An Enhanced Security Enabled Sharing of Protected Cloud Storage Services by Trapdoor Commitment Based on RSA Signature Assumption

D. Palanikkumar and M. Sowmya varshini

Abstract--- *Cloud computing is a technology which allows the users to share the data and the applications over the internet. An efficient sharing of secure cloud storage services (ESC) scheme allows the users to share the messages in hierarchical order. If a receiver wants to retrieve a message when he is using a system with limited bandwidth, CPU and memory, then the ESC scheme may not work well. The computational cost for decryption is too high, because the receiver does not perform any partial decipherment. Based on the RSA signature, a Trapdoor Commitment scheme is proposed to improve the security level. This scheme allows the receiver to create a short trapdoor and send it to the Cloud Service Provider (CSP) before retrieving files or messages for finding out part of the cipher text. This scheme allows the sender to participate in the partial signature, so as to reduce the computational cost.*

Keywords--- *Cloud Computing, Partial Decipherment, Partial Signature, Trapdoor Commitment*

I. INTRODUCTION

CLOUD computing means provision of services in a timely, on-demand manner, allows scaling up and down of resources. It is nothing but a virtualization of computing program through an internet connection rather than installing application. The beauty of cloud computing is resource sharing. This concept helps the cloud service providers to attain optimum utilization of resources and it is designed in such a way that it provides the flexibility for sharing the application as well as other network resources.

Cloud services can be divided into three stacks as IaaS, PaaS and SaaS. Infrastructure as a Service (IaaS) is the base layer of the cloud stack. It offers visualized computing resources. Platform as a Service (PaaS)offers an environment for developing an application. Software as a Service (SaaS) is the top most layer of the cloud computing stack which directly consumed by the end-users.

D. Palanikkumar, Assistant. Professor, Department of CSE, Anna University of Technology, Coimbatore, India, E-mail: palani.journals@gmail.com

M. Sowmya varshini, Assistant. Professor, Department of CSE, Shanmuganathan Engineering College, Arasampatti, Pudukkottai, India. E-mail: sowmiya.varshini@gmail.com

DOI: 10.9756/BIJRCE.1504

These cloud storage services enable users to access data in a cloud anytime and anywhere, using any devices, in a pay-as-you-go manner. In a user hierarchy, the upper-level user may wish to read all the files stored in the cloud, whereas only the qualified lower-level users can read the files. The upper-level user can store all the important files in the cloud. Although some important files are stored in the cloud, the cloud service provider has no idea of any information regarding the files.

The concept of Attribute Based Encryption (ABE) [2],[6],[7] schemes can enable a sender to encrypt a message to multiple recipients. In this scheme, totally there are two parties: attribute authorities and users, where the attribute authorities can generate the secret keys for all the users who are all in the same level. The message can be encrypted and decrypted by multiple recipients who are all in the same level. But this ABE scheme cannot support user hierarchy.

Based on the above analysis, an efficient sharing of secure cloud storage services (ESC) scheme was proposed by Q. Liu et al.,[15]. The concept of this scheme supports a user hierarchy. To support this concept, a hierarchical identity-based architecture in cloud computing is proposed to embody the user hierarchy in the sharing of secure cloud storage services. In this architecture, the root Private Key Generator (PKG) delegates the upper-level user as the lower-level PKG to generate the secret keys for all the lower-level users. So the authentication and secret key transmission can be carried out locally. Next, the upper-level user can encrypt and store all the important files and messages only once in the cloud. Only the authorized receivers can get the files from the cloud by using their own private keys. Although some important files in the cloud, the cloud service provider has no idea of any information regarding the file. The ESC scheme is collusion resistant. In this scheme, the lower-level user/receiver does not perform any partial decipherment. Also the lower-level user requires too much computational cost for decrypting the file. If a receiver wants to retrieve a file when he is using a PDA with limited bandwidth, CPU and memory, then this scheme may not work well.

To solve this problem, the Trapdoor Commitment scheme is proposed by using RSA signature. The signature can be realized by using RSA algorithm. Using this algorithm, the sender can sign the plaintext by using his private key and the encryption can be performed by using receiver's public key. The decryption can be performed by the receiver using his private key and the public key of the sender to verify the signature. The main purpose of digital signature is guarantees

the confidentiality, authentication of the information that transformed in insecurity and unreliable network.

The Trapdoor Commitment helps the users to construct a secure signature scheme. This scheme allows the sender to first hide a value by computing a commitment, and then reveals the hidden value together with some related information to open the commitment so that the receiver can check whether the commitment is de-committed correctly. By this way, the security of the system can be improved. This will largely reduce the computational cost for decryption.

II. RELATED WORK

An Identity based encryption from the weil pairing concept was introduced by D. Boneh et al., [4]. The IBE system defines about the chosen ciphertext security for identity-based encryption. This system is based on bilinear maps between groups. The primary drawback of IBE system is that, the attacker may intrude while decrypting the ciphertext. The main drawback of IBE system is distribution of public keys for each user in the system.

To defeat D. Boneh et al., [4] concept, a new construction for hierarchical identity-based encryption (HIBE) system is proposed by Genry et al., [13] which has chosen ciphertext security in the random oracle model under the Bilinear Diffie-Hellman (BDH) assumption. In this, the public key generator (PKG) must verify the identity proofs of the user. If it is valid, then only establish secure channels for transmitting private keys. So the hierarchical identity- based encryption scheme (HIBE) allows root PKG to distribute the workload by delegating private key generation and identity authentication to lower-level PKG.

The construction by Boneh et al., [5] provides a HIBE system with constant size ciphertext. In their scheme, the length of the ciphertext and private key as well as time needed for decryption grows linearly in the depth of the hierarchy. This scheme provides selective ID secure in the standard model and also in the random oracle model. It provides secure encryption system with short ciphertext. The system supports limited delegation where users can be given restricted private keys that allow delegation to bounded depth.

In the first recent work, Gentry et al., [12] proposed hierarchical identity based encryption (HIBE) system that has full security for more than a constant number of levels by using identity based broadcast encryption with key randomization. In all prior HIBE systems, the security reductions suffered from exponential degradation in the depth of the hierarchy, so these systems were only proven fully secure for identity hierarchies of constant depth. Hence this system is secure for polynomially many levels because it offers tight proof of security.

Liu et al., [15] described about an efficient privacy preserving keyword search scheme in cloud computing. When the user needs to encrypt and decrypt files containing certain keywords, which depletes too much CPU capability and memory power. In order to protect the user data privacy and user queries privacy, this scheme enables the service provider for searching the certain keyword on encrypted files. This scheme will largely reduce the clients computational overhead and enables the cloud service providers to search the keywords on encrypted files to preserve the user data privacy and user queries privacy efficiently.

A new type of encrypted access control is introduced in a system for Ciphertext-Policy Attribute Based Encryption by Bethencourt et al., [2]. Here a set of attributes which specifies the user's private keys and a party encrypting data can specify a policy over these attributes specifying which users are able to decrypt. Any monotonic tree access structure is needed to represent the policies and it is resistant to collusion attacks in which an attacker might obtain multiple private keys.

Multi-authority attribute based encryption scheme proposed by Chase [6]. This scheme allows any polynomial number of independent authorities to monitor attributes and distribute secret keys. The sender can choose a number and a set of attributes for each authority; he can then encrypt a message such that a user can only decrypt if he has at least that specified number of the given attributes from each authority. This scheme can tolerate an arbitrary number of corrupt authorities.

For better performance, Chase et al., [7] has proposed Improving privacy and security in multi-authority attribute-based encryption. This system allows the authorities to combine their information with all of the user attributes, which unnecessarily compromises the privacy of the user. Multi-authority attribute-based encryption enables a more realistic deployment of attribute-based access control, such that different authorities are responsible for issuing different sets of attributes.

For more security, a trapdoor commitment is introduced by M. Fischlin [8] in Trapdoor Commitment Schemes and Their Applications. The trapdoor commitment scheme has both commitment phase and de commitment phase for improving security. The message can be encrypted in commitment phase and the receiver has to decrypt the message by using private key in de commitment phase. The trapdoors turn out to be very useful for the design of secure cryptographic protocols involving commitment schemes.

The security of practical two-party RSA signature scheme was introduced by M. Bellare et al., [1]. In a two-party RSA signature scheme, a client and server, each holds a share of RSA decryption exponent. Then, compute an RSA signature under public keys. This concept is applicable to password-based security, where client's part of decryption is based on password. This scheme provides a security based on the assumptions about RSA and the hash functions used on it.

A digital signature scheme secure against adaptive chosen-message attacks was given by S. Goldwasser et al., [12]. This scheme is based on the property of being robust against an adaptive chosen-message attack. This concept deals with that no one can able to forge the signature when he receives for message .This scheme is potentially practical because signing and verifying signatures are fast and signatures are compact too.

III. PROBLEM DEFINITION

A. Hierarchical Identity-Based Architecture

The hierarchical identity-based architecture is to symbolize the user hierarchy in the sharing of the secure cloud storage services (see Fig.1). This architecture consists of a root Public Key Generator (PKG) and multiple domains. At the top level, the root PKG is a Trusted Third Party (TTP). The upper-level users and the lower-level users constitute a domain. The root PKG will generate the system parameters for encryption and the secret keys for the upper-level users, which in turn generates the system parameters and the secret keys for the lower-level users. Hence authentication and secret key transmission can be carried out locally in a domain.

In a domain, a sender can specify multiple recipients for an encrypted file by taking the number and the public keys of those recipients as inputs of a hierarchical identity-based encryption, so that the sender and all authenticated recipients can decrypt the file by using their private keys. This can be done by encrypting the file only once and store the copy of that file in the cloud. The intended recipients can decrypt the file from the cloud using their private keys. Therefore, the sender can efficiently share the secure cloud storage services with all the receivers in a domain.

B. Definition of the ESC Scheme

Let Bob and all the users in Company A constitute a domain, denoted Dom A. Suppose there are N users U1,…,UN in Dom A, whose public keys are denoted as ID-tuplei = (IDBob, IDi) for $1 \leq i \leq N$. In DomA, when the sender S wants to encrypt a file to R uesrs U1, . . . , UR ($1 \leq R \leq N$), he sends the following message to the CSP:

MSGS2CSP = One2ManyEnc (params, R, ID-tuple1,. . . , ID-tupleR, f)

where params are the system parameters, R is the number of the intended recipients, ID-tuple1, . . . , ID-tupleR are the ID-tuples of U1, . . . , UR, respectively, and f is the file. One2ManyEnc is a hierarchical identity-based encryption algorithm.

C. Construction of the ESC Scheme

The existing Efficient sharing of Secure Cloud storage services (ESC) scheme allows the sender to encrypt the file by using his/her public key, and store the copy of the encrypted file only once in the cloud. The sender as well as all the intended recipients can decrypt the file by using his/her own private keys. This scheme has four contributions:

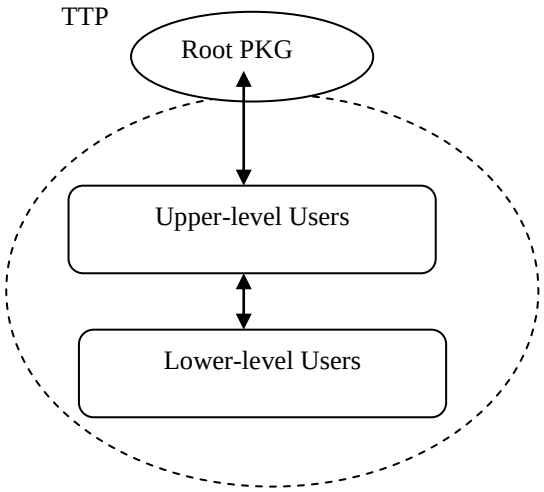

Figure 1: Hierarchical Identity-Based Architecture in Cloud

- The problem with efficient sharing of the secure cloud storage services scheme is that the user will enjoy a more scalable, and secure service.
- A hierarchical identity-based architecture in cloud computing is proposed to embody the user hierarchy in the sharing of the secure cloud storage services.
- The ESC scheme allows the user to encrypt the file/message only once and stores the copy in the cloud. But the sender and all the intended recipients can decrypt the message by using their private keys.
- The ESC scheme is collision resistant.

The ESC scheme consists of five polynomial time algorithms are as follows:

- RootSetup: The root PKG takes large security parameter as input to generate the system parameters and a root master key. The master key will be only known by the root PKG.
- DomSetup: The root PKG generates the secret keys for the lower-level PKGs, which, in turn generate the secret keys for the entities in their domains at the bottom level.
- One2ManyEnc: The sender takes system parameters, finite plaintext space, number of users and the ID-tuples of the recipients as inputs, and outputs a cipher text.
- UserDec: Bob takes system parameters, a private key, and the cipher text as inputs to recover the plaintext.
- RecipientsDec: The intended recipients takes system parameters, a private key, a master key, ID-tuple, and the cipher text as inputs to recover the plaintext.

The main drawback of this existing system is that we do not want TTP for generating a private key. The main reason is that the private key should not be revealed to any party including a partially trusted party. With that the computational overhead is high because receiver does not perform any partial decipherment. Also the lower-level user requires too much computational power for decrypting the file. If a lower-level user wants to retrieve the files/messages when he is using a PDA with limited bandwidth, CPU, and memory, this ESC scheme may not work well. The computational cost for

decrypting the file depends on the number of users in the hierarchy.

IV. IMPLEMENTATION

A. Trapdoor Commitment (TC) Scheme

The Commitment scheme allows the sender to first hide a value by computing a commitment, and then reveals the hidden value along with some information to open the commitment so that the receiver can check that the commitment is de committed correctly. So a secure commitment scheme should satisfy both the hiding property and the binding property. Once the commitment has been made, the sender cannot able to cheat the receiver by telling different value. At the same time the receiver is unable to know what value is committed actually.

In a Trapdoor Commitment (TC) scheme, there is one trapdoor that would allow the owner of the trapdoor to open the commitment. This scheme does not support binding property. It helps the users for the construction of secure signature scheme. There will a valid answer for de commitment that can be specified by the owner of the trapdoor usually the commitment receiver. The reason is that once getting such a valid answer to a commitment, the outsider cannot distinguish whether this answer is revealed by the sender. This is why trapdoor commitment scheme helps the users to improve the security.

The trapdoor commitment scheme consists of four algorithms, i.e., TCgen, TCcom, TCver, TCsim. The receiver runs the key generation algorithm TCgen to get a commitment public key and corresponding trapdoor. Given a value and the commitment public key, commitment algorithm TCcom generates a pair (com, dec), where com is the commitment to a value and dec is the related information used for de commitment. Next one is commitment verification algorithm TCver, it checks whether the value is valid to the commitment with respect to public key. Finally the simulation algorithm TCsim, allows the receiver to simulate a new answer for a commitment by using the trapdoor.

The Trapdoor Commitment scheme consists of four algorithms as follows:

- TCgen: The receiver runs the key generation algorithm TCgen to get a commitment public key pk and the corresponding trapdoor td.
- TCcom: Given a value r and the commitment public key pk, Commitment algorithm TCcom outputs a pair (com, dec), where com is the commitment to a value r and dec is the related information used to de
- TCver: A commitment verification algorithm TCver is used to check whether an answer (r, dec) is valid to a given commitment com with respect to public key pk.
- TCsim: A simulation algorithm allows the receiver, using the trapdoor td, to simulate a new answer (r', dec') for a commitment com when one answer (r, dec) is given.

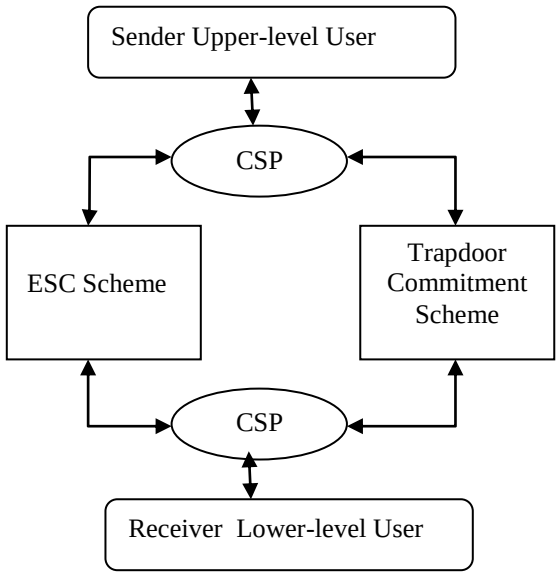

Figure 2: Construction of Trapdoor Commitment Scheme

Commit

The above Trapdoor Commitment scheme is formally proved to be secure under the strong RSA assumption.

B. Construction of the RSA-based Trapdoor Commitment Scheme

Our proposed work is that, in ESC scheme the sender can encrypt the file/message only once and store it in a cloud. Now the receivers can decrypt the files from the cloud. This exchange should be securely carried out in a hierarchy. For that, the receiver should send a short trapdoor to the Cloud Service Provider (CSP) for finding part of the cipher text without leaking any information about the plain text (see Fig.2). To reduce the cost, RSA-based partial signature concept helps the sender to send the message securely within a stipulated time.

The work is that, both the sender and the receiver have to register with TTP to send their signatures on message m. The registration is made with the help of their partial private keys. Based on the RSA signature in which first splits the sender S private key d into d1 and d2 so that $d = d1 + d2 \bmod \phi(n)$. Then, d2 is delivered to the TTP while keeps (d, d1, d2) as secrets.

If the sender wants to send their signature $\sigma S = h(m)d \bmod n$ to receiver R where h(.) is a cryptographically secure hash function. Now the sender sends partial signature $\sigma 1 = h(m)^{d_1} \bmod n$ to receiver and proves that σ1 prepared correctly in an interactive zero-knowledge way, enhanced by a trapdoor commitment scheme which depends on receiver's signature public key.

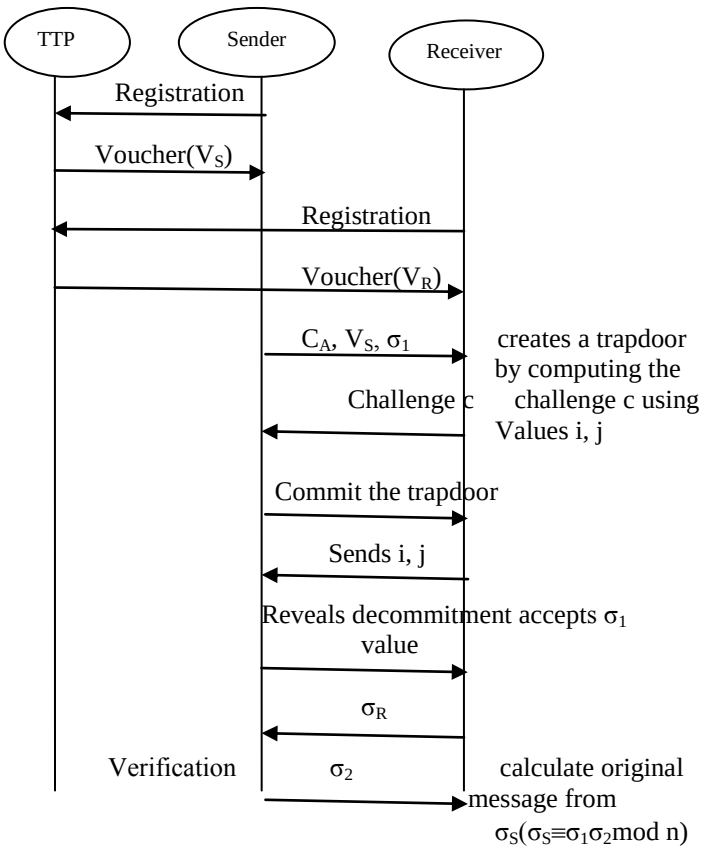

Figure 3: Working Process of RSA-based Trapdoor Commit Met Scheme

After that, the receiver sends their signature σR on message m to sender, the second partial signature $\sigma 2 = h(m)^{d_2} \bmod n$ is revealed by the sender. If the sender refuses to reveal the second partial signature, then the receiver will get it from TTP by sending request. After getting the sender's signature σS, the receiver now verifies that the signature is correct by using sender's public key i.e., σS = σ1 σ2 mod n. The original message can be extracted by the receiver using sender's public key by computing h(m)2 = (σ1 σ2)2e mod n. Fig.3 shows the overall working process of the RSA-based trapdoor commitment scheme.

i. Registration

Initially the sender needs to register with the Trusted Third Party (TTP). That is, the sender is required to get a long-term voucher VS from the TTP besides obtaining a certificate CS from a Certification Authority (CA).

The following procedures are required for a successful registration.

- Sender S sets an RSA modulus n = pq, where p and q are two large prime numbers and $\phi(n) = (p-1)(q-1)$.
- Then selects their random public key e, $1<e<\phi(n)$ and then calculates their private key d = e-1 mod $\phi(n)$.
- Sender needs to register their public key with a CA to get certificate CS, which binds her identity and public key (n, e) together.
- Sender randomly splits their private key d into d1 and d2, such that d = d1 + d2 mod $\phi(n)$ and computes e1 = d1-1 mod $\phi(n)$
- At the same time, sender generates a sample message-signature pair (ω, σω) where ω is a sample message and $\sigma\omega = \omega^{d_1} \bmod n$ is the signature of that sample message.
- Then, sender sends (CA, ω, σω, d2) to the TTP but keeps (d, d1, d2, e1) secret.
- Now the TTP checks that the sender's certificate is valid or not. If it is valid, TTP stores d2 securely, and creates a voucher VS which contains the TTP's signature (CA, ω, σω). This means that the TTP can issue a valid partial signature on behalf of sender by using secret d2.

The registration stage needs to be executed only once for a sufficiently long period. But the sender should register with the trusted third party (TTP) before they are served. The reason is that the TTP is unlikely to provide free service for settling disputes between users.

For enhancing efficiency, the sample message ω can be fixed as a constant. For each registration the TTP needs to keep a distinct secret d2 for each registered user. But this can be eliminated by some simple techniques i.e., by exploiting a secure symmetric key encryption algorithm and stores the result in its database. To extract a user's secret d2, the TTP needs to decrypt the corresponding record using the unique symmetric key.

ii. Trapdoor Commitment based on RSA Signature

We assume that there is a contract has been agreed between the sender and the receiver. This contract explicitly specifies the identities of sender, receiver, TTP, message along with reasonable deadline t. This scheme has the following procedures to follow.

1. First, the sender has to compute his/her partial signature $\sigma 1 = h(m)^{d_1} \bmod n$ and then sends the computed partial signature along with CA, and the voucher VS to the receiver.
2. The receiver now checks whether CA and VS is valid or not. Then the receiver has to verify the identities of sender, receiver and TTP are correctly specified as part of the contract.
   a) The receiver initiates a trapdoor with the sender after performing all those validations. For that the receiver picks two numbers at random and computes a challenge c by using partial signature σ1 and the sample message-signature σω to the sender.
   b) After getting c, sender calculates the response r and computes a trapdoor commitment algorithm TCcom(r, t) where t is a random number.
   c) The committed value is sent to receiver. To de commit the trapdoor, receiver must send the value

of that chosen random numbers for calculating the challenge c.

d) If the answer is positive, the sender reveals the respondence (r, t) to receiver to accept the partial signature σ1.

3. Now the receiver has to send their signature σR to the sender and it is verified by the sender.
4. If the receiver's signature is valid, then the sender will send their second partial signature σ2 to receiver.
5. The receiver has to verify that the second partial signature is valid. For that, it compares $\sigma S \equiv (\sigma 1\ \sigma 2) \bmod n$.
6. After getting the signature of the sender σS, the receiver can extract the original message from $\sigma S = h(m)^d \bmod n$.

In step 2(d), receiver accepts σ1 as valid and sends his signature σR to sender, since he is convinced that another partial signature σ2 can be revealed by either sender or the TTP. After that, if sender does not reveal the partial signature σ2 or only sends an invalid partial signature σ2 to receiver for a reasonable long period before the deadline t, then the receiver can route to the TTP to get the correct value of the partial signature σ2.

The trapdoor commitment scheme enhances the security by using partial signature σ1. Specifically, using commitment scheme the receiver can be forced to prepare the challenge correctly. Otherwise the receiver cannot get the response r. And the receiver cannot forward the intermediate results to convince an outsider of the validity of partial signature σ1 after execution of the RSA-based trapdoor commitment scheme. This scheme makes the receiver unable to collude with one or more outsiders during the execution by generating the challenge c. By using, RSA-based trapdoor commitment scheme the sender can efficiently send their message by using partial signature. This scheme largely helps the users to securely share the file.

V. Experimental Results

A. Performance Evaluation

In Q. Liu et al., [15], efficient sharing of secure cloud storage services scheme allows the sender to encrypt the file once and store the copy of the file in the cloud. Then the receiver can retrieve the file from the cloud. But depends on the number of levels of the user, the computational cost for the decrypting the file is too high. But, the RSA-based Trapdoor commitment scheme largely helps the users to decrypt the file with low computational cost.

In the comparison, we analyze the overheads of computation in both the schemes. We take the number of modular exponentiations as the computational cost since exponentiation is the most expensive cryptographic operation.

For comparison, we assume that the length of RSA modulus n is 1200 bit, and that the hash function h(.) has 160-bit fixed output. Also assume that σB could be generated and verified by one modular exponentiation and for voucher VA by one modular exponentiation too.

In our proposed RSA-based Trapdoor Commitment scheme, the number of times to execute the exponentiation operation is low.

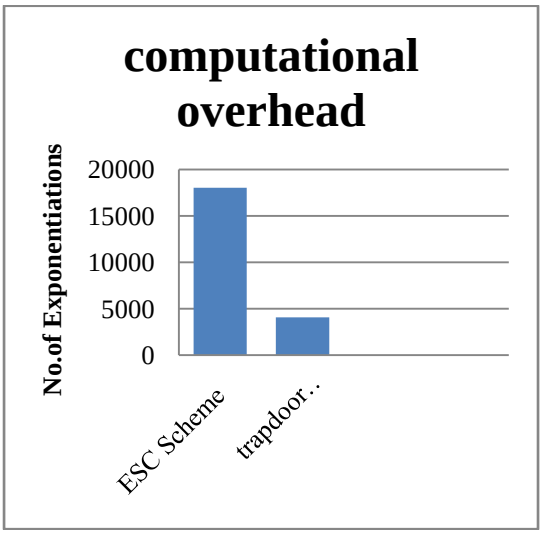

Figure 4: Computational Overhead

B. Security

Q. Liu et al., [15] efficient sharing of secure cloud storage services scheme is collision resistant only if the third party is trustable one. But in case of partially trusted or mistrusted third party, this ESC scheme may get failed, because the secret keys are generated by the Trusted Third Party (TTP). By using the secret keys only, the receiver can decrypt the messages. But the implementation of Trapdoor Commitment with RSA-based partial signature largely helps the users to share the files more securely than in the ESC scheme. Instead of knowing the full private key, the TTP knows only the part of sender's private key.

Our scheme overcomes the security flaw in ESC scheme. Namely, if sender is honest, the TTP cannot derive sender's private key d from d2 and other public information. Otherwise, the RSA signature can be broken as follows. For any RSA public key (n,e), an attacker first chooses an even number d2, and then inquiries the signing for a polynomial number of adaptively chosen messages m. Then, from the corresponding answers σ, the attacker computes $\sigma 1 = \sigma (h(m)^{d_2})\text{-}1 \bmod n$. Finally, the attacker calls the TTP as a subroutine to get the private key d. In fact, the above reduction is also valid to prove that except sender itself, anybody cannot forge a valid partial signature σ1 for a new message. So, we conclude that our scheme is more advantageous than existing scheme.

VI. Conclusion

Cloud computing is one of the current most important and promising technologies. In this paper, based on the standard RSA-based signature, we proposed a new idea of RSA-based Trapdoor Commitment in a user hierarchy. By using this idea, the sender can encrypt the message and the receiver can decrypt the message in a secured way. The efficiency of our proposed work is by computing the computational cost. The computational overhead for our proposed Trapdoor

Commitment scheme is low. But in case of existing efficient sharing of secure cloud storage services only allows the users to encrypt the message in a hierarchy with a high computational cost. The security level of RSA-based Trapdoor Commitment scheme can be improved by using sender's partial signature concept.

REFERENCES

[1] M. Bellare and R. Sandhu, "The Security of Practical Two-Party RSA Signature Schemes", 2001 Available: http://www-cse.ucsd.edu/users/mihir/papers/

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption", In Proceedings of IEEE ISSP, 2007.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing", In Proceedings of CRYPTO 2001, Volume 2139 of LNCS, 2001.

[4] D. Boneh, X. Boyen, and E. Goh, "Hierarchical identity based encryption with constant size ciphertext", in Proceedings of EUROCRYPT 2005,Volume 3494 of LNCS, 2005.

[5] M. Chase "Multi-authority attribute based encryption", In Proceedings of TCC 2007, Volume 4392 of LNCS, 2007.

[6] M. Chase and S. Chow,"Improving privacy and security in multi-authority attribute-based encryption", In Proceedings of ACM CCS 2009.

[7] M. Fischlin, "Trapdoor Commitment Schemes and Their Applications,"

[8] PhD. Dissertation, Fachbereich Mathematik, Johann Wolfgang Goethe-Universität Frankfurt am Main, Frankfurt, Germany, 2001.

[9] S. Goldwasser, S.Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," SIAM J. Comput., Vol. 17, no. 2, Pp. 281–308, Apr. 1988.

[10] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption", In Proceedings of EUROCRYPT 2002, volume 2332 of LNCS, Pages 466-481, 2002

[11] Q. Liu; G. Wang; J. Wu, "An efficient sharing of secure cloud storage services",Computer and Information Technology, 2010.

[12] S. Yu, K. Ren, and W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks", In Proceedings of IEEE INFOCOM 2009.

Palanikkumar Durai Thirunavukkarasu is an Assistant Professor in Anna University of Technology, Coimbatore. He has received his Master's degree from Government College of Engineering, Tirunelveli. Submitted Synopsis of his PhD programme in the area of QoS in Web Services. His research interest includes QoS in web services, cloud computing, business intelligence and datawarehousing/mining.

Sowmya varshini. M is an Assistant Professor in Shanmuganathan Engineering College, Arasampatti. She has completed her Bachelor of Engineering in Arulmigu kalasalingam college of Engineering, Krishnankoil. Completed her Master of Engineering in Anna University of Technology, Coimbatore, India. Her area of interest is cloud computing.