Set of Security Parameters for Cloud Computing Storage System

Nikkita Prakash Jain
PG student,
Dr. D. Y.Patil College of Engineering Ambi,
Pune, India

ABSTRACT
Storage-as-a-Service offered by cloud service suppliers (CSPs) could be a paid facility that permits organizations to source their sensitive information to be hold on remote servers. During this paper, we tend to propose a cloud-based storage theme permits} the info owner to learn from the facilities offered by the CSP and enables indirect mutual trust between them. The planned theme has four vital features: (i) it permits the owner to source sensitive information to a CSP, and perform full block-level dynamic operations on the outsourced information, i.e., block modification, insertion, deletion, and append, (ii) it ensures that licensed users (i.e., those that have the proper to access the owner's file) receive the most recent version of the outsourced information, (iii) it allows indirect mutual trust between the owner and also the CSP, and (iv) it permits the owner to grant or revoke access to the outsourced information. we tend to discuss the protection problems with the planned theme. Besides, we tend to justify its performance through theoretical analysis and a model implementation on Amazon cloud platform to judge storage, communication, and computation overheads.

IndexTerm
Outsourcing information storage, dynamic atmosphere, mutual trust, access management

1. INTRODUCTION
In the current era of digital world, numerous organizations turn out an oversized quantity of sensitive knowledge together with personal info, electronic health records, and money knowledge. The native management of such Bob dingnagian quantity of knowledge is problematic and dear thanks to the wants of high storage capability and qualified personnel. Therefore, Storage-as-a-Service offered by cloud service suppliers (CSPs) emerged as an answer to mitigate the burden of enormous native knowledge storage and cut back the upkeep price by means that of outsourcing knowledge storage.

Since information| the info| the information} owner physically releases sensitive data to a far off CSP, there square measure some issues relating to confidentiality, integrity, and access management of the information. The confidentiality feature may be secure by the owner via encrypting the information before outsourcing to remote servers. For substantiating information integrity over cloud servers, researchers have planned demonstrable information possession technique to validate the ne plus ultra of information keep on remote sites. variety of PDP protocols are bestowed to expeditiously validate the integrity of information, e.g., [1]–[8]. Proof of irretrievability [9]–[12] was introduced as a stronger technique than PDP within the sense that the complete record may be reconstructed from Parts of the information that square measure dependably keep on the servers. Commonly, ancient access management techniques assume the existence of the information owner And also the storage servers within the same trust domain. This assumption, however, not holds once the information is Outsourced to a foreign CSP, that takes the complete charge of the outsourced knowledge management, and resides outside the trust domain of the information owner. A possible answer is often conferred to alter the owner to enforce access management of the information hold on a foreign entrusted CSP. Through this answer, the information is encrypted underneath an explicit key that is shared solely with the approved users. The unauthorized users, as well as the CSP, are unable to access the information since they are doing not have the decoding key. This general answer has been wide incorporated into existing schemes [13]–[16], that aim at providing knowledge storage security on entrusted remote servers. Another category of solutions utilizes attribute-based encoding to realize fine-grained access management [17], [18].

2. LITERATURE SURVEY
1. F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures 2008. Checking data possession in networked information systems such as those related to critical infrastructures (power facilities, airports, data vaults, defense systems, etc.) is a matter of crucial importance. Remote data possession checking protocols permit to check that a remote server can access an uncorrupted file in such a way that the verifier does not need to know beforehand the entire file that is being verified. Unfortunately, current protocols only allow a limited number of successive verifications or are impractical from the computational point of view. In this paper, we present a new remote data possession checking protocol such that: 1) it allows an unlimited number of file integrity verifications; 2) its maximum running time can be chosen at set-up time and traded off against storage at the verifier.

2. G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," 2008 Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this problem using either public key cryptography or requiring the client to outsource its data in encrypted form.

3. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing 2009

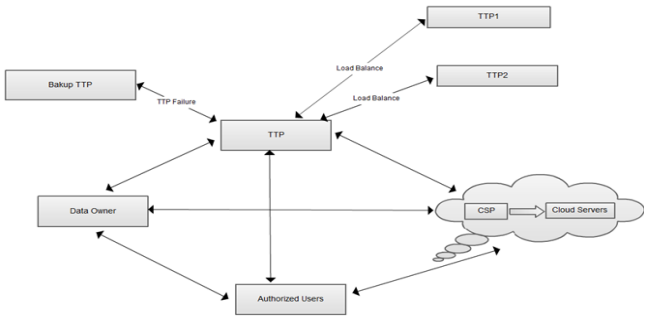
Cloud Computing has been envisioned as the next-generation architecture o f IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new

security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public verifiability or dynamic data operations, this paper achieves both. We first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, we improve the Proof of Irretrievability model [1] by manipulating the classic Merle Hash Tree (MHT) construction for block tag authentication. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

4. C. Erway, A. K¨upc¸ ¨ u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession,"2009

As storage-outsourcing services and resource-sharing networks have become popular, the problem of efficiently proving the integrity of data stored at entrusted servers has received increased attention. In the provable data possession (PDP) model, the client pre-processes the data and then sends it to an entrusted server for storage, while keeping a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). However, the original PDP scheme applies only to static (or append-only) files. We present a definitional framework and efficient constructions for dynamic provable data possession (DPDP), which extends the PDP model to support provable updates to stored data. We use a new version of authenticated dictionaries based on rank information. The price of dynamic updates is a performance change from O(1) to O(log n) (or O(n ǫ log n)), for a file consisting of n blocks, while maintaining the same (or better, respectively) probability of misbehavior detection. Our experiments show that this slowdown is very low in practice (e.g., 415KB proof size and 30ms computational overhead for a 1GB file). We also show how to apply our DPDP scheme to outsourced file systems and version control systems (e.g., CVS).

3. IMPLEMENTATION DETAILS



System components and relations : The cloud computing storage model thought of during this work consists of four main parts as illustrated in Fig. 1: (i) a knowledge owner that may be a corporation generating sensitive data to be hold on within the cloud and created on the market for controlled external use; (ii) a CSP WHO manages cloud servers and provides paid cupboard space on its infrastructure to store the owner's files and create them available for approved users; (iii) approved users – a set of owner's shoppers WHO have the proper to access the remote data; and (iv) a sure third party (TTP), an entity WHO is sure by all different system parts, and has capabilities to detect/specify dishonest parties.

In Fig. 1, the relations between completely different system parts are painted by double-sided arrows, where solid and dotted arrows represent trust and distrust relations, severally. as an example, the info owner, the authorized users, and therefore the CSP trust the TTP. On the opposite hand, the info owner and therefore the approved users have mutual distrust relations with the CSP. Thus, the TTP is used to alter indirect mutual trust between these 3 components. there's an instantaneous trust relation between the data owner and therefore the approved users.

Outsourcing, updating, and accessing. the information owner has a file F consisting of m blocks. For confidentiality, the owner encrypts the information before causing to cloud servers. when information outsourcing, the owner will act with the CSP to perform block-level operations on the file. additionally, the owner enforces access management by granting or revoking access rights to the outsourced data. To access the information, the approved user sends a data-access request to the CSP, and receives the information file in associate encrypted type that may be decrypted employing a secret key generated by the approved user (more details can be explained later).

Security necessities. Confidentiality: outsourced information must be protected against the TTP, the CSP, and users that aren't granted access. Integrity: outsourced information is needed to stay intact on cloud servers. The data owner and approved users should be enabled to acknowledge data corruption over the CSP aspect. Newness: receiving the foremost recent version of the outsourced information file is an essential demand of cloud-based storage systems. There should be a detection mechanism if the CSP ignores any data-update requests issued by the owner. Access control: solely approved users ar allowed to access the outsourced information. Revoked users will browse This article has been accepted for publication in an exceedingly future issue of this journal, however has not been totally altered. Content might modification before final publication. 3 unmodified information, however, they need to not be able to read updated/new blocks. CSP's defense: the CSP should be safeguarded against false accusations that will be claimed by dishonest owner/users, and such a malicious behavior is needed to be disclosed.

4. SYSTEM FEATURES

4.1. Lazy Revocation:

The planned theme during this work permits the info owner to revoke the proper of some users for accessing the outsourced knowledge. In lazy revocation, it's acceptable for revoked users to browse unqualified knowledge blocks. However, updated or new blocks should not be accessed by such revoked users. The notation of lazy revocation was 1st introduced in [20]. the thought is that permitting revoked users to browse unchanged knowledge blocks isn't a big loss in security. This is often comparable to accessing the blocks from paid copies. Updated or new blocks following a evocation square measure encrypted beneath new keys. Lazy revocation trades re-encryption and knowledge access price for a degree of security. However, it

causes fragmentation of coding keys, i.e., knowledge blocks may have a lot of than one key. Lazy revocation has been incorporated into many crypto logic systems [19], [21], [22].

4.2 Key Rotation

Key rotation [13] may be a technique during which a sequence of keys is often generated from associate degree initial key and a master secret key. The sequence of keys has 2 main properties: (i) solely the owner of the master secret key is able to come up with succeeding key within the sequence from the current key, and (ii) any approved user knowing a key within the sequence is in a position to come up with all previous versions of that key. In alternative words, given the i-th key Ki within the sequence, it's computationally unfeasible to compute keys for l > i while not having the master secret key, however it's straightforward to cipher keys for j < i. The projected theme during this work utilizes the key rotation technique [13]. Let N = pq denote the RSA modulus (p&q area unit prime numbers), a public key = (N, e), and a master secret key d. The key d is understood solely to the data owner, and impotence ≡ one mod (p − 1)(q − 1).

4.3 Broadcast cryptography

Broadcast cryptography (bENC) permits a broadcaster to encrypt a message for Associate in Nursing discretional set of a gaggle of users. The users within the set area unit solely allowed to rewrite the message. However, though all users outside the subset conspire they cannot access the encrypted message. The planned theme uses bENC [23] to enforce access control in outsourced knowledge. The bENC [23] consists of 3 algorithms: SETUP, ENCRYPT, and rewrite.

SETUP. This algorithmic program takes as input the quantity of system users n. It defines a linear cluster G of prime order p with a generator g, a cyclic increasing cluster GT , and a linear map ˆe : G × G → GT . The algorithmic program picks a random α ∈ Zp, computes gi = g(αi) ∈ G for i = 1, 2, . . . , n,n + 2, . . . , 2n, and sets v = gγ ∈ G for γ ∈R Zp. The outputs square easure a public key PK = (g, g1, . . . , gn, gn+2, . . . , g2n, v) ∈ G2n+1, and n personal keys 1≤i≤n, wherever di = gγ i∈ G.

ENCRYPT. This algorithmic rule takes as input a set S ⊆ , and a public key PK. It outputs a pair (Hdr, K), wherever Hdr is named the header (broadcast ciphertext), and K could be a message encoding key. Hdr = (C0, C1) ∈ G2, wherever for t ∈R Zp, C0 = gt and C1 = (v · j∈S gn+1−j)t. The key K = ˆe(gn+1, g)t is employed to inscribe a message M (symmetric encryption) to be broadcast to the set S. DECRYPT. This algorithmic rule takes as input a set S ⊆ , a user-ID i ∈ , the non-public key di for user i, the header Hdr = (C0, C1), and therefore the public key PK. If i ∈ S, the algorithmic rule outputs the key K=ˆe(gi,C1)/ˆe(di. Π j∈S gn+1−j+i, C0), which may be wont to decrypt the encrypted version of M.

4.4 Notations

− F = may be a record − h may be a cryptanalytic hash perform

− DEK may be a encoding key

− EDEK may be a radically symmetrical cryptography algorithmic rule underneath DEK, e.g., AES (advanced cryptography standard)

– E −1 DEK may be a radically symmetrical cryptography underneath DEK

− F is associate encrypted version of the file blocks

− FHTTP may be a combined hash price for F, and is computed and keep by the TTP

− THTTP may be a combined hash price for the BST, and is computed and keep by the TTP

− ctr may be a counter unbroken by the info owner to point the version of the foremost recent key

− Rot = ctr, bENC(Kctr) may be a rotator, where bENC(Kctr) may be a broadcast cryptography of Kctr

− ⊕ is associate XOR operator

4.5 Block standing Table

The block standing table (BST) could be a tiny dynamic system used to reconstruct and access file blocks outsourced to the CSP. The BST consists of 3 columns: serial number (SN), block variety (BN), and key version (KV). metal is AN compartmentalization to the file blocks. It indicates the physical position of every block within the record. BN is a counter won't to build a logical numbering/indexing to the file blocks. Thus, the relation between BN and SN is viewed as a mapping between the logical number BN and also the physical position metal. The column KV indicates the version of the key that's wont to encrypt every block within the record.

The BST is enforced as a coupled list to modify the insertion and deletion of table entries. throughout implementation, SN isn't required to be keep within the table; SN is taken into account to be the entry/table index. Thus, each table entry contains simply 2 integers BN and potential unit (8 bytes), i.e., the full table size is 8m bytes, where m is the number of file blocks. When an information file is at first created, the owner initializes both ctr and potential unit of every block to one. If block modification or insertion operations area unit to be performed following a revocation, car is incremented by one and potential unit of that modified/new block is about to be up to ctr.

4.6 Procedural Steps of the planned theme

4.6.1 Setup and File Preparation

The system setup has 2 parts: one is completed on the owner side, and also the different is completed on the TTP aspect. Owner Role. the info owner initializes ctr to one, and generates Associate in Nursing initial secret key Kctr/K1. Kctr are often rotated forward following user revocations, and turned backward to alter licensed users to access blocks that ar encrypted below older versions of Kctr. For a file F = 1≤j≤m, the owner generates a BST with SNj = BNj = j and KVj = ctr. To achieve privacy-preserving, the owner creates Associate in Nursing encrypted file version F {*bj*}= 1≤j≤m, wherever ˜bj = EDEK(BNj ||bj) and DEK = h(Kctr).2 Moreover, the owner creates a rotator Rot = ctr, bENC(Kctr) , wherever bENC allows only licensed users to decipher Kctr and access the outsourced file. The owner sends to the TTP, and deletes the info file from its native storage. Embedding BNj with the block bj throughout the coding process supports in reconstructing the file blocks within the correct order (more details are going to be explained later).

4.6.2 Dynamic Operations on the Outsourced Information

The dynamic operations within the projected theme area unit performed at the block level via missive of invitation within the General Form Block Op, TEntryBlockOp,j,KVj,h(˜bj), RevFlag,b∗,wherever BlockOp corresponds to dam modification (denoted by BM), block insertion (denoted by BI), or block deletion (denoted by BD). TEntryBlockOp indicates AN entry in BSTO similar to the issued dynamic request. The parameter j indicates the block index on that the dynamic operation is to be

performed, KVj is that the worth of the key version at index j of BSTO before running a modification operation, and h(˜bj) is the hash worth of the block at index j before modification/ deletion. RevFlag may be a 1-bit flag (true/false and is initialized to false) to point whether or not a revocation as been performed, and b∗ is that the new block worth.

4.6.3 Knowledge Access and Cheating Detection:

Fig. six shows the verifications performed for the information received from the CSP, and presents however approved users get access to the outsourced file. An authorized user sends a data-access request to both the CSP and therefore the TTP to access the outsourced file. For achieving non-repudiation, the CSP generates two signatures σF and σT for F and BSTC, severally. The user receives from the CSP, and from the TTP. The approved user verifies the signatures, and takings with the information access procedure given that each signatures area unit valid. The approved user verifies the contents of BSTC entries by computing THU = ⊕mj =1 h(BNj||KVj), and comparing it with the authentic worth THTTP received from the TTP. If the user claims that THU = THTTP, a report is issued to the owner and therefore the TTP is invoked to determine the dishonest party. In case of THU = THTTP , the user continues to verify the contents of the file F by computing FHU = ⊕mj =1 h(˜bj) and scrutiny with FHTTP . If there's a dispute that FHU = FHTTP, the owner is knowing and We resort to the TTP to resolve such a conflict.

4.7 Performance Analysis

4.7.1 Settings and Overheads

The data file F utilized in our performance analysis is of size 1GB with 4KB block size. while not loss of generality, we assume that the specified security level is 128-bit. Thus, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS This article has been accepted for publication in a very future issue of this journal, however has not been absolutely emended. Content might modification before final publication. 8 we utilize a cryptological hash h of size 256 bits (e.g., HA-256), AN elliptic curve outlined over Evariste Galois field GF(p) with |p| = 256 bits (used for bENC), and BLS (Boneh-Lynn-Shacham) signature [24] of size 256 bits (used to figure σF and σT). Here we have a tendency to appraise the performance of the planned scheme by analyzing the storage, communication, and computation overheads. we have a tendency to investigate overheads that the planned theme brings to a cloud storage system for static knowledge with solely confidentiality demand. This investigation demonstrates whether or not the options of our scheme come back at an inexpensive value.

4.7.2 Comments:

1. Storage overhead:

It's the extra cupboard space used to store necessary info apart from the outsourced file F. The overhead on the owner facet is attributable to storing BSTO. associate entry of BSTO is of size eight bytes (two integers), and also the total variety of entries equals the number of file blocks m. throughout implementation atomic number 50 is not required to be hold on in BSTO; atomic number 50 is taken into account to be the entry/table index (BSTO is enforced as a joined list). the scale of BSTO for the file F is merely 2MB (0.2% of F). BSTO size is any reduced if the file F is split into larger blocks (e.g., 16KB). Like the owner, the storage overhead on the CSP facet comes from the storage of BSTC. To resolve disputes that may arise concerning information integrity or age property, the TTP stores FHTTP and THTTP , every of size 256 bits. Besides, the TTP stores Rot = ctr, bENC(Kctr) that enables the information owner to enforce access management for the outsourced information. The ctr is four bytes, and bENC has storage quality O(√ n), that is sensible for an organization data owner) with n = one hundred,000 users. A point on the elliptic curve accustomed implement bENC will be diagrammatical by 257 bits (≈ thirty two bytes) itemization compressed representation [25]. Therefore, the storage overhead on the TTP facet is near 10KB, that is freelance of the outsourced file size. Overall, the storage overhead for the file F is a smaller amount than four.01MB (≈ 0.4% of F).

2. Communication overhead:

It's the extra info sent together with the outsourced information blocks. During dynamic operations, the communication overhead on the owner aspect comes from the transmission of a block operation BlockOP (can be diagrammatic by one byte), a table entry TEntry Block OP (8 bytes), and a block index j (4 bytes). If a block is to be changed following a revocation process, KVj (4 bytes) is shipped to the TTP. Moreover, in case of a block modification/deletion, the owner sends a hash (32 bytes) of the block to be modified/deleted to the TTP for change FHTTP . Recall that the owner also sends Rot (4 + thirty two √ n bytes) to the TTP if block modifications/ insertions area unit to be performed following user revocations. Therefore, within the worst case state of affairs (i.e., block modifications following revocations), the owner's overhead is a smaller amount than 10KB. The Rot represents the foremost factor in the communication overhead, and so the overhead is simply forty five bytes if block codification/deletion operations area unit to be preformed while not revocations (only thirteen bytes for insertion operations). In sensible applications, the frequency of dynamic requests to the outsourced information is on top of that of user revocations. Hence, the communication overhead thanks to dynamic changes on the information is regarding a hundred and twenty fifth of the block size (the block is 4KB in our analysis**).**

3.Computation Overhead.

A cloud storage system for static information with solely confidentiality demand has computation value for encrypting the info before outsourcing and decrypting the info when being received from the cloud servers. For the projected theme, the computation overhead on the owner facet attributable to dynamic operations modification/insertion) comes from computing DEK = h(Kctr) and encrypting the updated/ inserted block, i.e., the overhead is one hash and one cryptography operations. If a block modification/ insertion operation is to be performed following a revocation of 1 or a lot of users, the owner performs FR to roll Kctr forward, and bENC to get the Rot. Hence, the computation overhead on the owner facet for the dynamic operations is h + EDEK + atomic number 87 + Enc (worst case scenario). change BSTO and BSTC is completed without usage of scientific discipline operations (add, remove, or modify a table entry). To replicate the foremost recent version of the outsourced data, the TTP updates the values FHTTP and THTTP . If no revocation has been performed before causation a modify request, solely FHTTP is updated on the TTP facet. Therefore, the utmost computation overhead on the TTP facet for change each FHTTP and THTTP is four h.

4.8 Implementation and Experimental Evaluation

4.8.1 Implementation:

We have enforced the planned theme on high of Amazon Elastic work out Cloud (Amazon EC2) and Amazon straightforward Storage Service (Amazon S3) [26] cloud platforms. Our implementation of the planned theme consists of 4 modules: OModule (owner module), CModule (CSP module), UModule

(user module), and TModule (TTP module). OModule, that runs on the owner side, may be a library to be employed by the owner to perform the owner role within the setup and file preparation phase. Moreover, this library is employed by the owner during the dynamic operations on the outsourced knowledge. CModule may be a library that runs on Amazon EC2 and is used by the CSP to store, update, and retrieve knowledge from Amazon S3. UModule may be a library to be run at the licensed users' aspect, and embody functionalities that enable users to act with the TTP and therefore the CSP to retrieve and access the outsourced knowledge. TModule is a library employed by the TTP to perform the TTP role in the setup and file preparation part. Moreover, the TTP uses this library throughout the dynamic operations and to determine the cheating party within the system.

4.8.2 Experimental Analysis:

Here we tend to describe the experimental analysis of the computation overhead the projected theme brings to a cloud storage system that has been addressing static data with solely confidentiality demand. Owner computation overhead. To by experimentation evil- IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS This article has been accepted for publication in an exceedingly future issue of this journal, however has not been totally altered. Content might modification before final publication 10 ate the computation overhead on the owner aspect as a result of the dynamic operations, we've got reformed one hundred completely different block operations (modify, insert, append, and delete) with variety of approved users starting from twenty,000 to 100,000. we've got run our experiment 3 times, each time with a special revocation proportion. within the initial time, five-hitter of one hundred dynamic operations square measure dead following revocations. we tend to redoubled the revocation proportion to 100 percent for the second time and 2 hundredth for the third time. Fig. eight shows the owner's average computation overhead per operation. For an oversized organization (data owner) with 100,000 users, acting dynamic operations and implementing access management with five-hitter revocations add regarding sixty three milliseconds of overhead. With 100 percent and 2 hundredth revocation percentages, that square measure high percentages than a median value in sensible applications, the owner overhead is 0.12 and 0.25 seconds, severally.

5. CONCLUSIONS

In this paper, we've got projected a cloud-based storage scheme that supports outsourcing of dynamic knowledge, where the owner is capable of not solely archiving and accessing the data keep by the CSP, however additionally change and scaling this knowledge on the remote servers. The projected scheme allows the approved users to confirm that they are receiving the foremost recent version of the outsourced data. Moreover, just in case of dispute relating to knowledge integrity/ newness, a TTP is in a position to see the dishonest party. the info owner enforces access management for the outsourced knowledge by combining 3 crypto logical techniques: broadcast coding, lazy revocation, and key rotation. We have got studied the safety options of the projected theme.

We have investigated the overheads superimposed by our scheme once incorporated into a cloud storage model for static information with solely confidentiality demand. The storage overhead is ≈ zero.4% of the outsourced information size, the communication overhead owing to block-level dynamic changes on the info is ≈ one hundred and twenty fifth of the block size, and the communication overhead owing to retrieving the info is ≈ 0.2% of the outsourced information size. For an outsized organization with one zero five users, playing dynamic operations and enforcing access management add regarding sixty three milliseconds of overhead. Therefore, vital options of outsourcing data storage may be supported while not excessive overheads in storage, communication, and computation.

6. REFERENCES

[1] 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 Communications Security, ser. CCS '07, 2007, pp. 598–609.

[2] F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. on Knowl. And Data Eng., vol. 20, no. 8, 2008.

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

[4] C. Erway, A. K¨upc¸ ¨ u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proceedings of the 16th ACM Conference on Computer and Communications Security, 2009, pp. 213–222.

[5] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proceedings of the 14th European Conference on Research in Computer Security, 2009, pp. 355–370.

[6] A. F. Barsoum and M. A. Hasan, "Provable possession and replication of data over cloud servers," Centre For Applied Cryptographic Research, Report 2010/32, 2010,

[7] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: multiple-replica provable data possession," in 28th IEEE ICDCS, 2008, pp. 411–420.

[8] A. F. Barsoum and M. A. Hasan, "On verifying dynamic multiple data copies over cloud servers," Cryptology ePrint Archive, Report 2011/447, 2011, 2011, ttp://eprint.iacr.org/.

[9] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: a high-availability and integrity layer for cloud storage," in CCS '09: Proceedings of the 16th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2009, pp. 187–198.

[10] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography, 2009.

[11] A. Juels and B. S. Kaliski, "PORs: Proofs of Retrievability for large files," in CCS'07: Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 584–597.

[12] H. Shacham and B. Waters, "Compact proofs of retrievability," in ASIACRYPT '08, 2008, pp. 90–107.

[13]] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in Proceedings of the FAST 03: File and Storage Technologies, 2003.

[14] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in Proceedings

of the Network and Distributed System Security Symposium, NDSS, 2003.

[15] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in NDSS, 2005.

[16] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data," in Proceedings of the 33rd International Conference on Very Large Data Bases. ACM, 2007, pp. 123–134.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in CCS '06, 2006, pp. 89–98.

[18] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in INFOCOM'10, 2010, pp. 534–542.

[19] R. A. Popa, J. R. Lorch, D. Molnar, H. J. Wang, and L. Zhuang, "Enabling security in cloud storage SLAs with cloudproof," in Proceedings of the 2011 USENIX conference, 2011.

[20] K. E. Fu, "Group sharing and random access in cryptographic storage file systems," Master's thesis, MIT, Tech. Rep., 1999.

[21] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and efficient access to outsourced data," in Proceedings of the 2009 ACM workshop on Cloud computing security, 2009, pp. 55–66.

[22] M. Backes, C. Cachin, and A. Oprea, "Secure key-updating for lazy revocation," in 11th European Symposium on Research in Computer Security, 2006, pp. 327–346.

[23] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in Advances in Cryptology - CRYPTO, 2005, pp. 258–275.

[24] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, London, UK, 2001, pp. 514–532.

[25] P. S. L. M. Barreto and M. Naehrig, "IEEE P1363.3 submission: Pairing-friendly elliptic curves of prime order with embedding degree 12," New Jersey: IEEE Standards Association, 2006.

[26] Amazon Web Service, http://aws.amazon.com/.

[27] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in Proceedings of SAC 2005, volume 3897 of LNCS. Springer-Verlag, 2005, pp. 319–331.

[28] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," Cryptology ePrint Archive, Report 2006/150, 2006.

[29] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, ser. CRYPTO '01. Springer-Verlag, 2001, pp. 41–62.

[30] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in EUROCRYPT, 1998, pp. 127–144.

[31] M. J. Atallah, K. B. Frikken, and M. Blanton, "Dynamic and efficient key management for access hierarchies," in Proceedings of the 12th ACM Conference on Computer and Communications Security, ser. CCS '05. ACM, 2005, pp. 190–202.

[32] J. Feng, Y. Chen, W.-S. Ku, and P. Liu, "Analysis of integrity vulnerabilities and a non-repudiation protocol for cloud data storage platforms," in Proceedings of the 2010 39th International Conference on Parallel Processing, 2010, pp. 251–258.

[33] J. Feng, Y. Chen, and D. H. Summerville, "A fair multi-party non-repudiation scheme for storage clouds," in 2011 International Conference on Collaboration Technologies and Systems, 2011, pp. 457– 465.