IJRSET Volume 2, Issue 4 Pages: 9-17

ISSN 2394-739X

International Journal for Research in Science Engineering and Technology

A Novel Approach in Hybrid Cloud for Secured Authorized Deduplication

[1] **S. SATHYA ANAND**, [2] **Mrs. D. CHITRA., MCA, M.Phil.**
[1]Research Scholar, [2]Assistant Professor of Computer Science,
[1]PG and Research Dept of Comp Sci, [2]PG and Research Dept of Comp Sci,
[1]Govt Arts College (Autonomous), [2]Govt Arts College (Autonomous),
[1]Salem-636007, [2]Salem-636007.

Abstract:-

Cloud computing is a globalised concept and there are no borders within the cloud. Because of the attractive features of cloud computing many organizations are using cloud storage for storing their critical information. The data can be stored remotely in the cloud by the users and can be accessed using thin clients as and when required. One of the major issue in cloud today is data security in cloud computing. Storage of data in the cloud can be risky because of use of Internet by cloud based services which means less control over the stored data. One of the major concern in cloud is how do we grab all the benefits of the cloud while maintaining security controls over the organizations assets. The main objective of this research paper is Data Deduplication, which is an efficient data compression technique for eliminating the duplication copies. This work recovers the two main issues 1.Storage issue and 2.Security issue in cloud computing. We have proposed a new technique for duplication check by giving different privileges to users. And new Convergent Encryption is introduced to maintain efficient security level when data is outsourced. The concept of tag and tokens were introduced. All these process is carried in Hybrid clouds which includes both the public and private cloud.

Keywords:-Cloud, Security, Hybrid,Deduplication, Encryption.

1. INTRODUCTION

Cloud computing is receiving a great deal of attention, both in publications and from individuals to researchers. Cloud computing is a Internet based computing where virtual shared servers provides software, infrastructure, platform devices and other resources to customers on pay-as-you-use basis. The cloud makes it possible to access the information from anywhere and at any time across the world unlike a computer which needs a physical location to access the information. This computing technology is mainly implemented where large amount of data are being processed which requires a huge storage space and high security standards. The main criteria are to have a proper Internet connection for the computing technique. There are many definitions today which attempt to address cloud from the perspective of academicians, architects, engineers, developers, managers, and consumers. This document focuses on a definition that is specifically tailored to the unique perspectives of IT network and security professionals. The keys to understanding how cloud architecture impacts security architecture are a common and concise lexicon, coupled with a consistent taxonomy of offerings by which cloud services and architecture can be deconstructed, mapped to a model of compensating security and operational controls, risk assessment and management frameworks, and in turn to compliance standards.

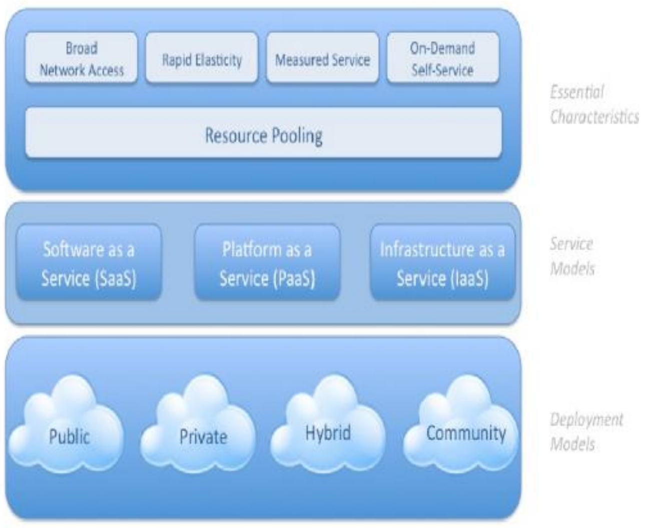

Figure 1.1: Cloud Computing Architecture

The earlier version of the Cloud Security Alliance's guidance featured definitions that were written prior to the published work of the scientists at the U.S. National Institute of Standards and Technology (NIST) and their efforts around defining cloud computing.

1.1 Hybrid Cloud & its Model

It can be either wired or wireless connection. As mentioned earlier the files can be accessed from anywhere across the world with the help of internet through clouds. The cloud computing technology doesn't have a standard physical architecture instead a third-party provides the cloud service and the consumers pay according to their usage. As the cloud service is used by most users with different privileges which provides the means of access, managing the huge amount of data is so difficult. So to make it more easily we have proposed a new technique called data Deduplication. Data Deduplication is new specialized compression technique to eliminate the duplicate copies of the repeating data. This actually increases the storage space and minimizes the number of data that must be sent. Instead of repeating the same copies a physical copy is made and it is referred when the data is repeated. Duplication can take place either at the file level or the block level. In the former level it eliminates the copy of the same file but in later level it eliminates the non-identical data. Though Deduplication copies bring a lot of benefits for cloud computing, we have a one more major issue which is security. Security issues should be eliminated or high security standards must be followed when data is being handled in cloud environment. Normally, we follow the basic encryption-decryption algorithm to encrypt the data while transferring and decrypting the data while receiving. The drawback in this technique is that different users may form different encryption keys through which deduplication are not efficiently possible. So, we propose a new technique called Convergent key algorithm to enforce the data confidentiality which allows the Deduplication technique feasible. Here the data copy is encrypted/ decrypted with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After the key generation and encryption process the users send the ciphertext to the cloud and retain the key value. Since the encryption process is carried out with the content of the data the same key value and same ciphertext will be generated when same data is accessed at different places. Regarding the security standards, as we discussed earlier we provide privileges to users for accessing the data, a separate protocol is formed where only the authorized users will be allowed to perform the duplicate check and a pointer from a server indicates them not to upload the file when duplicate copies is found. This process is carried out only to privileged users. Thus the convergent encryption allows the cloud to perform the Deduplication process and provides the ownership where only authorized users are allowed.

At a high level, our ultimate aim is to develop an enterprise network where the secured cloud service is carried out with large number of users. Who will use S-CSP and store the data with deduplication technique. So, in this network the deduplication section is carried out often which provides a rich storage space. There are three entities described in this model 1.Users, Private cloud and S-CSP in public cloud. The S-CSP carries out the deduplication process and stores the original data content. S-CSP – This is the entity which provides the storage service in the public cloud. It provides the data outsourcing and stores data on behalf of users. To minimize the storage cost the S-CSP carries out the deduplication technique where the duplicate copies are eliminated. In this model this S-CSP is considered to have high computational power and huge storage capacity.

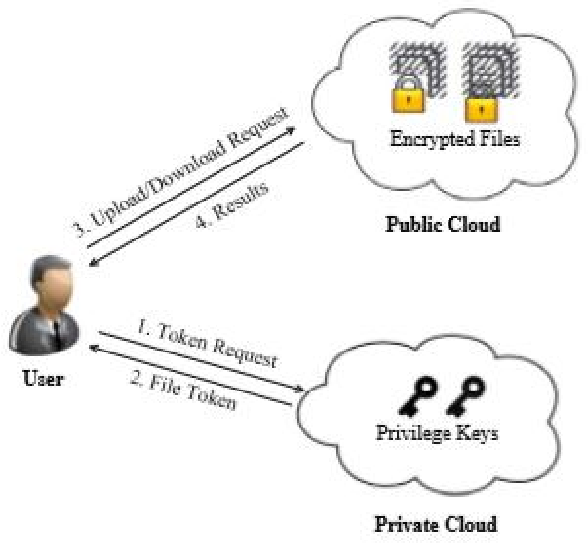

Figure 1.2: Architecture for Hybrid Cloud

Data users – A user is an entity who wants to outsource the data to S-CSP and access them later depending upon its requirement. So here certain privilege system is carried out where only certain users are allowed to either upload or download the file they need. Each file is maintained with unique convergent encryption key. Private Cloud - Compared with the traditional Deduplication architecture in cloud computing, this is a new entity introduced for facilitating user's secure usage of cloud service. Specifically, since the computing resources at data user/owner side are restricted and the public cloud is not fully trusted in practice, private cloud is able to provide data user/owner with an execution environment and infrastructure working as an interface between user and the public cloud. The private keys for the privileges are managed by the private cloud, who answers the file token requests from the users. The interface offered by the private cloud allows user to submit files and queries to be securely stored and computed respectively.

2. RELATED WORK

Luckily, the impossibility result [122] does not require that the Arbiter be involved in each transaction, but simply that the Arbiter exists. If Alice and Bob are both well behaved, there is no need for the Arbiter to do anything (or even know an exchange took place). **Micali [110], Asokan, Schunter and Waidner [5], and Asokan, Shoup and Waidner [7, 6]** investigated this optimistic fair exchange scenario in which the Arbiter gets involved only in case of a dispute. Two such protocols [7, 78] were analyzed in [141] (see also [12]).

Asokan, Shoup and Waidner (ASW) [7] gave the first provably secure and completely fair optimistic exchange protocol for exchanging digital signatures. Later on, Belenkiy et al. [16] gave a protocol for buying digital content in exchange for e-cash, building on top of the ASW protocol. They provided an optimization for the Arbiter so that, unlike in the ASW protocol, the amount of work that the Arbiter is required to do depends only logarithmically on the size of the file. They also assume there is an additional TTP (which we call the Tracker) that provides a means of verification that the file actually contains the right content (e.g., using hashes). Such entities certifying hashes already exist in current BitTorrent systems [52].

Belenkiy et al. [16] used e-cash (introduced by Chaum [50]), in particular, endorsed e-cash [44] in their constructions. The reason is that other forms of payments (signatures or electronic checks used in [7, 107]) do not provide any privacy. In our protocols, any form of payment can be employed, but we will also use endorsed e-cash in our sample instantiation since it is efficient and anonymous. See Section 2.7 for more discussion on employing different payment systems.

Jing-Jang Hwang et al. [22], has proposed a business model for cloud computing for data security using data encryption and decryption algorithms. In this method cloud service provider has responsible for data storage and data encryption/decryption tasks, which takes more computational overhead for process of data in cloud server. The main disadvantage of this method is, there is no control of data for data owner i. e, data owner has completely trusted with cloud service provider and he has more computational overhead.

Junzuo et al. [23], proposed an Attribute Based Encryption (ABE) and verifiable data decryption method to provide data security in cloud based system. They have been designed the data decryption algorithm based on the user requested attributes of the out sourced encrypted data. One of the main efficiency drawbacks of this method is, cloud service provider has more computational and storage overhead for verification of user attributes with the outsourced encrypted data. While introducing third party auditor we can reduces the storage, computation, and communication overheads of the cloud server, which improves the efficiency of the cloud data storage.

Fatemi Moghaddam et al. in [24], discussed the performance of six different symmetric key RSA data encryption algorithms in cloud computing environment. They have proposed two separate cloud servers; one for data server and other for key cloud server and the data encryption and decryption process at the client side. The main drawback of this method is to maintaining two separate servers for data security in cloud, which creates a more storage and computation overheads.

Qiang Guoet.al [19] gives the unique definition for trust in cloud computing and various issues related to trust are discussed here. An extensible trust evaluation model named ETEC has been proposed which includes a time-variant comprehensive evaluation method for expressing direct trust and a space variant evaluation property for calculating recommendation trust. An algorithm based on ETEC model is also shown here. This model also calculates the trust degree very effectively and reasonably in cloud computing environments. Other approaches to protect data privacy in a cloud environment include using direct encryption and proxy re-encryption. In these cryptographic schemes, data is allowed to be encrypted directly to the users with whom the Owners wish to share the data [20], [21]. There are numerous security issues for cloud computing as it encompasses many technologies including networks, databases, operating systems, virtualization, resource scheduling, transaction management, load balancing, concurrency control and memory management.

3. PROPOSED SYSTEM & ITS CONTRIBUTIONS

Since we have proposed the efficient way to minimize the storage cost. We have to concentrate on security aspects also because this is the another major issue in cloud service. So, in this paper we have compared two standard encryption algorithms, 1.SHA-1(Secure Hash algorithm) and 2. HMAC-(Hash based message authentication code).

SHA-1 (Secure Hash Algorithm)is a most commonly used from SHA series of cryptographic hash functions, designed by the National Security Agency of USA and published as their government standard.SHA-1 produce the 160-bit hash value. Original SHA (or SHA-0) also produce 160-bit hash value, but SHA-0 has been withdrawn by the NSA shortly after publication and was superseded by the revised version commonly referred to as SHA-1.

SHA-1 produces a message digest .Normally The input data is often called the message, and the hash value is often called the message digest or simply the digest. A message digest can also serve as a means of reliably identifying a file. Each hashing function forms a unique key depend on the file size and content produce in the file .even if there is a slight changes in the file the key for the file will be changed completely for the whole file. The main entities in the proposed algorithm are cloud users, cloud storage server, cloud manager, key splitter servers, share holder servers, security servers, log editor which are defined in detail as follows:

1. **User:** The user can create, update and delete his/her profile, store and retrieve the data.

File_Info	UsersShare(xu)	UsersShare(f(xu))

2. **Cloud Storage Server:** It is a model of data storage on virtualized storage pools or servers located remotely. Cloud storage can be used by users to store their data. Users can buy storage capacity from the cloud hosting companies. The main responsibilities of cloud storage server are storing the encrypted document, storing the splited encryption key values for the purpose of key Management.

3. **Key Management Server:** Key splitter server splits the encryption keys into different shares and store the splited keys in different share holder servers.

CloudUser_Id	File_Info	ShareHldrServer_Id

4. **Share Holder Server:** These servers store the shares for the different keys for different users. Share holders can be of two types. Primary share holder directly receives the shares from the cloud manager. Secondary share holders are the share holders at the leaf level and these share holders receive their shares through primary share holders.

CloudUser_Id	File_Info	xi	f(xi)

5. **Log editor:** It checks the share holder servers timely to see if the shares are getting modified.

ShareHldServer_Id	CloudUser_Id Value(t-60)	Value(t)	Value_Status

6. **Security server:** It has the encryption decryption algorithm.

Encryption process

Step 1- Split the letter of modified plaintext.
Step 2- Assign the position (i) of the letter.
Step 3- Generate the ASCII value of plaintext letter.
Step 4- E=(p+k+i) p-plaintext, k-shared key, i-position
Step 5- Generate the ASCII character of the corresponding decimal value in the result from the above given formula. This would be the cipher text.

Decryption process

Step 1- Generate the ASCII value of the cipher text character.
Step 2- Same encryption key is used.
Step 3- Assign the position i of the cipher text.
Step 4- D=((c-k-i)+256) p-plaintext, k-shared key, i-position.
Step 5- Generate the ASCII character of the corresponding decimal value in the result from the above given formula. This would be the original plain text.

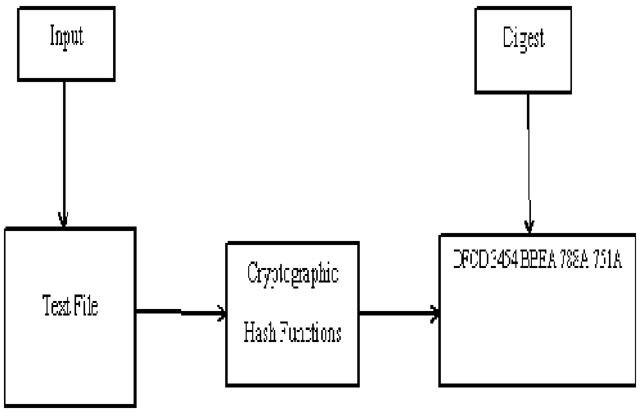

Figure 3.1: Example of Proposed Structure

Algorithm Framework

Append padding bits

File is "padded" with a 1 and as many 0's as necessary to bring the content length to 64 bits fewer than an even multiple of 512.

Append Length

64 bits are appended to the end of the padded contents. These bits hold the binary format of 64 bits indicating the length of the original file

Prepare Processing Functions

SHA1 requires 80 processing functions defined as:

- f(t;B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)
- f(t;B,C,D) = B XOR C XOR D (20 <= t <= 39)
- f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)
- f(t;B,C,D) = B XOR C XOR D (60 <= t <= 79)

Main loop

```
for i from 0 to 79
if 0 ≤ i ≤ 19 then
if = (b and c) or ((not b) and d)
k = 0x5A827999
else if 20 ≤ i ≤ 39
f = b xor c xor d
k = 0x6ED9EBA1
else if 40 ≤ i ≤ 59
f = (b and c) or (b and d) or (c and d)
k = 0x8F1BBCDC
else if 60 ≤ i ≤ 79
f = b xor c xor d
k = 0xCA62C1D6
temp = (a leftrotate 5) + f + e + k + w[i]
e = d
d = c
c = b leftrotate 30
b = aa = temp
```

HMAC

Hash-based message authentication code (HMAC) is a mechanism for calculating a message authentication code involving a hash function. This can be used to verify the integrity and authenticity of a message. HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and on the size and quality of the key.

3.1. COMPARISON OF SHA-1 AND HMAC

- SHA-1 generates a fixed size output of 20-bytes for an arbitrarily long message; but so does an HMAC when it uses SHA-1
- In HMAC Server side implementations are few in number, and also very inconsistent.
- SHA-1 uses an iterative algorithm. It generates digests by first splitting contentsinto blocks of 64 bytes and, one after the other, combining those blocks together to generate the 20 byte digest.
- SHA-1 uses padding that incorporates the length of the original file .Suppose the original length of the file is 10 bytes and the modified one is 15 bytes.
- If you decide to build the own API's using HMAC, It will be very hard because a single character difference will result in a completely different value.
- Time consumption is comparatively less compared to HMAC
- It's very difficult to compute HMAC for larger files
- In some HMAC fails to access the tokens of the text files.
- HMAC uses SHA-1 for its complete implementation.

4. IMPLEMENTATION & RESULTS

In this section, we illustrate some of the performance benefits of our proposed scheme. An initial simulation scenario has one parent node and five heterogeneous child nodes, each with different transition probabilities, states, and cost matrices. We increase the number of nodes up to 30 in different simulation scenarios. We compare the performance of the proposed scheme with an existing scheme, in which PKG nodes are selected randomly without consideration of the security context. Selecting a node under attack or a compromised node to function in the PKG process would pose a risk to the network security. By contrast, the proposed scheme takes into account the security conditions derived from intrusion detection systems to select the best nodes for reconstructing the full secret. In our scheme, the nodes with low security levels will be eliminated from reconstructing the full secret. Therefore, our scheme will have better performance than the existing scheme. In addition, the proposed scheme also takes into account the energy levels of the nodes in order to improve overall network lifetime and functionality.

4.1. Performance Evaluation parameters

In this Section, we first analyze our proposed approach in high level for satisfying security service requirements, and then provide the analysis of communication overhead and computational complexity. We also present simulation results.

i) Cost Analysis

Fig. 6.1 shows the cost comparison when there are more nodes in the network. With the number of available nodes in the network increases from 6 to 30, the cost of all schemes becomes lower since there are more nodes that can be selected. The cost of the proposed scheme is shown to be lower than existing scheme in all circumstances.

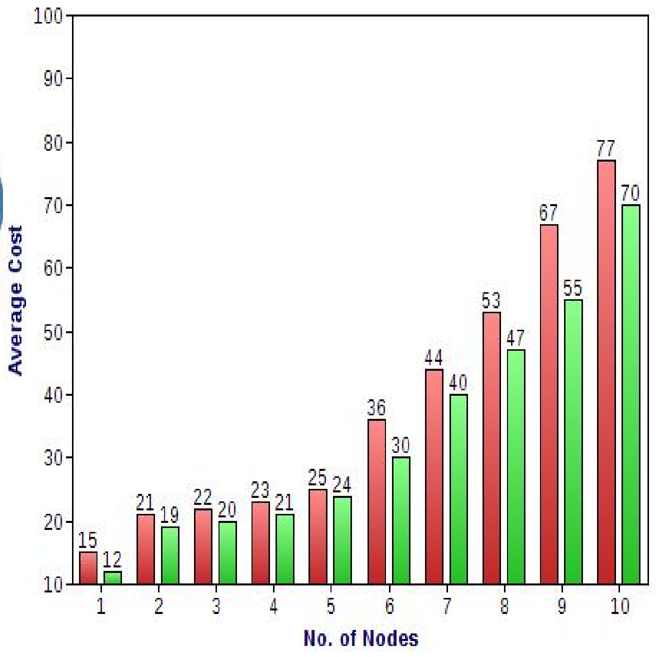

Figure 4.1: Cost Comparison

ii) Comparison Key Extract

However, our test shows its performance is quite low. Even if the master key length is only 256 bits, the encryption/decryption speed is below 1KB/s, which cannot be acceptable in most cases. Moreover, the size of ciphertext is thousands times that of plain text. This conclusion is obvious, since for each bit of the message, the Encrypt algorithm returns two P-bit numbers. Therefore, considering both the time and space cost.

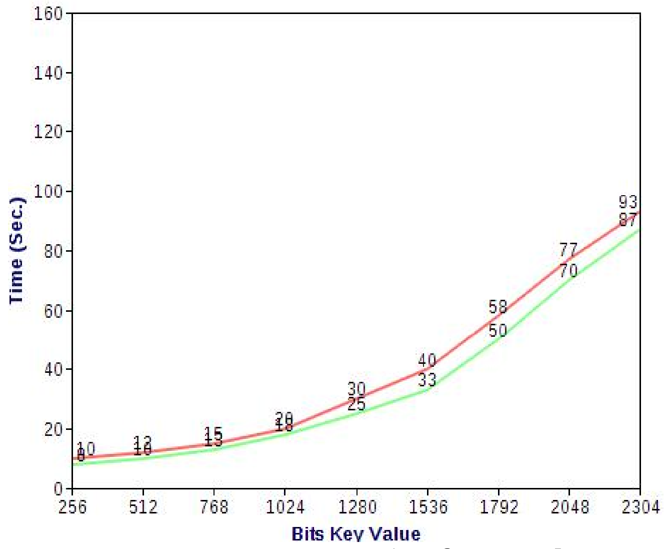

Figure 4.2: Key Extraction from nodes

iii) Encryption Time

Threshold Cryptography requires nodes to obtain authentic system parameters. But as soon as a node has obtained these, it may encrypt to all other nodes, and also check signatures generated by all other nodes — no certificates are required.

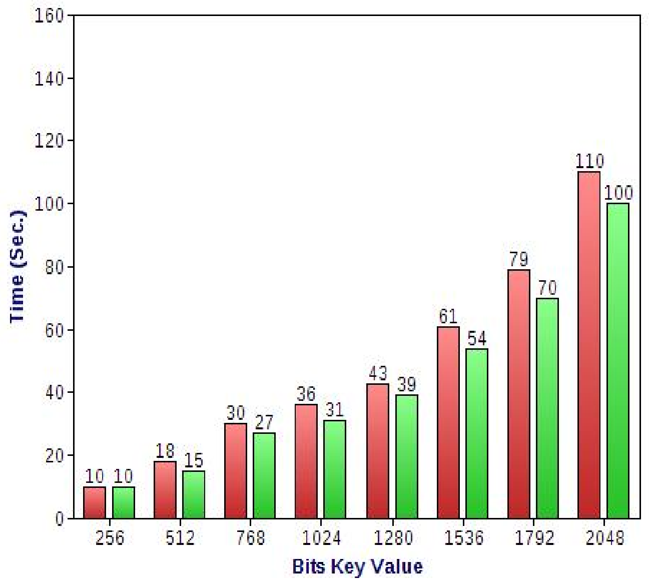

Figure 4.3: Encryption Time from nodes

iv) Decryption Time

In an IBC, the PKG will know the private keys of all the users, so the PKG is more trusted in this sense. Furthermore, when the private key of a user is to be issued from the PKG, a confidential channel must be available. Otherwise, the private key may be compromised. This is not a problem in a PKI because only public keys are transmitted.

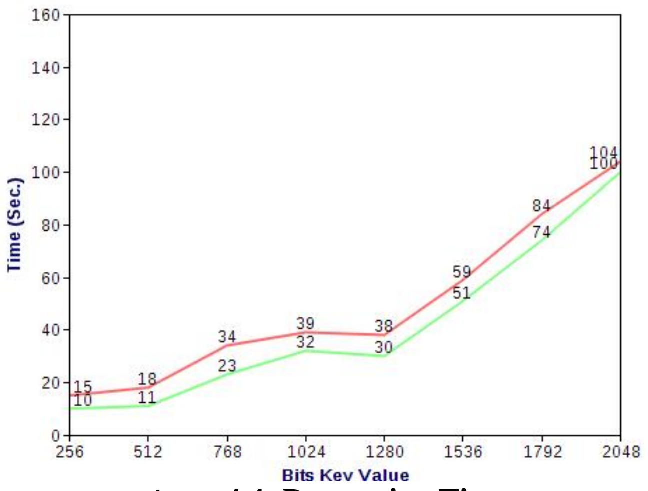

Figure 4.4: Decryption Time

v) Ciphertext Overhead

While the conventional PKI based key management approaches assume each node's public/private key pair is self-generated, and the public key is propagated in the network. In order to identify each node, the public key has to be signed by a trusted certificate authority (CA). The certificates are also required to spread in the network, so that each node can get other nodes' certificate. Propagating these public keys and certificates consumes a lot of network bandwidth, and also causes a large network/connection setup delay.

CONCLUSIONS

Key management is the toughest part to manage in cryptosystems. In the cloud platform, there is always a possibility of insider attack or outsider attack. Keys can be accessed or stolen by employees without the knowledge of end users. Our aim is to provide secrecy to the data as well as keys that are stored in cloud systems. Our proposed technique provides better data security and key management in cloud systems. This technique also provides better security against byzantine failure, server concluding and data modification attacks. The cryptographic techniques always play a major role in the design of each stage of the key management. The art of the design can be better evaluated from the conceptual level to the implementation of the simulation study. The security of design will be dissected more by the research community and the new design will come out quickly and easily reusable as popular "design patterns" using cryptography terminologies.

REFERENCES

[1]. C. Adjih, T. Clausen, A. Laouiti, P. M¨uhlethaler, and D. Raffo. Securing the OLSR protocol. In IFIP Annual Mediterranean Ad Hoc Networking Workshop, pages 25–27, 2003.

[2] M. Akane, H. Kato, Y. Morikawa, Y. Nogami, and Y. Sakemi. Integer Variable _-Based Ate Pairing. In Pairing 2008, volume 5209 of Lecture Notes in Computer Science, pages 178–191. Springer-Verlag, 2008.

[3] T. R. Andel and A. Yasinsac. Surveying Security Analysis Techniques in MANET Routing Protocols. IEEE Communications Surveys & Tutorials, 9(4):70–84, 2007.

[4] P. G. Argyroudis and D. O'Mahony. Secure Routing for Mobile Ad hoc Networks. IEEE Communications Surveys & Tutorials, 7(3):2–21, 2005.

[5] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for Key Management — Part 1: General (Revisited). NIST Special Publication 800-57, March 2007.

[6] D. Beaver. Foundations of Secure Interactive Computing. In Advances in Cryptology – CRYPTO '91, volume 576 of Lecture Notes in Computer Science, pages 377–391. Springer-Verlag, 1992.

[7] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In ACM Conference on Computer and Communications Security, pages 62–73, 1993.

[8] I. F. Blake, G. Seroussi, and N. P. Smart, editors. Advances in Elliptic Curve Cryptography, volume 317 of London Mathematical Society Lecture Note Series. Cambridge University Press, 2005.

[9] G. R. Blakley. Safeguarding cryptographic keys. In Proc. AFIPS 1979 National Computer Conference, volume 48 of AFIPS Conference proceedings, pages 313–317. AFIPS Press, 1979.

[10] A. Boldyreva. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In International Workshop on Practice and Theory in Public Key Cryptography, volume 2567 of Lecture Notes in Computer Science, pages 31–46. Springer-Verlag, 2003.

[11] D. Boneh and M. Franklin. Efficient generation of shared RSA keys. Journal of the ACM, 48(4):702–722, July 2001. 96 REFERENCES

[12] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In Advances in Cryptology – CRYPTO 2001, volume 2139 of Lecture Notes in Computer Science, pages 213–229. Springer-Verlag, 2001.

[13] D. Boneh, C. Gentry, and M. Hamburg. Space-Efficient Identity Based Encryption Without Pairings. In 48th Annual IEEE Symposium on Foundations of Computer Science, pages 647–657. IEEE Computer Society, 2007.

[14] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In Colin Boyd, editor, Advances in Cryptology – ASIACRYPT 2001, volume 2248 of Lecture Notes in Computer Science, pages 514–532. Springer-Verlag, 2001.

[15] J. C. Cha and J. H. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. In Y. Desmedt, editor, Public Key Cryptography, volume 2567 of Lecture Notes in Computer Science, pages 18–30. Springer-Verlag, 2003.

[16] D. Chaum, C. Crepeau, and I. Damg°ard. Multiparty unconditionally secure protocols. In Proceedings of the 20th Annual ACM Symposium on the Theory of Computing, pages 11–19. ACM Press, May 1988.

[17] L. Chen and J. Malone-Lee. Improved Identity-Based Signcryption. In Public Key Cryptography - PKC 2005, volume 3386 of Lecture Notes in Computer Science, pages 362–379. Springer-Verlag, 2005.

[18] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In 26th Annual Symposium on Foundations of Computer Science, pages 383–395. IEEE Computer Society, 1985.

[19] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.

http://www.ietf.org/rfc/rfc3626.txt (2008-08-24).

[20] C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In Cryptography and Coding, volume 2260 of Lecture Notes in Computer Science, pages 360–363. Springer-Verlag, 2001.

[21] I. Damg°ard. Secret Sharing. Unpublished manuscript, 2006. http://www.daimi.au.dk/ ivan/SecretSharing.pdf (2008-12-19). REFERENCES 97

[22] V. Daza, P. Morillo, and C. R`afols. On Dynamic Distribution of Private Keys over MANETs. Electronic Notes in Theoretical Computer Science, 171(1):33–41, 2007.