Key Aggregate Cryptosystem for Secure Group Data Sharing on Cloud

Smital Erande[1]

[1]Savitribai Phule Pune University, VACOE, Ahmednagar, India
smital.dhavane[at]gmail.com
[2]Savitribai Phule Pune University, VACOE, Ahmednagar, India

Abstract: ***Search over scrambled information is a basically critical empowering system in distributed computing, where encryption-before outsourcing is a principal answer for securing client information protection in the untrusted cloud server environment. Many secure hunt plans have been concentrating on the single-donor situation, where the outsourced dataset or the protected searchable file of the dataset are encoded and overseen by a solitary proprietor, regularly in view of symmetric cryptography. In this paper, we concentrate on an alternate yet additionally difficult situation where the outsourced dataset can be contributed from different proprietors and are searchable by numerous clients, i.e. multi-client multi-supporter case. Propelled by trait based encryption (ABE), we show the main characteristic based watchword look conspire with proficient client disavowal (ABKS-UR) that empowers versatile fine-grained (i.e. document level) look approval. Our plan permits different proprietors to encode and outsource their information to the cloud server freely. Clients can create their own particular pursuit abilities without depending on a constantly online trusted power. Fine-grained seek approval is additionally actualized by the proprietor implemented get to strategy on the list of every record. Promote, by fusing intermediary re-encryption and lethargic re-encryption procedures, we can appoint overwhelming framework redesign workload amid client disavowal to the creative semi-trusted cloud server. We formalize the security definition and demonstrate the proposed ABKS-UR plot specifically secure against picked catchphrase assault. To assemble certainty of information client in the proposed secure inquiry framework, we additionally outline a query item check conspire. At long last, execution assessment demonstrates that the productivity of our plan. In ABKS-UR, the get to approach is connected to the figure message in plaintext shape, which may likewise release some private data about end-clients. Existing techniques just halfway shroud the characteristic values in the get to approaches, while the trait names are still unprotected, these issues are change in our plan to give more security. While transferring a record time server is connected with document to give access to record to restricted time simply after that time document is inaccessible for shoppers additionally property blossom channel create characteristics of record while transferring and this traits are store with document. Quality power in our plan relegate open key to client while transferring documents on cloud furthermore records mystery key and private key to information customer while transferring. Subsequent to entering watchword client buyer will get best rank outcome relies on characteristic and time and can download that document if customer having key of that record and can decode record.***

Keywords: Searchable encryption, data sharing, cloud storage, data privacy

1. Introduction

Cloud storage has emerged as a promising solution for providing ubiquitous, convenient, and on-demand accesses to large amounts of data shared over the Internet. Today, millions of users are sharing personal data, such as photos and videos, with their friends through social network applications based on cloud storage on a daily basis.

However, while enjoying the convenience of sharing data via cloud storage, users are also increasingly concerned about inadvertent data leaks in the cloud. These such data leaks, caused by a malicious adversary or a misbehaving cloud operator, can usually lead to serious breaches of personal privacy or business secrets (e.g., the recent high profile incident of celebrity photos being leaked in iCloud). To address users concerns over potential data leaks in cloud storage, a common approach is for the data owner to encrypt all the data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted by those who have the decryption keys. Such cloud storage is often called the cryptographic cloud storage [6]. The data encryption makes it challenging for users to search and then selectively retrieve only the data containing given keywords. To utilize a searchable encryption scheme in which the data owner is required to encrypt potential keywords and upload them to the cloud together with encrypted data, like as, for retrieving data matching a keyword, the user will send the corresponding keyword trapdoor to the cloud for performing search over the encrypted data. Combination of searchable encryption scheme with cryptographic cloud storage can achieve the basic security requirements of a cloud storage, implementing such a system for large scale applications involving millions of users and billions of files may still be hindered by practical issues involving the efficient management of encryption keys, which, to the best of our knowledge, are largely ignored in the literature. Firstly, basic need for selectively sharing encrypted data with different users (e.g., sharing a photo with certain friends in a social network application, or sharing a business document with some colleagues on a cloud drive) usually demands different encryption keys to be used for different files, via secure channels large keys must not only be distributed to users, but also be securely stored and managed by the users in their devices. In addition, a large number of trapdoors must be generated by users and submitted to the cloud in order to perform a keyword search over many files.

Here, this challenge we address by proposing the novel concept of key-aggregate searchable encryption (KASE), and instantiating the concept through a concrete KASE scheme. This approach can apply any cloud storage that supports the searchable group data sharing functionality, which means any user may selectively share a group of selected files with a group of selected users, while allowing the latter to perform keyword search over the former. For supporting searchable group data sharing the important requirements for efficient key management are twofold. A data owner needs to distribute a single aggregate key (instead of a group of keys) to a user for sharing any number of files and the second one, user needs to submit a single aggregate trapdoor (instead of a group of trapdoors) to the cloud for performing keyword search over any number of shared files. This concept is the first known scheme that can satisfy both requirements (the key-aggregate cryptosystem [4], which has inspired our work, can satisfy the first requirement but not the second).

2. Literature Survey

SPICE: Simple Privacy-Preserving Identity-Management for Cloud Environment

Identity security and privacy have been regarded as one of the top seven cloud security threats. There are a few identity management solutions proposed recently trying to tackle these problems. However, none of these can satisfy all desirable properties. In particular, unlink ability ensures that none of the cloud service providers (CSPs), even if they collude, can link the transactions of the same user. On the other hand, delegable authentication is unique to the cloud platform, in which several CSPs may join together to provide a packaged service, with one of them being the source provider which interacts with the clients and performs authentication while the others will be transparent to the clients. Note that CSPs may have different authentication mechanisms that rely on different attributes. Moreover, each CSP is limited to see only the attributes that it concerns [1]. The new thing of the our scheme come from combining and exploiting two group signatures so that we can randomize the signature to make the same signature look different for multiple uses of it and hide some parts of the messages which are not the concerns of the CSP. Our scheme is quite applicable to cloud systems due to its simplicity and efficiency.

Privacy-Preserving Public Auditing for Secure Cloud Storage

Remotely store their own data by users and enjoy the on-demand high-quality services and applications from a shared pool of configurable computing resources, without the burden of local data storage and maintenance using cloud storage. Additional, users should be use the cloud storage as if it is local, without worrying about the need to verify its integrity. For cloud storage, enabling public audit ability is of critical importance so that users can resort to a third-party auditor (TPA) to check the integrity of outsourced data and be worry free. Safely presents an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. Secure cloud storage system supporting privacy-preserving public auditing is proposed in this paper. Further enhance our result to enable the TPA to perform audits for multiple users efficiently and simultaneously. Performance and extensive security analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design [2].

Storing Shared Data on the Cloud via Security-Mediator

Recently, lots of institutes outsource data storage to the cloud such that a member (owner) of an organization can easily share data with other members (users). Just due to presence of security concerns in the cloud, both owners and users are suggested to verify the integrity of cloud data with Provable Data Possession (PDP) before further utilization on data. However, previous methods either unnecessarily reveal the identity of a data owner to the untrusted cloud or any public verifiers, or introduce significant overheads on verification metadata to preserve anonymity. In this paper, we propose a simple and efficient publicly verifiable approach to ensure cloud data integrity without sacrificing the anonymity of data owners nor requiring significant verification metadata. Our purpose, decouples the anonymity protection mechanism from the PDP. So that institutes can employ its own anonymous authentication mechanism, and the cloud is oblivious to that since it only deals with typical PDP-metadata, consequently, there is no extra storage overhead when compared with existing non-anonymous PDP solutions. Security analyses prove our scheme is secure, and experiment results demonstrate our scheme is efficient [4].

Dynamic Secure Cloud Storage with Provenance

One concern in using cloud storage is that the sensitive data should be confidential to the servers which are outside the trust domain of data owners. Another issue is that the user may want to preserve his/her anonymity in the sharing or accessing of the data (such as in Web 2.0 applications). Getting enjoy the benefits of cloud storage, we need a confidential data sharing mechanism which is fine-grained (one can specify who can access which classes of his/her encrypted files), dynamic (the total number of users is not fixed in the setup, and any new user can decrypt previously encrypted messages), scalable (space requirement does not depend on the number of decryptors), accountable (anonymity can be revoked if necessary) and secure (trust level is minimized). This paper addresses the problem of building a secure cloud storage system which supports dynamic users and data provenance. Previous system is based on specific constructions and does not offer all of the aforementioned desirable properties. Most importantly, dynamic user is not supported. We study the various features offered by cryptographic anonymous authentication and encryption mechanisms; and instantiate our design with verifier-local revocable group signature and identity-based broadcast encryption with constant size ciphertexts and private keys. To realize our concept, we equip the broadcast encryption with the dynamic ciphertext update feature, and give formal security guarantee against adaptive chosen-ciphertext decryption and update attacks [5].

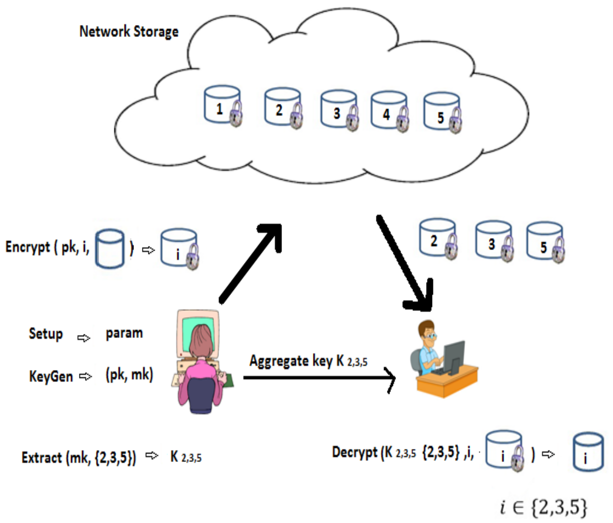
3. Proposed System

- Key assignment schemes aim to minimize the expense in storing and managing secret keys for general cryptographic use.

- Utilizing a tree structure, a key for a given branch can be used to derive the keys of its descendant nodes (but not the other way round).

- Just granting the parent key implicitly grants all he keys of its descendant nodes. Sandhu proposed a method to generate a tree hierarchy of symmetric keys by using repeated evaluations of pseudorandom function/block-cipher on a fixed secret.

- The concept can be generalized from a tree to a graph. More advanced cryptographic key assignment schemes support access policy that can be modelled by an acyclic graph or a cyclic graph.



- Most of these schemes produce keys for symmetric-key cryptosystems, even though the key derivations may require modular arithmetic as used in public-key cryptosystems, which are generally more expensive than "symmetric-key operations" such as pseudorandom function.

4. Mathematical Model

Set S = I, P, R, O

Where I = set of all inputs given to the system. (User name, password, encryption key)

P= Set of process to generate the output.
R= Set of rules.
O= Set of Output.

I= α, β, γ, δ
α = login (id,password)

- Enter id and password

- Validate with database: Select * from user where userid='id' and password='password'

- if (userid==id and password==password) then
- login successful

- else login unsuccessful

β =login result

- if (userid==id and password==password) then

- login successful

- else login unsuccessful

γ = pk and mk
δ = Pf
Where pk=public key, mk= master key and Pf = file to be encrypted.
P= p0, p1, p2, p3
P0= Login to system
P1= Encrypt (Pf)

- Divide x into two 32-bit halves: xL, xR

- For i = 1to 16:

- x L = XL XOR Pi

- xR = F(XL) XOR xR

- Swap XL and xR

- Swap XL and xR (Undo the last swap.)

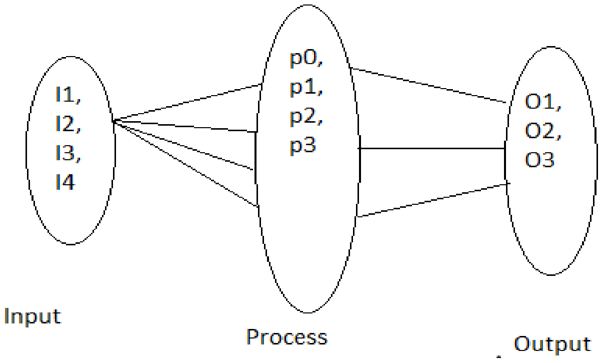
- xR = xR XOR P17

- xL = xL XOR P18

- Recombine xL and xR

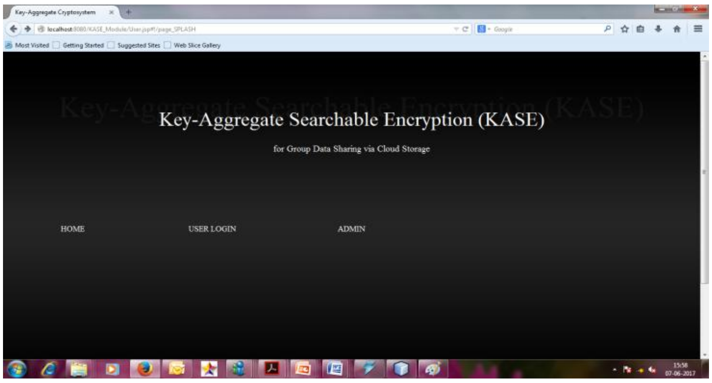
P2= Generate aggregate key
P3=Decrypt (Ef)
R= R0, R1
R0= Verify (id,password).
R1= Activation Status.
O=O1, O2, O3
O1= Ef encrypted _le
O2= Ak aggregate key
pair (pk,msk)
where pk-public key and
msk-master key
O3= Df Decrypted file

```
Constants: int Nb = 4;

change someday
int Nr = 10, 12, or 14; rounds,

Inputs:
 array in of 4*Nb bytes
 array out of 4*Nb bytes
 array w of 4*Nb*(Nr+1) bytes

Internal work array:
state, 2-dim array of 4*Nb bytes, 4 rows and Nb cols

Algorithm: void InvCipher(byte[] in, byte[] out, byte[] w)
byte[][] state = new byte[4][Nb];
state = in;
AddRoundKey(state, w, Nr*Nb, (Nr+1)*Nb - 1);
for (int round = Nr-1; round >= 1; round-)
InvShiftRows(state);
InvSubBytes(state);
AddRoundKey(state, w, round*Nb, (round+1)*Nb-1);
MixColumns(state);
InvShiftRows(state);
InvSubBytes(state);
AddRoundKey(state, w, 0, Nb - 1);
out = state;
```



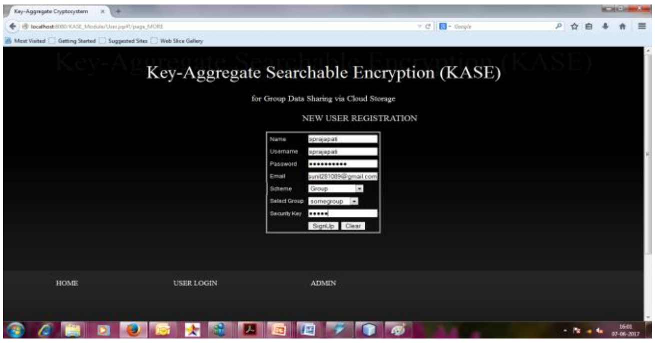
5. Results

The proposed system result is as shown in the given below:


Key-Aggregate Searchable Encryption (KASE)

for Group Data Sharing via Cloud Storage

Figure: Home Page

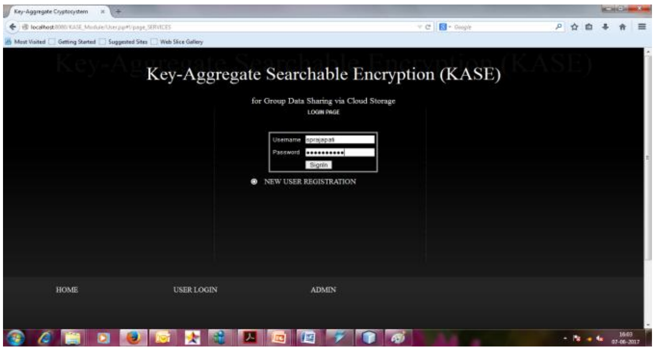

Key-Aggregate Searchable Encryption (KASE)

for Group Data Sharing via Cloud Storage

NEW USER REGISTRATION

Name: sprajapati
Username: sprajapati
Password: ●●●●●●●●●●●
Email: sunil281089@gmail.com
Scheme: Group
Select Group: somegroup
Security Key: ●●●●●

Figure: User Registration

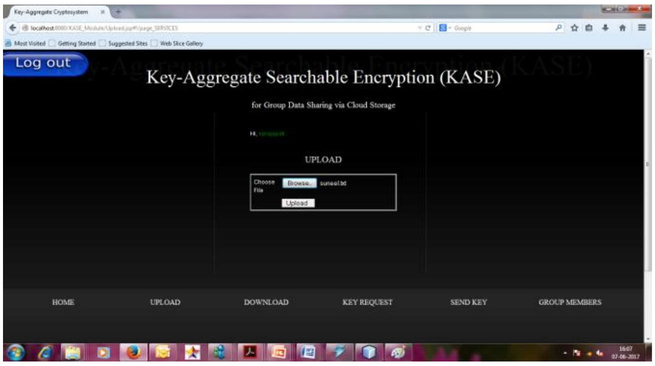

Key-Aggregate Searchable Encryption (KASE)

for Group Data Sharing via Cloud Storage

LOGIN PAGE

Username: sprajapati
Password: ●●●●●●●●●●●

Figure: User Login

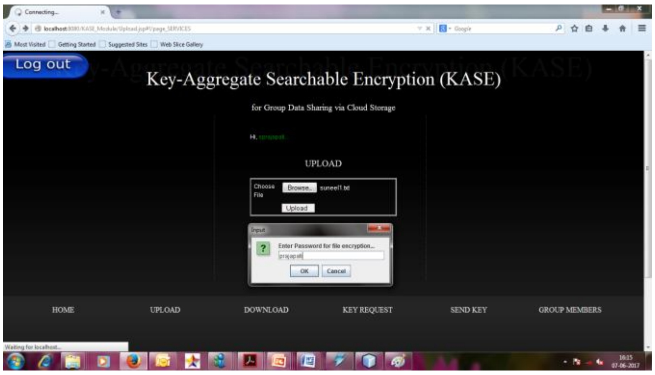

Key-Aggregate Searchable Encryption (KASE)

for Group Data Sharing via Cloud Storage

UPLOAD

Choose File: suneel.txt

Figure: File Uploading


Key-Aggregate Searchable Encryption (KASE)

for Group Data Sharing via Cloud Storage

UPLOAD

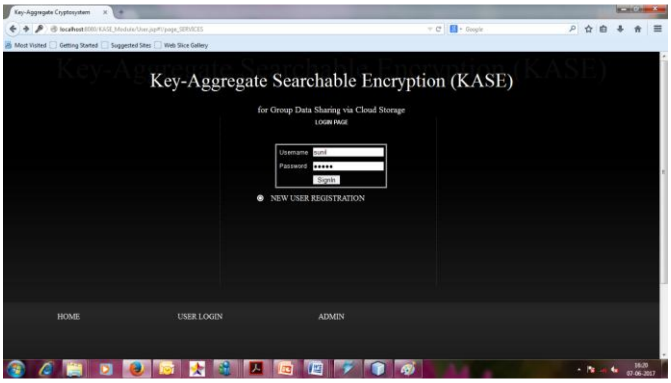
Choose File: suneel1.txt

Input

Enter Password for file encryption...

prajapati

Figure: Enter file encryption key while uploading the file


Key-Aggregate Cryptosystem
localhost:8080/KASE_Module/User.jsp#!/page_SERVICES
Google
Most Visited
Getting Started
Suggested Sites
Web Slice Gallery

Key-Aggregate Searchable Encryption (KASE)

for Group Data Sharing via Cloud Storage

LOGIN PAGE

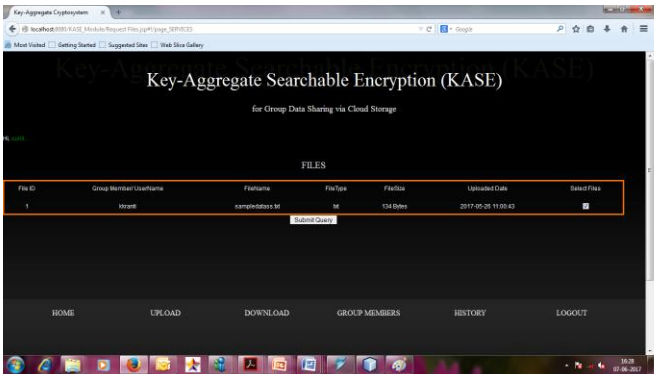
Username: sunil
Password: •••••

Signin

● NEW USER REGISTRATION

HOME USER LOGIN ADMIN

Figure: New user login for key request


Key-Aggregate Cryptosystem
localhost:8080/KASE_Module/Request Files.jsp#!/page_SERVICES
Google
Most Visited
Getting Started
Suggested Sites
Web Slice Gallery

Key-Aggregate Searchable Encryption (KASE)

for Group Data Sharing via Cloud Storage

Hi, sunil..

FILES

File ID	Group Member/ UserName	FileName	FileType	FileSize	Uploaded Date	Select Files
1	kkranti	sampledatass.txt	txt	134 Bytes	2017-05-26 11:00:43	☑

Submit Query

HOME UPLOAD DOWNLOAD GROUP MEMBERS HISTORY LOGOUT

Figure: Select one or more file for key request

6. Conclusion

In this system, we design the first verifiable attribute based keyword search scheme in the cloud environment, which enables scalable and _ne-grained owner-enforced encrypted data search supporting multiple data owners and data users. Compared with existing public key authorized keyword search scheme, our scheme could achieve system scalability and fine-grainedness at the same time. Different from search scheme with predicate encryption, our scheme enables a flexible authorized keyword search over arbitrarily structured data. In addition, by using proxy re-encryption and lazy re-encryption techniques, the proposed scheme is better suited to the cloud outsourcing model and enjoys efficient user revocation. On the other hand, we make the whole search process verifiable and data user can be assured of the authenticity of the returned search result. We also formally prove the proposed scheme semantically secure in the selective model. In a proposed scheme, the owner only needs to distribute a single key to a user when sharing lots of documents with the user and the user only needs to submit a single trapdoor when he queries over all documents shared by the same owner. However, if a user wants to query over documents shared by multiple owners, he must generate multiple trapdoors to the cloud. How to reduce the number of trapdoors under multi-owners setting is a future work. Moreover, federated clouds have attracted a lot of attention nowadays, but our KASE cannot be applied in this case directly. It is also a future work to provide the solution for KASE in the case of federated clouds.

References

[1] S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M. Yiu, "SPICE - Simple Privacy- Preserving Identity-Management for Cloud Environment," in Applied Cryptography and Network Security - ACNS 2012, ser. LNCS, vol. 7341. Springer, 2012, pp. 526-543.

[2] L. Hardesty, "Secure computers aren't so secure," MIT press, 2009, http://www.physorg.com/news176107396.html.

[3] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, 2013.

[4] B. Wang, S. S. M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013.

[5] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, "Dynamic Secure Cloud Storage with Provenance," in Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442-464.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Veri_ably Encrypted Signatures from Bilinear Maps," in Proceedings of Advances in Cryptology - EUROCRYPT â€.03, ser. LNCS, vol. 2656. Springer, 2003, pp. 416-432.

[7] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and E_cient Key Management for Access Hierarchies," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.

[8] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," in Proceedings of ACM Workshop on Cloud Computing Security (CCSW '09). ACM, 2009, pp. 103-114.

[9] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," in Proceedings of Information Security and Cryptology (Inscrypt '07), ser. LNCS, vol. 4990. Springer, 2007, pp. 384-398.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data," in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06). ACM, 2006, pp. 89-98.

[11] S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Transactions on Computer Systems (TOCS), vol. 1, no. 3, pp. 239-248, 1983