ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2 Issue 1 Jan-March 2014
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Concept of Dynamic Data Storage and Indirect Mutual Trust in Cloud Computing Environment

[I]Christo Paul.E, [II]Saravanakumar.S, [III]Jones Merlin.E
[I]2nd year-M.E CSE, Srinivasan Engineering College, Perambalur, Tamil Nadu, India
[II]Assistant Professor, Dept. of CSE, Srinivasan Engineering College, Perambalur, Tamil Nadu, India
[III]1st year-M.E CSE, Srinivasan Engineering College, Perambalur, Tamil Nadu, India

Abstract
A data owner pays for a desired level of security and must get some compensation in case of any misbehaviour committed by the CSP. On the other hand, the CSP needs a protection from any false accusation that may be claimed by the owner to get illegal compensations. In this paper, we propose a cloud-based storage scheme that allows the data owner to benefit from the facilities offered by the CSP and enables indirect mutual trust between them. The proposed scheme has four important features: First one, it allows the owner to outsource sensitive data to a CSP, and perform full block-level dynamic operations on the outsourced data, i.e., block modification, insertion, deletion, and append. Second one, it ensures that authorized users (i.e., those who have the right to access the owner's file) receive the latest version of the outsourced data. Next one, it enables indirect mutual trust between the owner and the CSP. Last one, it allows the owner to grant or revoke access to the outsourced data. We discuss the security issues of the proposed scheme. Besides, we justify its Performance through theoretical analysis and experimental evaluation of storage, communication, and computation overheads.

Keywords
Outsourcing Data Storage, Dynamic Environment, Mutual Trust, Access Control

I. Introduction

Cloud computing has received considerable attention from both academia and industry due to a number of important advantages including: cost effectiveness, low management overhead, immediate access to a wide range of applications, flexibility to scale up and down Information Technology (IT) capacity, and mobility where customers can access information wherever they are, rather than having to remain at their desks. Cloud computing is a distributed computational model over a large pool of shared-virtualized computing resources (e.g., storage, processing power, memory, applications, service and network bandwidth).Cloud service Provider(CSPs) offer different classes of services : Storage-as-a-Service (SaaS), Application-as-a-Service, and Platform-as-a-Service that allow organizations to concentrate on their core business and leave the IT operations to experts. In the current era of digital world, different organizations produce a large amount of sensitive data including personal information, electronic health records, and financial data.

SaaS offered by CSPs is an emerging solution to mitigate the burden of large local data storage and reduce the maintenance cost via the concept of outsourcing data storage. Through outsourcing data storage scenario, data owners delegate the storage and management of their data to a CSP in exchange for pre-specified fees metered in GB/month. Such outsourcing of data storage enables owners to store more data on remote servers than on private computer systems.

Moreover, the CSP often provides better disaster recovery by replicating the data on multiple servers across multiple data centre achieving a higher level of availability. Thus, many authorized users are allowed to access the remotely stored data from different geographic locations making it more convenient for them. Since the owner physically releases sensitive data to a remote CSP, there are some concerns regarding confidentiality, integrity, and access control of the data. In some practical applications, data confidentiality is not only a privacy concern, but also a juristic issue.

For example, in e-Health applications inside the USA the usage and exposure of protected health information should meet the policies admitted by Health Insurance Portability and Accountability Act (HIPAA), and thus keeping the data private on the remote storage servers is not just an option, but a demand. The confidentiality feature can be guaranteed by the owner via encrypting the data before outsourcing to remote servers.

For verifying data integrity over cloud servers, researchers have proposed provable data possession (PDP) technique to validate the intactness of data stored on remote sites. A number of PDP protocols have been presented to efficiently validate the integrity of static data. Another class of PDP schemes was concerned with the dynamic behaviour of data over remote servers. This class allows the owner to outsource a data file and perform updating or scaling operations on the outsourced data.

Later, a verifier validates that the remote servers keep the data intact and compatible with the dynamic requests issued by the owner. A complementary line of research on PDP has focused on multiple data copies stored over different servers.

Proof of irretrievability (POR) was introduced as a stronger technique than PDP in the sense that the entire data file can be reconstructed from portions of the data that are reliably stored on the servers. In this work, we propose a scheme that addresses some important issues related to outsourcing the storage of data, namely data dynamic, newness, mutual trust, and access control.

One of the core design principles of data outsourcing is to provide dynamic scalability of data for various applications. This means that the remotely stored data can be not only accessed by authorized users, but also updated and scaled by the owner. After updating, the authorized users should receive the latest version of the data (newness property), i.e., a technique is required to detect whether the received data is stale.

This issue is crucial for applications in which critical decisions are taken based on the received data. For example, in e-Health applications a physician may write a prescription based on a patient's medical history received from remote servers. If such

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

medical data is not up-to-date, the given prescription may conflict with the patient's current circumstances causing severe health problems.

Mutual trust between the data owner and the CSP is another imperative issue, which is addressed in the proposed scheme. A mechanism is introduced to determine the dishonest party, i.e., misbehaviour from any side is detected and the responsible party is identified. Last but not least, the access control is considered, which allows the data owner to grant or revoke access rights to the outsourced data.

Main Contributions

Our contributions can be summarized in two main points.

(1). The design and implementation of a cloud-based storage scheme that has the following features:

(a). It allows a data owner to outsource the data to a remote CSP, and perform full dynamic operations at the block-level, i.e., it supports operations such as block modification, insertion, deletion, and append.

(b). It ensures the newness property, i.e., the authorized users receive the most recent version of the data.

(c). It establishes indirect mutual trust between the data owner and the CSP since each party resides in a different trust domain.

(d). It enforces the access control for the outsourced data.

(2). We discuss the security features of the proposed scheme. Besides, we justify its performance through theoretical analysis and experimental evaluation of storage, communication, and computation overheads.

II. Related Work

Existing research close to our work can be found in the areas of integrity verification of outsourced data, Cryptographic file systems in distributed networks, and access control of outsourced data. Different variations of PDP protocols have been presented for static or warehoused data.

Some other PDP schemes consider the case of dynamic data that are usually more prevailing in practical applications. While the schemes are for a single copy of a data file, PDP schemes have been presented for multiple copies of static data].

Reference addresses a PDP construction for multiple copies of dynamic data. Proof of retrievability (POR) is a complementary approach to PDP, and is stronger than PDP in the sense that the entire data file can be reconstructed from portions of the data that are reliably stored on the servers.

This is due to encoding of the data file, for example using erasure codes, before outsourcing to remote servers. Based on proxy re-encryption have introduced a secure distributed storage protocol.

In their protocol, a data owner encrypts the blocks with symmetric data keys, which are encrypted using a master public key. The data owner keeps a master private key to decrypt the symmetric data keys.

Using the master private key and the authorized user's public key, the owner generates proxy re-encryption keys. A semi-trusted server then uses the proxy re encryption keys to translate a cipher text into a form that can be decrypted by a specific granted user, and thus enforces access control for the data.

Discussion

Some aspects related to outsourcing data storage are beyond the setting of both PDP and POR, e.g., enforcing access control, and ensuring the newness of data delivered to authorized users. Even in the case of dynamic PDP, a verifier can validate the correctness of data, but the server is still able to cheat and return stale data to authorized users after the auditing process is done. The schemes have focused on access control and secure sharing of data on untrusted servers.

The issues of full block-level dynamic operations (modify, insert, delete, and append), and achieving mutual trust between the data owners and the remote servers are outside their scope. Although have presented an efficient access control technique and handled full data dynamic over remote servers, data integrity, newness property, and mutual trust are not addressed.

Authorized users in CloudProof are not performing immediate checking for freshness of received data; the attestations are sent at the end of each epoch to the owner for completing the auditing task.

Instantaneous validation of data freshness is crucial before taking any decisions based on the received data from the cloud. CloudProof guarantees write-serializability, which is outside the scope of our current work as we are focusing on owner-write-users-read applications.

III. Our System and Assumptions

System components and relations. The cloud computing storage model considered in this work consists of four main components as illustrated in Fig. 1: (i) a data owner that can be an organization generating sensitive data to be stored in the cloud and made available for controlled external use; (ii) a CSP who manages cloud servers and provides paid storage space on its infrastructure to store the owner's files and make them available for authorized users; (iii) authorized users – a set of owner's clients who have the right to access the remote data; and (iv) a trusted third party (TTP), an entity who is trusted by all other system components, and has expertise and capabilities to detect and specify dishonest parties.

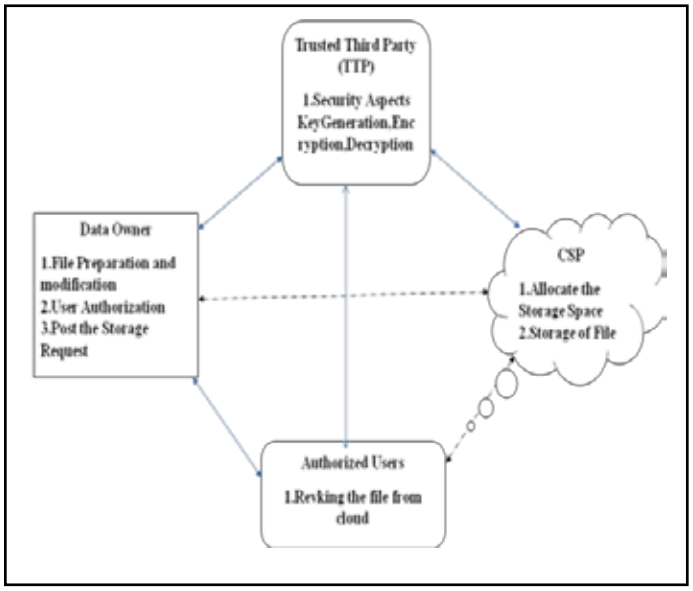

Fig. 1: Overall Architecture

In fig. 1, the relations between different system components are represented by double-sided arrows, where solid and dashed arrows represent trust and distrust relations, respectively. For example, the data owner, the authorized users, and the CSP trust the TTP. On the other hand, the data owner and the authorized users

have mutual distrust relations with the CSP. Thus, the TTP is used to enable indirect mutual trust between these three components. There is a direct trust relation between the data owner and the authorized users.

IV. System Preliminaries

A. Lazy Revocation

The proposed scheme in this work allows the data owner to revoke the right of some users for accessing the outsourced data. In lazy revocation, it is acceptable for revoked users to read (decrypt) unmodified data blocks. However, updated or new blocks must not be accessed by such revoked users.

The idea is that allowing revoked users to read unchanged data blocks is not a significant loss in security.

This is equivalent to accessing the blocks from cashed copies. Updated or new blocks following a revocation are encrypted under new keys. Lazy revocation trades re-encryption and data access cost for a degree of security. However, it causes fragmentation of encryption keys, i.e., data blocks could have more than one key.

B. Key Rotation

Key rotation is a technique in which a sequence of keys can be generated from an initial key and a master secret key. The sequence of keys has two main properties:

- Only the owner of the master secret key is able to generate the next key in the sequence from the current key, and
- Any authorized user knowing a key in the sequence is able to generate all previous versions of that key. In other words, given the i-th key Ki in the sequence, it is computationally infeasible to compute keys Kl for l > i without having the master secret key, but it is easy to compute keys K_j for j < i.

The first property enables the data owner to revoke access to the data by producing new keys in the sequence, which are used to encrypt updated/new blocks following a revocation (lazy revocation).

It is intended to prevent a user revoked during the i-th time from getting access to data blocks encrypted during the l-th time for l > i. The second property allows authorized users to maintain access to blocks that are encrypted under older versions of the current key.

It enables the data owner to transfer only a single key Ki to authorized users for accessing all data blocks that are encrypted under keys Kig (rather than transferring a potentially large set of keys fK1;K2; : : : ;Kig). Thus, the second property reduces the communication overhead on the owner side. The proposed scheme in this work utilizes the key rotation technique]. Let N = pq denote the RSA modulus (p&q are prime numbers), a public key = (N; e), and a master secret key d. The key d is known only to the data owner, and ed = 1 mod (p- 1) (q - 1).

Whenever a user's access is revoked, the data owner generates a new key in the sequence (rotating forward). Let ctr indicate the index/version number of the current key in the keys sequence. The owner generates the next key by exponentiation Kctr with the master secret key d: Kctr+1 = Kdctr mod N.

Authorized users can recursively generate older versions of the current key by exponentiations with the public key component e: Kctr-1 = Kectr mod N (rotating backward). The RSA encryption is used as a pseudorandom number generator; it is unlikely that repeated encryption results in cycling, for otherwise, it can be used to factor the RSA modulus N.

C. Broadcast Encryption

Broadcast encryption (bENC) allows a broadcaster to encrypt a message for an arbitrary subset of a group of users. The users in the subset are only allowed to decrypt the message. However, even if all users outside the subset collude they cannot access the encrypted message. Such systems have the collusion resistance property, and are used in many practical applications including TV subscription services and DVD content protection. The proposed scheme in this work uses bENC to enforce access control in outsourced data. The bENC is composed of three algorithms: SETUP, ENCRYPT, and DECRYPT.

1. Setup

This algorithm takes as input the number of system users n. It defines a bilinear group G of prime order p with a generator g, a cyclic multiplicative group G_T, and a bilinear map $\hat{e} : G \times G \rightarrow G_T$, which has the properties of bilinearity, computability, and non-degeneracy . The algorithm picks a random $\alpha \in Z_p$, computes $g_i = g^{(\alpha i)} \in G$ for i = 1,2,...,n,n+2,...,2n, and sets $v = g^{\beta} \in G$ for $\beta \in_R Z_p$. The outputs are a public key $PK = (g, g_1,...,g_n, g_{n+2},...,g_{2n},v) \in G^{2n+1}$, and n private keys $\{di\}_{1 \le i \le n}$, where $d_i = g_i^{\beta} \in G$.

2. Encrypt

This algorithm takes as input a subset $S \subseteq \{1,2,...,n\}$, and a public key PK. It outputs a pair (Hdr, K), where Hdr is called the header (broadcast ciphertext), and K is a message encryption key. $Hdr = (C_0, C_1) \in G^2$, where for $t \in_R Z_p$, $C_0 = g^t$ and $C_1 = (v \cdot_{\pi j \in} S\, g_{n+1-j})^t$. The key $K = \hat{e}(g_{n+1}, g)^t$ is used to encrypt a message M (symmetric encryption) to be broadcast to the subset S.

3. Decrypt

This algorithm takes as input a subset $S \subseteq \{1,2,...,n\}$, a user-ID $i \in \{1,2,...,n\}$, the private key d_i for user i, the header $Hdr = (C_0, C_1)$, and the public key PK. If $i \in S$, the algorithm outputs the key $K = \hat{e}(g_i,C_1)/\hat{e}(d_i \cdot_{\pi j \in S, j \neq I} g_{n+1-j+i},C_0)$, which can be used to decrypt the encrypted version of M.

In the above construction of the bENC, a private key contains only one element of G, and the broadcast ciphertext (Hdr) consists of two elements of G. On the other hand, the public key PK is comprised of 2n + 1 elements of G.

A second construction, which is a generalization of the first one was presented in to trade the PK size for the Hdr size. The main idea is to run multiple parallel instances of the first construction, where each instance can broadcast to at most B users. Setting $B = [\sqrt{n}]$ results in a system with $O(\sqrt{n})$ elements of G for each of PK and Hdr. The private key is still just one element.

D. Block Status Table

The Block Status Table (BST) is a small dynamic data structure used to reconstruct and access file blocks outsourced to the CSP. The BST consists of three columns: Serial Number (SN), Block Number (BN), and Key Version (KV). SN is an indexing to the file blocks. It indicates the physical position of each block in the data file. BN is a counter used to make a logical numbering/ indexing to the file blocks. Thus, the relation between BN and SN can be viewed as a mapping between the logical number BN and the physical position SN. KV indicates the version of the key that is used to encrypt each block in the data file.

The BST is implemented as a linked list to simplify the insertion

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

and deletion of table entries. During implementation, SN is not needed to be stored in the table; SN is considered to be the entry/ table index. Thus, each table entry contains just two integers BN and KV (8 bytes), i.e., the total table size is 8m bytes, where m is the number of file blocks. When a data file is initially created, the owner initializes both ctr and KV of each block to 1. If block modification or insertion operations are to be performed following a revocation, ctr is incremented by 1 and KV of that modified/ new block is set to be equal to ctr.

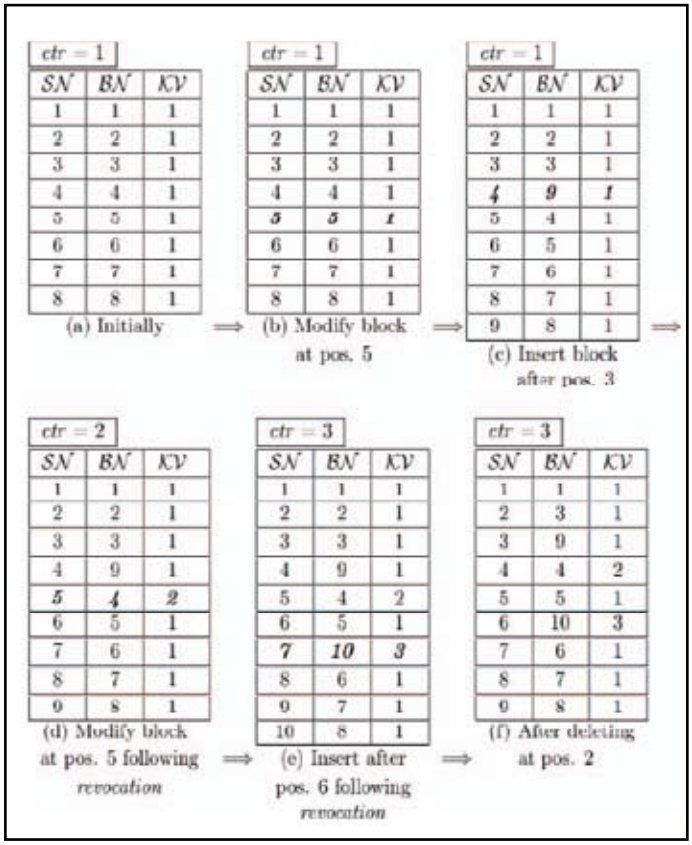

Fig. 2: Changes in BST Due to Different Dynamic Operation on a File F=$\{b_j\}_{1\le j\le 8}$

When a data file is initially created, the owner initializes both ctr and KV of each block to 1. If block modification or insertion operations are to be performed following a revocation, ctr is incremented by 1 and KV of that modified/new block is set to be equal to ctr .Figure shows some examples demonstrating the changes in the BST due to dynamic operations on a data file F = {bj} 1≤j≤8. When the file blocks are initially created (Fig.(a)), ctr is initialized to 1, SNj =BNj = j, and KVj = 1: 1 ≤ j ≤ 8. Fig. (b) shows no change for updating the block at position 5 since no revocation is performed.

To insert a new block after position 3 in the file F, Fig. (c) shows that a new entry h4,9,1i is inserted in the BST after SN3, where 4 is the physical position of the newly inserted block, 9 is the new logical block number computed by incrementing the maximum of all previous logical block numbers, and 1 is the version of the key used for encryption.

A first revocation in the system increments ctr by 1 (ctr = 2). Modifying the block at position 5 following a revocation (Fig. (d)) results in setting KV5 = ctr. Thus, the table entry at position 5 becomes h5, 4, 2i. (Fig. (e)) shows that a new block is to be inserted after position 6 following a second revocation, which increments ctr to be 3. In Fig. (e), a new table entry h7, 10, 3i is inserted after SN6, where KV7 is set to be equal to ctr (the most recent key version). Deleting a block at position 2 from the data file requires deleting the table entry at SN2 and shifting all subsequent entries one position up (Fig. (f)) Note that during all dynamic operations, SN indicates the actual physical positions of the data blocks in F.

V. Experimental Evaluation

In this section we experimentally evaluate the computation overhead the proposed scheme brings to a cloud storage system that has been dealing with static data with only confidentiality requirement.

The experiments are conducted using .NET on a system with an Intel(R) Xeon (R) 2-GHz processor and 3GB RAM running Windows XP. Algorithms (hashing, broadcast encryption, digital signatures, etc.) are implemented using MIRACL library version 5.5.4.

For a 128-bit security level, bENC uses an elliptic curve with a 256-bit group order. In the experiments, we utilize SHA-256, 256-bit BLS signature, and Barreto-Naehrig (BN) curve defined over prime field GF(p) with p = 256 bits and embedding degree = 12 (the BN curve with these parameters is provided by the MIRACL library).

To evaluate the computation overhead on the owner side due to dynamic operations, we perform 100 different block operations from which 50% are executed following revocations (this percent is higher than an average value in practical applications).

Scalability (i.e., how the system performs when more users are added) is an important feature of cloud storage systems. The access control of the proposed scheme depends on the square root of the total number of system users.

In the worst case, the TTP executes only 4 hashes per dynamic request to reflect the change on the outsourced data. Thus, the maximum computation overhead on the TTP side is about 0.08 milliseconds, i.e., the proposed scheme brings light overhead on the TTP during the normal system operations.

The computation overhead on the user side due to data access comes from five aspects divided into two groups. The first group involves signatures verification and hash operations to verify the received data (file and table). The second group involves broadcast decryption, backward key rotations, and hash operations to compute the DEK.

The first group costs about 10.77 seconds, which can be easily hidden in the receiving time of the data (1GB file and 2MB table). To investigate the computation time of the second group, we access the file after running 100 different block operations (50% of them are done following revocations). Moreover, we implement the backward key rotations in the optimized way. The second group costs about 1.03 seconds, which can be considered as the user's computation overhead due to data access.

As a response to the data access request, the CSP computes two signatures: F and T . Thus, the computation overhead on the CSP side due to data access is about 10.75 seconds and can be easily hidden in the transmission time of the data (1GB file and 2MB table).

To identify the dishonest party in the system in case of disputes, the TTP verifies two signatures (F and T), computes combined hashes for the data (file and table), and compare the computes hashes with the authentic values (THTTP and FHTTP).

Thus, the computation overhead on the TTP side is about 10.77 seconds. Through our experiments, we use only one desktop

computer to simulate the TTP and accomplish its work. In practice, the TTP may choose to split the work among a few devices or use a single device with a multi-core processor which is becoming prevalent these days, and thus the computation time on the TTP side is significantly reduced in many applications.

VI. Conclusion

Outsourcing data to remote servers has become a growing trend for many organizations to alleviate the burden of local data storage and maintenance. In this work we have studied different aspects of outsourcing data storage: block-level data dynamic, newness, mutual trust, and access control.

We have proposed a cloud-based storage scheme which supports outsourcing of dynamic data, where the owner is capable of not only archiving and accessing the data stored by the CSP, but also updating and scaling this data on the remote servers. The proposed scheme enables the authorized users to ensure that they are receiving the most recent version of the outsourced data. Moreover, in case of dispute regarding data integrity/newness, a TTP is able to determine the dishonest party. The data owner enforces access control for the outsourced data by combining three cryptographic techniques: broadcast encryption, lazy revocation, and key rotation. We have studied the security features of the proposed scheme.

VII. Acknowledgement

First and foremost, The authors would like to thank the God Almighty, who guides us always in the path of knowledge and wisdom.We thank the editors and anonymous reviewers for their valuable comments to significantly improve the quality of this paper.We are very much grateful to all the staff members and my friends who helped a lot to complete this research work and journal publication.

References

[1] *C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in INFOCOM, 2010, pp. 525–533.*

[2] *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, ser. CCSW '09. ACM, 2009, pp. 55–66.*

[3] *C. Erway, A. Kupu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in CCS'09: Proceedings of the 16th ACM Conference on Computer and Communications Security, New York, NY, USA, 2009, pp. 213–222.*

[4] *G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in SecureComm '08: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, New York, NY, USA, 2008, pp.1–10.*

[5] *G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security, New York, NY, USA, 2007,pp. 598–609.*

[6] *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.*

[7] *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 Conference on File and Storage Technologies. USENIX, 2003.*

[8] *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. 514–53.*

Christo Paul.E received the B.Tech. degree in Information Technology from Roever Engineering College affiliated to Anna University, Tamil Nadu, India, 2012 and also he got 37th university Rank in that academic year of 2012. He joined Srinivasan Engineering College affiliated to Anna University in 2012 for the M.E. degree in Computer Science and Engineering. His research interests include cloud computing, mobile computing and applied cryptography.

Saravanakumar.S received the B.E. and M.E. degrees in Computer Science and Engineering from Annamalai University, Tamil Nadu, India in 2009 and 2011, respectively. He is an Assistant Professor and supervisor of M.E. students at Srinivasan Engineering College affiliated to Anna University. His research interests include networking, cryptography, network security and cloud computing.

Jones Merlin.E received the B.Tech. degree in Information Technology from Roever Engineering College affiliated to Anna University, Tamil Nadu, India, 2013 and also she got 37th university Rank in that academic year of 2013 . She joined Srinivasan Engineering College affiliated to Anna University in 2013 for the M.E. degree in Computer Science and Engineering. Her research interests include cloud computing, mobile computing, data mining and applied cryptography.