A Novel Identity based Secure Distributed Data Storage System in Cloud Computing based on Database –as-a-Service

kilani kusuma [#1], P.S.Avadhani [*2]

Department of Computer Science & System Engineering,
Andhra University College of Engineering,
Visakhapatnam, AP (INDIA)

Abstract

Now a day's by using secure distributed data storage schema, we can reduce a lot of burden of maintaining a large number of files from the data owner to proxy servers. In general proxy servers are those which can convert encrypted files for the data owner to encrypted files for the data receiver without knowing the original information. For space complexity the data owner will remove the original files from his system. As data was stored on a remote server, we must mainly concentrate on two major issues like confidentiality and integrity of the outsourced data. In this paper, we have proposed two new identity-based secure distributed data storage (IBSDDS) schemes. Our two new schemes can capture the following properties: (1) Firstly whenever the data/file which is uploaded by file owner on remote server he will decide the file access permission independently on his own without the help of any third party private key generator (PKG).(2) For one query, a receiver can only access appropriate one file, instead of all files that are stored by the owner; (3) Our two new schemes are secure against the collusion attacks, namely even if the receiver can compromise the proxy servers, he cannot obtain the owner's secret key. To the best of our knowledge, it is the first IBSDDS schemes where access permissions is made by the owner for an exact file and collusion attacks can be protected in the standard model.

Keywords

Access Control Policy, Data Security, Identity-Based Data Storage Systems,

1. Introduction

Cloud computing is a new technology which emerges with a lot of facilities that are really beneficial for the end users to manage their personal files with the convenient notion called database-as-a-service (DAS) **[1], [2], [3].** By using this new DAS schemes, any data user can outsource his/her encrypted data files to untrusted proxy servers. Proxy servers can also perform some functions on the outsourced ciphertexts data which was stored by the data owners without knowing anything about the original files. Unfortunately, this technique hasn't been in practice extensively. The main reason which is behind this cause is users are generally concentrating on the main factors like confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party user. As the data owner stores his valuable data on a third party proxy server, he/she will remove the original content in their system in order to reduce the space wastage. Therefore, how to guarantee the outsoured files which can't be accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Furthermore, we also need to concentrate how efficiently the data user can access his/her files which were stored by them in proxy server. Consequently, a lot of research around these topics grows significantly.

In cloud computing domain, confidentiality is proposed mainly to prevent unauthorized users from accessing the most sensitive data as it is subject to unauthorized disclose and access after being outsourced. Since the introduction of DAS, the confidentiality of outsourced data has been the primary focus among the research community. To provide confidentiality to the outsourced data, encryption schemes are deployed **[4], [5], [6], [7], [8].**

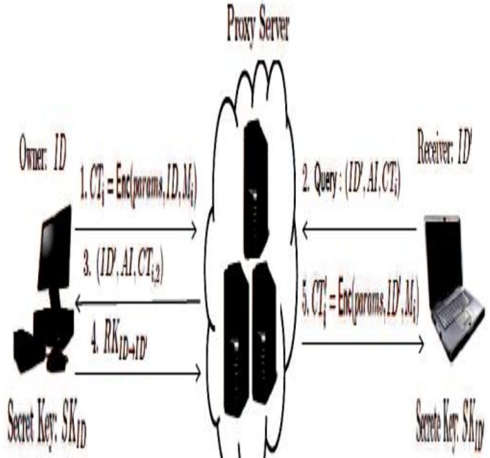


Fig. 1: The Model of Identity-Based Secure Distributed Data Storage Scheme

In cloud computing domain, integrity can prevent outsourced data from being re- placed and modified. Some schemes have been proposed to protect the integrity of the outsourced data, such as proof of retrievability [9], [10], [11], [12], [13] and provable data possession [14], [15], [16]. In these schemes, digital signature schemes and message authentication codes (MAC) are deployed. Query in data storage is executed between a receiver and a proxy server. The proxy server can perform some functions on the outsourced ciphertexts and convert them to those for the receiver which is shown clearly in figure 1. As a result, the receiver can obtain the data outsourced by the owner without the proxy server knowing the content of the data [17], [18], [19], [20].

2. Related Work and Assumptions

In this section, we review schemes related to identity- based secure distributed data storage (IBSDDS) schemes.

2.1 Data Storage Systems

Data storage systems are the new systems which enable users to store their data to external proxy servers to enhance the access and availability, and reduce the maintenance cost. Author Samaritan and Author Vimercati **[21]** addressed the privacy issues in data outsourcing expanding from the data confidentiality to data utility, and pointed out the main research directions in the protection of the externally stored data.

2.2 Networked File Systems

In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions. The interactions between the proxy servers and receivers are executed in a secure channel. Therefore, these systems cannot provide an end-to-end data security, namely they cannot ensure the confidentiality of the data stored at the proxy server [22]. In these schemes, a receiver authenticates himself to the proxy server using his password. Then, the proxy server passes the authentication result to the file owner. The owner will make an access permission according to the received information.

2.3 Identity-based Proxy Re-encryption Schema (IBPRES)

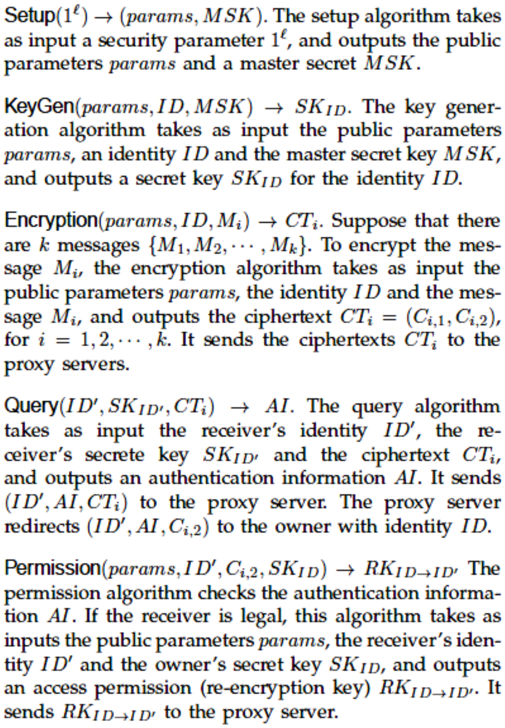
Proxy cryptosystem was introduced by Mambo and Okamoto to delegate the decryption power to a designated decryptor. Then, Blaze, Bleumer, and Strauss proposed an atomic proxy cryptosystem where a semi-trusted proxy server can transfer a ciphertext for the original decryptor to a ciphertext for the designated decryptor without knowing the plaintext.

Proxy cryptosystem as an efficient primitive has been used in email forwarding, law enforcement and data storage. Identity- based cryptosystem introduced by Shamir is a sys- tem where the public key can be any arbitrary string and the secret key is issued by a trusted party called the private key generator (PKG). Being different from public key infrastructure (PKI), two parties can communicate directly without verifying their public key certificates in identity-based systems. The first secure and practical identity-base encryption (IBE) was proposed by Boneh and Franklin based on pairing.

3. Identity-based Secure Distributed Data Storage (IBSDDS)

There are mainly four entities in identity-based secure distributed data storage (IBSDDS) scheme: the private key generator (PKG), the data owner, the proxy server and the receiver. The PKG validates the users' identities and issues secret keys to them. The data owner encrypts his data and outsources the ciphertexts to the proxy servers.

Proxy servers store the encrypted data and transfer the ciphertext for the owner to the ciphertext for the receiver when they obtain access permission (re-encryption key) from the owner. The receiver authenticates himself to the owner and decrypts the re-encrypted ciphertext to obtain the data. An IBSDDS scheme consists of the following algorithms:



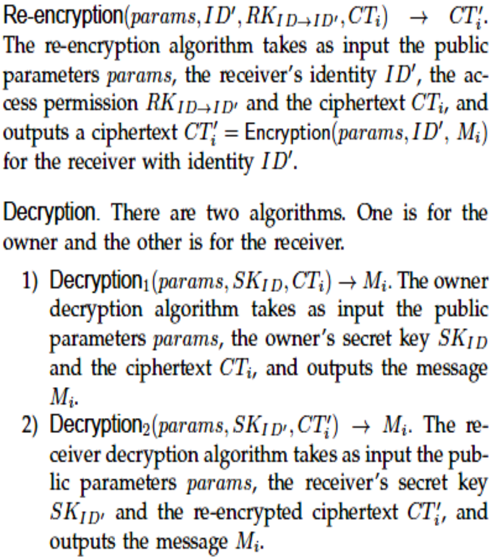
Setup$(1^\ell) \rightarrow (params, MSK)$. The setup algorithm takes as input a security parameter 1^ℓ, and outputs the public parameters $params$ and a master secret MSK.

KeyGen$(params, ID, MSK) \rightarrow SK_{ID}$. The key generation algorithm takes as input the public parameters $params$, an identity ID and the master secret key MSK, and outputs a secret key SK_{ID} for the identity ID.

Encryption$(params, ID, M_i) \rightarrow CT_i$. Suppose that there are k messages $\{M_1, M_2, \cdots, M_k\}$. To encrypt the message M_i, the encryption algorithm takes as input the public parameters $params$, the identity ID and the message M_i, and outputs the ciphertext $CT_i = (C_{i,1}, C_{i,2})$, for $i = 1, 2, \cdots, k$. It sends the ciphertexts CT_i to the proxy servers.

Query$(ID', SK_{ID'}, CT_i) \rightarrow AI$. The query algorithm takes as input the receiver's identity ID', the receiver's secrete key $SK_{ID'}$ and the ciphertext CT_i, and outputs an authentication information AI. It sends (ID', AI, CT_i) to the proxy server. The proxy server redirects $(ID', AI, C_{i,2})$ to the owner with identity ID.

Permission$(params, ID', C_{i,2}, SK_{ID}) \rightarrow RK_{ID \rightarrow ID'}$ The permission algorithm checks the authentication information AI. If the receiver is legal, this algorithm takes as inputs the public parameters $params$, the receiver's identity ID' and the owner's secret key SK_{ID}, and outputs an access permission (re-encryption key) $RK_{ID \rightarrow ID'}$. It sends $RK_{ID \rightarrow ID'}$ to the proxy server.



Re-encryption$(params, ID', RK_{ID \rightarrow ID'}, CT_i) \rightarrow CT_i'$. The re-encryption algorithm takes as input the public parameters $params$, the receiver's identity ID', the access permission $RK_{ID \rightarrow ID'}$ and the ciphertext CT_i, and outputs a ciphertext $CT_i' =$ Encryption$(params, ID', M_i)$ for the receiver with identity ID'.

Decryption. There are two algorithms. One is for the owner and the other is for the receiver.

1) Decryption$_1(params, SK_{ID}, CT_i) \rightarrow M_i$. The owner decryption algorithm takes as input the public parameters $params$, the owner's secret key SK_{ID} and the ciphertext CT_i, and outputs the message M_i.
2) Decryption$_2(params, SK_{ID'}, CT_i') \rightarrow M_i$. The receiver decryption algorithm takes as input the public parameters $params$, the receiver's secret key $SK_{ID'}$ and the re-encrypted ciphertext CT_i', and outputs the message M_i.

4. Implementation Modules

Implementation is a stage where the theoretical design is automatically converted into practical form. This is mainly divided into following modules.

1. Data Owner Module
2. Private key Generator Module
3. Proxy Server Module
4. The Receiver Module

1. Data Owner Module

In this module, first the new data owner registers and then gets a valid login credentials. After logged in, the data owner has the permission to upload their file into the Cloud Server. The data owner encrypts his data and outsources the ciphertexts to the proxy servers.

2. Private Key Generator Module

In this module, the private key generator (PKG) validates the users' identities and issues secret keys to them. The key is generated and sent to their respective mail id's with the file name and the corresponding key values.

3. Proxy Server Module

Proxy servers store the encrypted data and transfer the cipher text for the owner to the cipher text for the receiver when they obtain access permission (re-encryption key) from the owner. In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions. The interactions between the proxy servers and receivers are executed in a secure channel. Therefore, these systems cannot provide an end-to-end data security, namely they cannot ensure the confidentiality of the data stored at the proxy server. In these schemes, a receiver authenticates himself to the proxy server using his password. Then, the proxy server passes the authentication result to the file owner. The owner will make access permission according to the received information.

4. Receiver Module

The receiver authenticates himself to the owner and decrypts the re-encrypted Ciphertext to obtain the data. In these systems, an end to-end security is provided by cryptographic protocols which are executed by the file owner to prevent proxy servers and unauthorized users from modifying and accessing the sensitive files. These systems can be divided into two types: shared file system and non-shared system. In shared file systems the owner can share his files with a group of users. Cryptographic techniques deployed in these systems are key sharing, key agreement and key revocation. In non-shared file systems in order to share a file with another user, the owner can compute an access key for the user using his secret key. In these two systems, the integrity of the sensitive files is provided by digital signature schemes and message authentication codes (MAC).

5. Enhanced System Architecture

The following diagram represents the enhanced system architecture /flow of IBSDDS model which is clearly shown in figure.2. In the proposed paper, we have extended the same IBSDDS schema with more enhanced security by including a mailing concept like each and every user whenever registers in his/her account ,they should provide a valid email id, where the data owner uploads any data can select any of the user from the set of user list .The data owner will now try to give key for accessing the uploaded file for the selected user, the key is sent to the selected user registered mail id and the end user should verify his valid mail id and with that key only he can decrypt the data if not he cannot access his data files.

6. Conclusion

Now a day's by using secure distributed data storage schema, we can reduce a lot of burden of maintaining a large number of files from the data owner to proxy servers. Identity-based secure distributed data storage (IBSDDS) schemes are a special kind of distributed data storage schemes where users are identified by their identities and can communicate without the need of verifying the public key certificates. In this paper, we proposed two new IBSDDS schemes in standard model where, for one query, the receiver can only access one file, instead of all files. Furthermore, the access permission can be made by the owner, instead of the trusted party. Notably, our schemes are secure against the collusion attacks. The first scheme is CPA secure, while the second one is CCA secure.

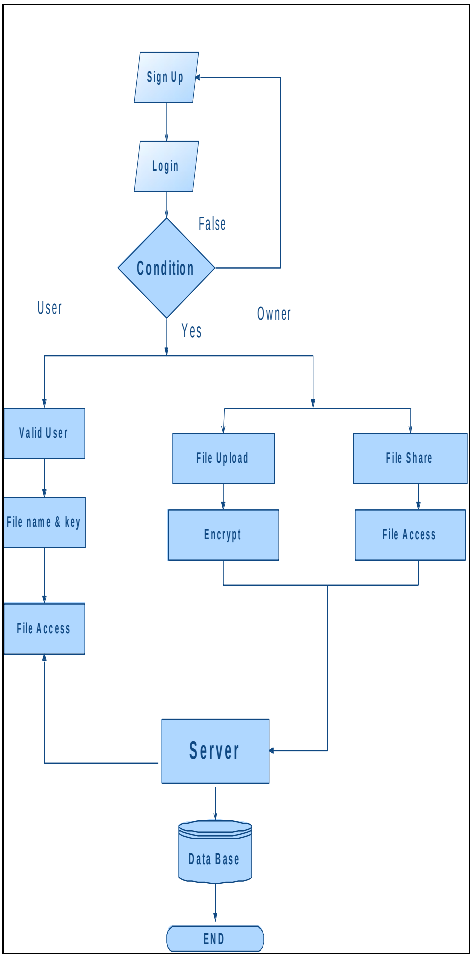


Fig. 2. Enhanced IBSDDS Architecture

7. References

[1] H. Hacig¨um¨us, B. R. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in *Proceedings: SIGMOD Conference - SIGMOD'02* (M. J. Franklin, B. Moon, and A. Ailamaki, eds.), vol. 2002, (Madison, Wisconsin, USA), pp. 216–227, ACM, Jun. 2002.

[2] L. Bouganim and P. Pucheral, "Chip-secured data access: Confidential data on untrusted servers," in *Proc. International Conference on Very Large Data Bases - VLDB'02*, (Hong Kong, China), pp. 131–142, Morgan Kaufmann, Aug. 2002.

[3] U. Maheshwari, R. Vingralek, and W. Shapiro, "How to build a trusted database system on untrusted storage," in *Proc. Symposium on Operating System Design and Implementation - OSDI'00*, (San Diego, California, USA), pp. 135–150, USENIX, Oct. 2000.

[4] A. Ivan and Y. Dodis, "Proxy cryptography revisited," in *Proc.Network and Distributed System Security Symposium - NDSS'03*, (San Diego, California, USA), pp. 1–20, The Internet Society, Feb. 2003.

[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in *Proc. Network and Distributed System Security Symposium - NDSS'05*, (San Diego, California, USA), pp. 1–15, The Internet Society, Feb. 2005.

[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[7] S. D. C. di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P. Samarati, "Effficient and private access to outsourced data," in *Proc. International Conference on Distributed Computing Systems - ICDCS'11*, (Minneapolis, Minnesota, USA), pp. 710–719, IEEE, Jun. 2011.

[8] H.-Y. Lin and W.-G. Tzeng, "A secure erasure code-based cloud storage system with secure data forwarding," *IEEE Transactions on Parallel and Distributed Systems*, Digital Object Indentifier 10.1109/TPDS.2011.252 2012.

[9] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. Advances in Cryptology - ASIACRYPT'08* (J. Pieprzyk, ed.), vol. 5350 of *Lecture Notes in Computer Science*, (Melbourne, Australia), pp. 90–107, Springer, Dec. 2008.

[10] A. Juels and B. S. K. Jr., "PORs: Proofs of retrievability for large files," in *Proceedings: ACM Conference on Computer and Communications Security - CCS'07* (P. Ning, S. D. C. di Vimercati, and P. F.Syverson, eds.), (Alexandria, Virginia, USA), pp. 584–597, ACM,Oct. 2007.

[11] Y. Dodis1, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in *Proc. Theory of Cryptography Conference - TCC'09* (O. Reingold, ed.), vol. 5444 of *Lecture Notes in Computer Science*, (San Francisco, CA, USA), pp. 109–127, Springer, Mar.2009.

[12] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in *Proc. ACM Cloud Computing Security Workshop - CCSW'09*, (Chicago, Illinois, USA), pp. 43–53, ACM, Nov. 13 2009.

[13] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in *Proc. Advances in Cryptology - ASIACRYPT'09* (M. Matsui, ed.), vol. 5912 of *Lecture Notes in Computer Science*, (Tokyo, Japan), pp. 319–333, Springer, Dec. 2009.

[14] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson,and D. Song, "Provable data possession at untrusted stores,"in *Proc. ACM Conference on Computer and Communications Security- CCS'07* (P. Ning, S. D. C. di Vimercati, and P. F. Syverson, eds.),(Alexandria, Virginia, USA), pp. 598–610, ACM, Oct. 2007.

[15] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. International conference on Security and privacy in communication netowrks - SecureComm'08*, (Istanbul, Turkey), Sep., ACM, 2008.

[16] C. C. Erway, A. K¨upc¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proc. ACM Conference on Computer and Communications Security - CCS'09* (E. Al-Shaer, S. Jha, and A. D. Keromytis, eds.), (Chicago, Illinois, USA), pp. 213–222, ACM, Nov. 2009.

[17] B. Carbunar and R. Sion, "Toward private joins on outsourced data," *IEEE Transactions on Knowlege and Data Engineering*, vol. 9, no. 24, pp. 1699–1710, 2012.

[18] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Transactions on Knowlege and Data Engineering*, p. Digital Object Indentifier 10.1109/TKDE.2011.78.

[19] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[20] M. L. Yiu, G. Ghinita, C. S. Jensen, and P. Kalnis, "Enabling search services on outsourced private spatial data," *The VLDB Journal*, vol. 19, no. 3, pp. 363–384, 2010.

[21] P. Samarati and S. D. C. di Vimercati, "Data protection in outsourcing scenarios: Issues and directions," in *Proc. ACM Symposium on Information, Computer and Communications Security - ASIACCS'10* (D. Feng, D. A. Basin, and P. Liu, eds.), (Beijing, China), pp. 1–14, ACM, Apr. 2010.

[22] V. Kher and Y. Kim, "Securing distributed storage: Challenges, techniques, and systems," in *Proc. ACM Workshop On Storage Security And Survivability - StorageSS'05* (V. Atluri, P. Samarati, W. Yurcik, L. Brumbaugh, and Y. Zhou, eds.), (Fairfax, VA, USA), pp. 9–25, ACM, Nov. 2005.

8. About the Authors

kilani kusuma is currently pursuing her 2 Years M.Tech (CSSE) in Computer Science and System Engineering at Andhra University College of Engineering, Visakhapatnam. Her area of interests includes Cryptography and Network Security.

P.S.Avadhani is currently working as Professor, in Computer Science and System Engineering at Andhra University College of Engineering, Visakhapatnam. His research interests include Cryptography, Network Security and Computer Graphics