An Improved Security Enabled Distribution of Protected Cloud Storage Services by Zero-Knowledge Proof based on RSA Assumption

M. Sowmya Varshini
PG Scholar
Department of CSE
Anna University of Technology, Coimbatore

D. Palanikkumar
Asst.Professor
Department of CSE
Anna University of Technology, Coimbatore

G. Rathi
PG Scholar
Department of CSE
Anna University of Technology, Coimbatore

ABSTRACT

Cloud computing dynamically provides high quality cloud-based secure services and applications over the internet. The efficient sharing of secure cloud storage services (ESC) scheme which allows the upper-level user to share the secure cloud storage services with multiple lower-level users. In hierarchical identity-based architecture, the sender needs to encrypt a file only once and store only one copy of the corresponding ciphertext in a cloud. The lower-level user needs to decrypt a file which will increase the computational overhead, because the lower-level user does not perform any partial decipherment. In this paper, we propose a Trapdoor commitment scheme that enables a lower-level user to send a short trapdoor to the cloud service provider before retrieving files. This scheme allows the CSP to participate in the partial decipherment, so as to reduce computational overhead on the users without leaking any information about the plaintext. If a lower-level user wants to retrieve a file with limited bandwidth, CPU and memory, the trapdoor which will largely helps to reduce computational power.

Keywords

Hierarchical identity-based encryption; secure storage; trapdoor; partial decipherment.

1. INTRODUCTION

Cloud computing is one of the current most important technologies, provides computation, data access and storage services. It provides some services to the user. Most of the cloud computing infrastructures consist of services delivered through shared data-centers. 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 existing Attribute Based Encryption (ABE) [2] [6] [7] schemes can enable a sender to encrypt a message to multiple recipients in an efficient way. But the ABE system cannot support the user hierarchy. In an ABE system, there are two parties: attribute authorities and users, where all the users at the same level. The attribute authorities are responsible for generating the secret keys for all the users at the time of authorization. Obviously, there exists an user hierarchy.

Based on the analysis, an efficient sharing of secure cloud storage services (ESC) scheme was introduced by Q. Liu et al [17]. This scheme consists of four contributions:

1) The efficient sharing of the secure cloud storage services allows a user to enjoy a more scalable and secure service.
2) A hierarchical identity-based architecture in cloud computing is proposed to symbolize the user hierarchy in the efficient sharing of the secure cloud storage services.
3) The ESC scheme allows the upper-level user to store the file in cloud in the form of cipher text. The lower-level users can decrypt the file from the cloud using his/her private keys.
4) The proposed ESC scheme is collusion resistant.

In this scheme, computational overhead is high because they do not perform any partial decipherment. Also a lower-level user requires too much computational power for decrypting the file. If a lower-level user wants to retrieve the files when he is using a PDA with limited bandwidth, CPU, and memory, this ESC scheme may not work well. To solve this problem, the Trapdoor Commitment scheme is proposed. This scheme allows the user to send the trapdoor which enables the cloud service provider (CSP) to find out the part of the ciphertext without leaking any information about the file. Security of the system is high, so the overall performance is increased. This will largely reduce the computational cost for decryption.

2. RELATED WORK

An Identity based encryption from the weil pairing concept was introduced by D. Boneh et al [4]. The security of the IBE system defines about the chosen ciphertext security for identity-based encryption. The IBE system deals with bilinear map operations between groups. The primary drawback of IBE system is that, the attacker may intrude while decrypting the ciphertext. One more drawback is distribution of separate 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

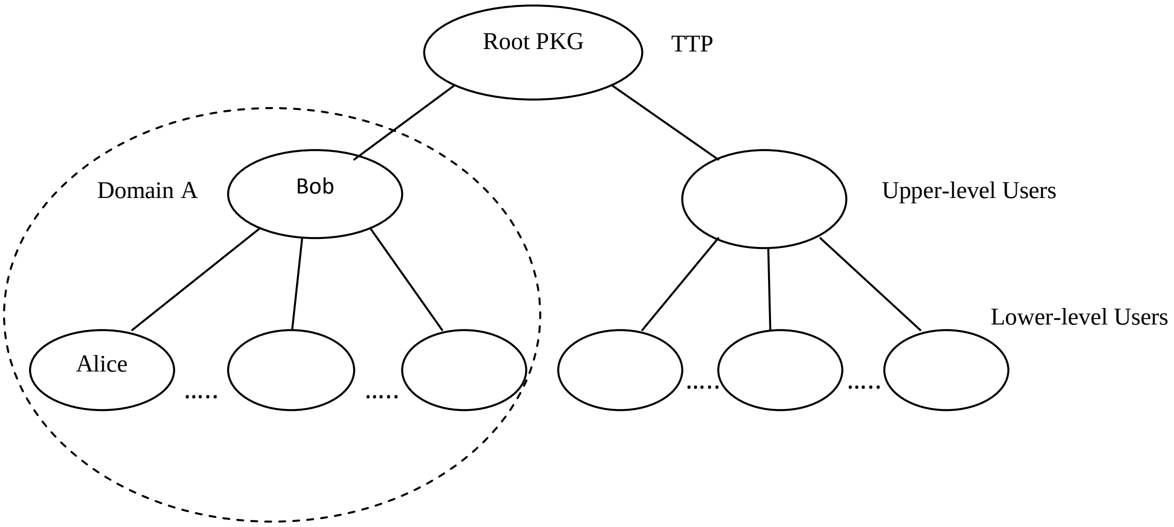

Fig 1: Hierarchical Identity Based Architecture

ciphertext and private key as well as time needed for decryption grows linearly based on 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.

An efficient privacy preserving keyword searching scheme in cloud computing concept is described by Liu et al [16]. His concept is to reduce the CPU capability and memory power by searching a certain keyword in the files. This scheme 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.

A scheme of multi-authority attribute based encryption proposed by Chase [6] allows any polynomial number of independent authorities to monitor attributes and to distribute secret keys. In this scheme, the sender can specify how many set of attributes are assigned to each authority. Then only the user can decrypt the message if he has at least specified number of attributes. 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's attributes, which unnecessarily compromises the privacy of the user.

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 for encryption and de commitment phase for decryption to improve the 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]. There are two notions are considered under this concept. The common-message protocols represent the abstraction, in secure two-party computation. This scheme is based on the deserve analysis. This scheme is more secure under the chosen-message attack.

A digital signature scheme provides security against adaptive chosen-message attacks given by S. Goldwasser et al [14]. This scheme is fully secure under an assumption against an adaptive chosen-message attack. The concept of this scheme is that no one can able to forge the signature of the users when he receives a message .This scheme is potentially practical because signing and verifying signatures are too fast.

3. PROBLEM DEFINITION

3.1 Definition of the ESC scheme

Let Bob and all the employees in Company A constitute a domain, denoted Dom_A. Suppose there are M employees E_1, …, E_M in Dom_A, whose public keys are denoted as ID-tuple$_i$ = (ID_{Bob}, ID_i) for $1 \leq i \leq M$. In Dom_A, when the sender X wants to encrypt a file to N employees E_1, . . . , E_N ($1 \leq N \leq M$), he sends the following message to the CSP: MSG_{X2CSP} = One2ManyEnc(params, N, ID-tuple$_1$,. . . , ID-tuple$_N$, f), where params are the system parameters, N is the number of intended recipients, ID-tuple$_1$, . . . , ID-tuple$_N$ are the ID tuples of E_1, . . . , E_N respectively and f is the file. One2ManyEnc is a hierarchical identity-based encryption algorithm.

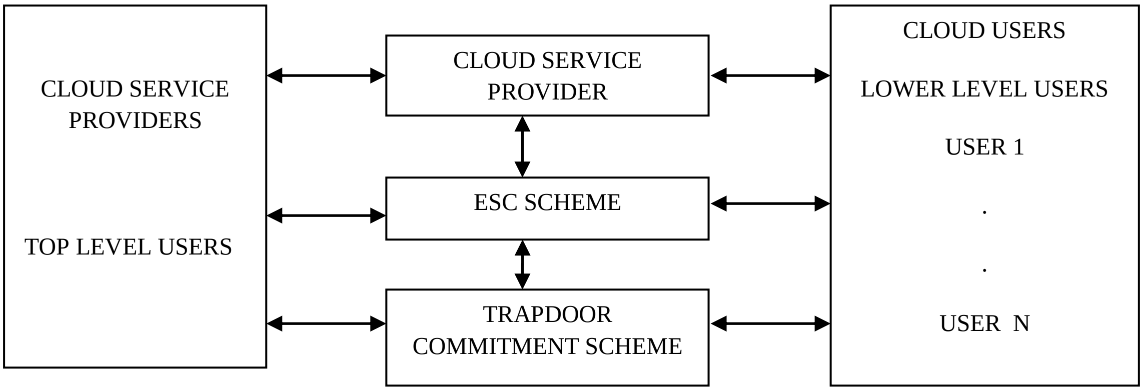

Fig 2: The working process of the Trapdoor Commitment scheme

3.2 Construction of the ESC scheme

The existing system introduces an efficient sharing of the secure cloud storage services (ESC) scheme is introduced by Q. Liu et al [17]. The Fig 1: shows the hierarchical identity-based architecture in cloud computing which represents the user hierarchy in the sharing of the secure cloud storage services. This architecture consists of a root PKG and multiple domains. The root PKG at the top level is a trusted third party (TTP). The members in a domain include a lower-level PKG and multiple entities, where the lower-level PKG at the second level is considered as the upper level user, and all the entities at the bottom level are the lower level users. In this hierarchical architecture, the root PKG generates the system parameters for the hierarchical identity based encryption system by D. Boneh et al [3] and 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. Therefore, Authentication and secret key transmission can be carried out, locally.

Definition 3.1(The ESC Scheme)

The ESC scheme consists of five polynomial time algorithms, given by Q. Liu et al [17] who just grab the idea from Boneh et al [4] scheme.

1) RootSetup: The root PKG takes a large security parameter as input to generate the system parameters and a root master key. The system parameters include a description of a finite plaintext space, and a description of a finite ciphertext space. The system parameters params will be publicly available, while the root master key is only known by the root PKG.
2) 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. A PKG takes system parameters, its private key, its master key, and ID-tuple$_i$ as inputs to generate a private key and a master key for any member with ID-tuple$_i$ in Domain A.
3) One2ManyEnc: The sender X takes system parameters, file which is to be sent, number of intended recipients and the ID-tuples of the number of recipients as inputs, and outputs a ciphertext.
4) UserDec: Bob takes system parameters, his private key, and the ciphertext as inputs to recover the plaintext.
5) RecipientsDec: The intended recipient takes system parameters, a private key, a master key, ID-tuple$_i$, and the ciphertext as inputs to recover the plaintext.

Using the ESC scheme, the sender *X* needs to encrypt a file only once, and store the copy of the corresponding ciphertext in a cloud. At that time it should not communicating with none of the recipients, but Bob, as well as all the intended recipients can respectively decrypt the cipher text by using their own private keys. This scheme is collusion resistant, in which only intented recipients can decrypt the file even if all of them collude. But a lower-level user needs to execute the bilinear map operations for almost $\frac{N+1}{2}+3$ times to decrypt a file, where N represents the number of users in the hierarchy. This will lead too much computational cost. At the same time, the main drawback of D. Boneh et al [3] scheme is that, secret keys are generated by the TTP. There is a chance for TTP to become an attacker too. To solve this problem, the trapdoor commitment scheme is proposed. But the trapdoor concept has been proposed by M. Fischlin [8]. Our work is to introduce a trapdoor in the user hierarchy in order to reduce the cost as well as to improve the security than existing scheme. In this scheme, a lower-level user needs to send a trapdoor to a CSP before the user wants to retrieve the files. By using this trapdoor commitment scheme, CSP helps the user to find part of the cipher text without leaking any information about the file. The Trapdoor Commitment scheme will largely reduce the computational cost for decryption.

4. IMPLEMENTATION

4.1 The Trapdoor Commitment Scheme

The trapdoor commitment scheme plays a vital role in cloud storage services. Our work is to construct an ESC scheme with a trapdoor. The Fig.2 shows the trapdoor commitment scheme, which enables CSPs to participate in the partial decipherment without leaking any information about the plaintext between a upper-level user and a lower-level user. Also, in ESC scheme there is a possibility for generating the secret keys by a mistrusted third party instead of Trusted Third Party. With a trapdoor commitment scheme, the upper-level user can securely share the files with the lower-level users. For secure sharing, RSA-based signing and verifying signatures concept largely helps the lower-level users to participate in partial decipherment. By the participation of Cloud Service Provider (CSP) of lower-level users in partial decipherment will lead to low computational cost. The basic definition for trapdoor commitment schemes and their applications were proposed by M. Fischlin [8] which is as follows:

Definition 4.1(Trapdoor commitment scheme)

The Trapdoor Commitment Scheme, which consists of four algorithms,

1) TCgen: The receiver first runs the key generation algorithm TCgen to get a commitment public key and the corresponding trapdoor.
2) TCcom: The trapdoor commitment algorithm TCcom outputs a pair (com, dec) by taking an input value r which is already given and the commitment public key. The output com is the commitment to a value r dec is the related information used to decommit com.
3) 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.
4) TCsim: A simulation algorithm allows the receiver, using the trapdoor, to simulate a new answer (r',dec') for a commitment com when one answer (r,dec) is given.

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

4.2 Construction of the ESC Scheme with Trapdoor Commitment

The Concept of ESC scheme is proposed by Q. Liu et al [ ], where the users can encrypt and decrypt the message in a user hierarchy. But our idea is to construct a trapdoor commitment scheme based on RSA signatures. Initially the sender has to sign the file by using his/her private key to get a signature as $\rho_{S.}$ By using the concept of RSA signature, first splits the private key of the upper-level user into two parts. Then encrypt the file by using first part of the private key and sign it. So the upper-level user S can get his/her partial signature as ρ_1 by signing the original file with respect to his/her partial private key. Now send the partial signature ρ_1 to the CSP and commit the trapdoor by executing TCcom commitment algorithm. There should be a valid answer to de commit the trapdoor by revealing that valid answer to the receiver.

Once getting the valid answer, the receiver now de commits the trapdoor and gets the sender's partial signature $\rho_{1.}$ After that, the receiver has to get another partial signature from the sender for decrypting the original file. For that, the receiver has to send his/her identity proof and that should be verified by the sender. If the verification is successful then the sender can send the second partial signature ρ_2 to the receiver. Now the receiver can get the full signature of the sender by combining the partial signatures. Now the receiver should check whether the second partial signature is valid or not. This can be done by combining the received partial signatures and verify that whether it produce the sender's full signature ρ_S. If so, then the receiver can decrypt the original file from the sender's signature by using his/her public key.

5. DISCUSSION

5.1 Performance

In Q. Liu et al's [17] ESC scheme, the upper-level user can encrypt the file only once and store the copy of the file in the cloud, so that the lower-level users can decrypt the file from the cloud. Based on the number of users in the user hierarchy, the computational cost is too high for decrypting the files from the cloud. Suppose the user hierarchy is too large then computing the cost will get increase for the authenticated lower-level user. But in case of Trapdoor Commitment scheme, there will be a trapdoor between the upper-level users and the lower-level users. Only the intended recipients can decrypt the file directly from the cloud service provider by de committing the trapdoor. In this scheme, the lower-level needs to execute only two times to decrypt the file. That is by allowing CSP to participate in partial decipherment. To achieve this concept, an RSA-based partial signature is needed to participate in decipherment to reduce the computational power.

5.2 Security

Q. Liu et al's [17] 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). 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. In this scheme, the upper-level sends the message by using partial signature. The partial signature can be done by using sender's private key. Instead of knowing the full private key, the TTP knows only the part of sender's private key. So there is no chance for attacks by outsiders. This proposed scheme is fully secure than the ESC scheme.

6. CONCLUSION

Cloud computing is one of the current most important and promising technologies. For more scalable service, ESC scheme is introduced which allows the user to share the secure cloud service services with multiple users who are all in a hierarchy. But the ESC scheme with trapdoor commitment helps the upper-level user to share the file in a secure way with the lower-level users in a hierarchy. The RSA-based partial signature helps the lower level users to decrypt the original file with low computational cost. This increases the security level than the ESC scheme.

7. REFERENCES

[1] M. Bellare and R. Sandhu, The Security of Practical Two-Party RSA Signature Schemes 2001 [Online]. 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, pages 321-334.

[3] D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Proceedings of UROCRYPT 2004, volume 3027 of LNCS, pages 223-38.

[4] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of CRYPTO 2001*, volume 2139 of *LNCS*, pages 213-229.

[5] D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Proceedings of EUROCRYPT 2005,*volume 3494 of *LNCS, pages 440-456.*

[6] M. Chase. Multi-authority attribute based encryption. In Proceedings of TCC 2007, volume 4392 of LNCS, pages 515-534.

[7] M. Chase and S. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *Proceedings of ACM CCS 2009*, pages 121-130.

[8] M. Fischlin, "Trapdoor Commitment Schemes and Their Applications," PhD. Dissertation, Fachbereich Mathematik, Johann Wolfgang Goethe-Universität Frankfurt am Main, Frankfurt, Germany, 2001.

[9] R. Gennaro, T. Rabin, and H. Krawczyk, "RSA-based undeniable signature," *J. Cryptology*, vol. 13, no. 4, pp. 397–416, 2000.

[10] R Gennaro. (2008), "Robust and Efficient Sharing of RSA Functions",Journal of Cryptology, Vol 13, No 2, pp 273-300.

[11] R. Gennaro, T. Rabin, and H. Krawczyk, "RSA-based undeniable signature," *J. Cryptology*, vol. 13, no. 4, pp. 397–416, 2000.

[12] C. Gentry and S. Halevi. Hierarchical identity base encryption with polynomially many levels. In *Proceedings of TCC 2009, volume 5444* of *LNCS, pages.437-456.*

[13] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In *Proceedings of ASIACRYPT 2002*, volume 2501 of *LNCS,* pages 548-566.

[14] 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.

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

[16] Q. Liu, G. Wang and J. Wu. An efficient privacy preserving keyword search scheme in cloud computing. In *Proceedings of IEEE CSE 2009/TrustCom 2009*, pages 715-720.

[17] Q. Liu; G. Wang; J. Wu. An efficient sharing of secure cloud storage services. Computer and Information Technology (CIT), 2010 IEEE 10th International Conference onDigital Object Identifier: 10.1109/CIT.2010.171 Publication Year: 2010 , Page(s): 922 – 929.

[18] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proc. CRYPTO'91*, 1991, vol. 576, LNCS, pp. 129–140, Springer-Verlag.

[19] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[20] J. Rompel, "One-way functions are necessary and sufficient for secure signatures," in *Proc. STOC'90*, 1990, pp. 387–394, ACM.

[21] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Proceedings of EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457-473.

[22] G. Wang, "An abuse-free fair contract signing protocol based on the RSA signature," in Proc. 14th Int. Conf. World Wide Web (WWW'05), 2005, pp. 412–421, ACM Press.

[23] G. Wang, J. Baek, D. S. Wong, and F. Bao, "On the generic and efficient constructions of secure designated confirmer signatures," in *Proc.PKC'07*, 2007, vol. 4450, LNCS, pp. 43–60, Spriger-Verlag.

[24] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *Proceedings of CRYPTO 2009*, volume 5677 of *LNCS*, pages 619-636.

[25] S. Yu, C. Wang, K. Ren, and W. Lou, Achieving secure, scalable, and fine-grained data access control in cloud computing. In *Proceedings of IEEE INFOCOM 2010*, pages 15-19.

[26] 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*, pages 19-25.