regarding computation, communication and storage of the proposed protocol in this part.

Computation cost.

In terms of KGC, data holders, the cloud server and the verifier, we present the measurement costs (TPA). For ease of operation, EG1 EG2 denotes exponentiations in G1 and G2, in MG1, MG2, in G1 and G2, in P, in mixture computations, and in H, map-to-point hash. In numerical analysis, we neglect the usual hash functions, since these functions' costs in contrast with other operations are negligible.

The primary computation of the KGC is generating system parameters and private key for each user. Thus, the main computational cost of KGC is $2E_{G1}$ + 1H. The dominated computation of data owner is generating tags for file blocks as $\sigma_i = s^{m_i} H_2(\text{fname}||i)^\eta$, which is the most expensive operation in the protocol but fortunately it can be done offline. For a file with n blocks, the cost of the data owner is $(2n+1)E_{G1} + nH$. The main cost of the verifier is generating a challenge and checking the validity of a proof. The verifier needs to perform 1 pairing operation, and 6 exponentiations in G_1 to generate a challenge when using the proof of equality of discrete logarithm. When checking a proof, the cost of the verifier is $(c+1)E_{G1} + cP + cH + (c-1)E_{G2}$.

The main computation cost of generating a proof by the cloud server is calculating the aggregation of σ_i, that is $\sigma = \prod_{i \in I} \sigma_i^{v_i}$, and the total cost is $2P+(2c-1)M_{G1} + E_{G2} + M_{G2}$.

Communication cost. In the challenge phase, the verifier sends (c_1, c_2, Q, *pf*) to the server, where $Q = \{(i,v_i)\}$ denotes the challenge set. In practice, we can employ a pseudorandom function θ and a pseudo-random permutation ψ as the public parameters. The verifier only needs to include a key k of θ and a key k2 of ψ in the challenge, instead of the challenge set Q,

https://ijgst.com.2024.v13.i1.pp116-124

which can reduce the communication cost significantly. In this case, the communication cost is of binary length $\log_2 c_1 + \log_2 c_2 + \log_2 k_1 + \log_2 k_2 + \log_2 pf$. In the response phase, the server returns (m', r, IDS(r||fname) as the response to the verifier. An identity-based signature usually contains two points (320 bits) of elliptic curves, thus the communication cost of the response is of binary length $l + \log_2 r + 320$.

Storage cost. Regarding the storage cost of the cloud server, the data owner and the verifier, since identity-based data auditing schemes are publicly verifiable, both the data and the tags are stored on the cloud server. An identity-based digital signature algorithm is used to protect the commitment r from being tampered by external and internal adversaries. In this case, what stored on the cloud are as follows.

Data	Tags	r‖IDS(r,fname)

The data owner needs to store nothing in the context of public verifiability. With the knowledge of the identity information of the data owner, a verifier is able to check the integrity of the data on behalf the data owner. Thus, what the verifier needs to store is the challenge set Q and the value ρ randomly selected by himself. The storage cost of a verifier can be reduced to $\log_2 k_1 + \log_2 k_2 + \log_2 q$ bits. The storage cost of the block tags is upper bounded by $[\log_2 (M)/\log_2 q]160$ bits when employing proper elliptic curves. The cloud server hosts the data, the tags, r and its signature. As a consequence, the storage cost of the cloud server is upper bounded by $\log_2 M + [\log_2 (M)/\log_2 q]160 + \log_2 r + len(IDS(r||fname))$.

Implementation Results

Pbc-0.5.13 with pbc wrapper-0.8.0 on Intel i7-4700MQ Processor @ 2.40GHz was used for the implementation. Memory is often enough, because only a polynomial area is needed for the method. We use a.param parameter, one of the basic settings in pbc library, in our implementation. Parameter a.param gives the most rapid symmetric pairing of default parameters. The time during which the protocol is applied is seen in the following two sections.

In the first part, we setup a file of a constant size of 1 MB, and observe the impact of the number of challenged blocks in terms of the time cost. In our setting, the size of a data block is bounded by the group order p, i.e., 160 bits. Hence, we have 50, 000 blocks in total. This implies that the timing results for Setup, Extract and TagGen steps are constant for this part. See Table I for more details. We can see that both the Setup algorithm and the Extract algorithm are extremely fast. The Setup algorithm picks some random values and compute a modular exponentiation in G_1 , which costs 4.8ms, and the Extract algorithm needs to perform one modular exponentiation in G_1 for generating the private key of a cloud user, which cost 0.1ms. The TagGen algorithm is

expensive and we show that the TagGen timing result consists of two phases, an off-line phase, where the data owner can preprocess $H_2(fname||i)^{\eta}$ without knowing the actual data; and an on-line phase, where the data owner needs to compute smi for each data block. (Note that since all those exponentiations are carried out with a same base, we can use a lookup table to accelerate those single-base multiple-exponent operations.) The time cost of off-line computation of generating tags for 1 MB file is 241.9 seconds while the on-online time cost is 20.3 seconds. This is an acceptable result compared with the previous preprocessing result. We then increase the number of challenged blocks from 50 to 1000 with an increment of 50 for each test to see the time cost of Challenge, GenProof and CheckProof steps. As one shall see from Figure 3, the timing cost of those three parts

https://ijgst.com.2024.v13.i1.pp116-124

increases with the increase of the number of challenges.

Setup	Extract	TagGen: off-line	TagGen: on-line	Challenge	GenProof	CheckProof
4.8 ms	0.1 ms	241.9 second	20.3 second	351 ns per challenge	1.3 ms per challenge	6.6 ms per challenge

TABLE I: SUMMERISE OF THE TIME COST FOR A 1 MB FILE

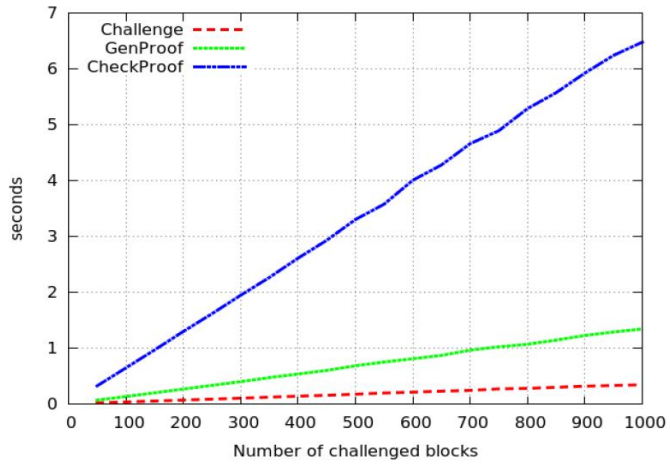

Fig. 3. Increasing number of challenges for fixed size of file

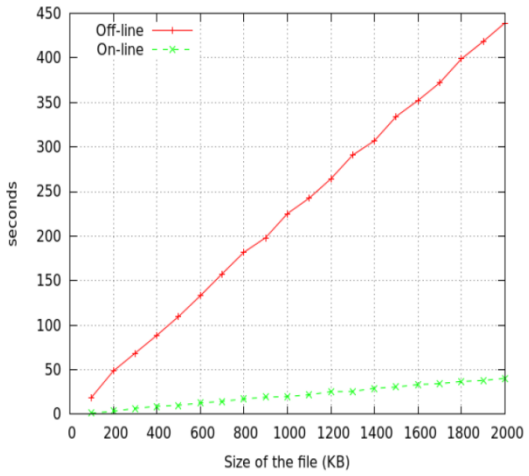

Fig. 4. Tag generation time for increased size of files

In the second part, we test the most expensive algorithm TagGen of the protocol by increasing the file size from 200 KB to 2 MB, that is, from 10, 000 blocks to 100, 000 blocks accordingly, and record the time for TagGen. As expected, both on-line and off-line time to generate tags for a given file increases almost linearly with the increase of the file size. Figure 4 gives more details. The implementation shows that generating tags is more expensive than other parts but fortunately, computing tags for a file is a one time task, as compared to challenging the outsourced data, which will be done repeatedly. Since the cloud users can do the off-line work completely parallelizable in advance, we pay only attention to the on-line cost. We can see that the efficiency of TagGen of our protocol is comparable to that of the existing well-known schemes. To generate tags for a 2 MB file, it costs almost 42 seconds. As such, one shall be able to anticipate the time cost of generating tags for any size of files.

V CONCLUSION

In this paper, we investigated a new primitive called identity-based remote data integrity checking for secure cloud storage. We formalized the security model of two important properties of this primitive, namely, soundness and perfect data privacy. We provided a new construction of of this primitive and showed that it achieves soundness and perfect data privacy.

Both the numerical analysis and the implementation demonstrated that the proposed protocol is efficient and practical.

REFERENCES

[1] Pooja Natu , Prof. Shikha Pachouly.(2014) “A Comparative Analysis of Provable Data Possession Schemes in Cloud”. In (IJCSIT) International Journal of

https://ijgst.com.2024.v13.i1.pp116-124

Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7927-7931.

[2] Zhou F., Peng S., Xu J., Xu Z. (2016) "Identity-Based Batch Provable Data Possession". In: Chen L., Han J. (eds) Provable Security. ProvSec 2016. Lecture Notes in Computer Science, vol 10005. Springer, Cham.

[3] Zhou, F., Peng, S., Xu, J., & Xu, Z. (2017). Identity-Based Batch Provable Data Possession with Detailed Analyses. *International Journal of Foundations of Computer Science*, *28*(06), 743-760.

[4] Prasad, B. S. V., Kishan, C. H., Praveen, S. P., & Teja, C. M. (2018). Identity-Based Data integrity checking in public cloud with bilinear pairings. *International Journal of Engineering & Technology*, *7*(2.7), 209-211.

[5] Shi, Y., Ma, Z., Qin, R., Wang, X., Wei, W., & Fan, H. (2019). Implementation of an Attribute-Based Encryption Scheme Based on SM9. *Applied Sciences*, *9*(15), 3074.

[6] Chandrasekaran, B., & Balakrishnan, R. (2018). An Efficient Tate Pairing Algorithm for a Decentralized Key-Policy Attribute Based Encryption Scheme in Cloud Environments. *Cryptography*, *2*(3), 14.

[7] Peng, S., Zhou, F., Wang, Q., Xu, Z., & Xu, J. (2017). Identity-based public multi-replica provable data possession. *IEEE Access*, *5*, 26990-27001.

[8] Li, C., & Wang, H. (2016, November). Efficient dynamic provable data possession from dynamic binary tree. In *International Conference on Provable Security* (pp. 101-111). Springer, Cham.

[9] Wadhwa, D., & Dabas, P. (2014, September). A coherent dynamic remote integrity check on cloud data utilizing homomorphic cryptosystem. In *2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)* (pp. 91-96). IEEE.

[10] Keerthana, A., & JayaPrakash, S. (2015). Scalable Identity-Based Distributed Provable Data Possession in Multi-Cloud Storage. *International Journal of Computer Science and Engineering Communications*, *3*(2), 889-892.

[11] Y. Yu et al., "Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 767-778, April 2017, doi: 10.1109/TIFS.2016.2615853.