

SECURE FILE STORAGE WITH HYBRID CRYPTOGRAPHICAL AND FRAGMENTATION SYSTEM

¹MS. C. ARCHANA, ²MOHD. OBAIDULLAH ANSARI, ³SHAIK AFTABUDDIN, ⁴MOHD KHAN

¹Assistant Professor, Dept.of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad,

^{2,3,4}BTech Student, Dept.of CSE, Teegala Krishna Reddy Engineering College, Meerpet, Hyderabad

obaidansary4u@gmail.com , aftab.shaik13@gmail.com , ohdkhan150866@gmail.com

ABSTRACT: *When you are storing our data in a public cloud, securing your data becomes a big challenge. Our data will be stored in remote cloud servers. In this case we have obeyed the provider license agreements. We need to trust providers blindly. So, it is very important to secure our data with encryption. We are implementing a secure cloud storage system with AES, Triple DES and Blowfish algorithms by applying fragmentation. The secret agencies can use our systems to share information. In our project we have the modules named Administrator, Data owner, Data User and Cloud server. The data owner will upload the files into the system and double encryption is used on the file. The generated cipher text is going to be divided into seven fragments. These fragments will upload into the firebase cloud. The user can download the files by requesting the file key. The user will receive the key via email after a request processed by the data owner. If the key is valid the file will download. while downloading the seven fragments will combine as a single fragment and double description will apply on the file. The plain text will be downloaded as a text file. The cloud can track the upload and downloads, the cloud can view data owners and data user's details.*

Keywords: *Hybrid cryptography, AES, cloud computing, DES, Cloud security*

I. INTRODUCTION

Cloud storage services are widely used by individuals and organizations due to the inherent benefits offered by them, for

example, affordability, availability, mobility. Globalization and outsourcing in modern organizations and the increasing adoption of the social web in individual's life have increased the needs of sharing data in a

collaborative environment. For example, cloud storage services such as Drop Box [3], Google Drive [4] and Microsoft OneDrive [5] have been widely used. In such services, users must rely on the service-level agreement (SLA) to entrust cloud service providers to provide a level of protection to their data. However, SLA-based data protection is vulnerable to different types of threats ranging from legal to ethical. Examples of such threats include data sovereignty, third-party data exploitation, insider attack. This means users cannot trust these cloud service providers for their data. Hence, protecting data on such widely used cloud storage solutions remains a primary concern to their users. If the problem is left unaddressed, cloud storage providers might fail to keep users' data secure and a large amount of private and sensitive data might be revealed with very high consequential financial, reputational and operational impact on both users and cloud storage providers. One of the most promising solutions to protect the data in cloud storage is encryption. A user encrypts the data before uploading it to the cloud and decrypts the data after it is downloaded. An untrusted cloud storage provider can only see the

encrypted data. A number of solutions have been proposed [28, 40, 43] using the above-explained cryptographic method to protect user's data from an untrusted cloud storage. These simple solutions do not support multi-party collaboration between organizations and individuals without a help from a trusted third party. To address this shortcoming, attribute-based encryption (ABE) [14, 34] techniques are often utilized in data storage [8, 29, 39, 41, 44] to support a more fine-grained access control mechanism. ABE is a public key 2 cryptographic system, in which a user encrypts the data using the public keys which are shared with other users so that they can decrypt the encrypted data using their private keys. Particularly, users in ABE do not have unique keys as the traditional public key systems. Instead, they have a key associated with their properties. A user can set a complex access policy based on attributes defined in a system for the encrypted data. Only users with attributes that satisfy the access policy can decrypt the encrypted data. The access policy is easy to express and understand by users since ABE uses a Boolean formula as an access policy. Therefore, we adopted an ABE scheme in our system One of the

challenging problems in applying ABE to cloud storage systems is managing an up-to-date access control list. For example, a user who was once granted access to the data might not be able to access the data due to various reasons such as change in a security level, leaving an organization or becoming malicious. In such cases, the data owner must be able to revoke access to any encrypted data for invalid users. Two possible revocation methods can be considered. The first is an indirect revocation that revokes users by redistributing keys only to valid users after updating ciphertext. The second is a direct revocation in which the system revokes access to ciphertext for invalid users by updating ciphertext and revocation list without key redistributions. In our system, we employ both methods to support a flexible user revocation. An indirect revocation can be achieved in cloud storage solutions using existing ABE-based techniques (e.g., an epoch counter as proposed in [39]). When a user becomes invalid or there is a need to revoke access for a user, the system first increases the epoch counter in the user's key; then updates the corresponding ciphertext using

the increased counter; and finally redistributes keys containing the new epoch counter to all valid users. Since invalid users do not receive a key with the new epoch counter, their access to the corresponding data is automatically revoked. However, this revocation method requires the redistribution of new keys to all users every time when a user becomes invalid. Hence, this may cause inefficiency in the system if the revocation events occur frequently.

II. LITERATURE SURVEY

Attribute-based encryption (ABE) system enables an access control mechanism over encrypted data by specifying access policies among private keys and cipher texts. There are two flavors of ABE, namely key-policy and cipher text-policy, depending on which of private keys or ciphertexts that access policies are associated with. In this paper we propose a new cryptosystem called Broadcast ABE for both flavors. Broadcast ABE can be used to construct ABE systems with direct revocation mechanism. Direct revocation has a useful property that revocation can be done without affecting any non-revoked users; in particular, it does not require users to update keys periodically.

For key-policy variant, our systems appear to be the first fullyfunctional directly revocable schemes. For ciphertext-policy variant, our systems improve the efficiency from the previously best revocable schemes; in particular, one of our schemes admits ciphertext and private key sizes roughly the same as the currently best (non-revocable) ciphertext-policy ABE. Broadcast ABE can also be utilized to construct multi-authority ABE in the disjunctive setting.

Abstract. In functional encryption (FE) schemes, ciphertexts and private keys are associated with attributes and decryption is possible whenever key and ciphertext attributes are suitably related. It is known that expressive realizations can be obtained from a simple functional encryption flavor called inner product encryption (IPE), where decryption is allowed whenever ciphertext and key attributes form orthogonal vectors. In this paper, we construct public-attribute 4 inner product encryption (PAIPE) systems, where ciphertext attributes are public (in contrast to attribute-hiding IPE systems). Our PAIPE schemes feature constant size ciphertexts for the zero and non-zero evaluations of inner products. These

schemes respectively imply an adaptively secure identity-based broadcast encryption scheme and an identitybased revocation mechanism that both feature short ciphertexts and rely on simple assumptions in prime order groups. We also introduce the notion of negated spatial encryption, which subsumes non-zero-mode PAIPE and can be seen as the revocation analogue of the spatial encryption primitive of Boneh and Hamburg.

Online social networks (OSNs) are immensely popular, with some claiming over 200 million users. Users share private content, such as personal information or photographs, using OSN applications. Users must trust the OSN service to protect personal information even as the OSN provider benefits from examining and sharing that information. We present Persona, an OSN where users dictate who may access their information. Persona hides user data with attribute-based encryption (ABE), allowing users to apply fine-grained policies over who may view their data. Persona provides an effective means of creating applications in which users, not the OSN, define policy over access to private

data. We demonstrate new cryptographic mechanisms that enhance the general applicability of ABE. We show how Persona provides the functionality of existing online social networks with additional privacy benefits. We describe an implementation of Persona that replicates Facebook applications and show that Persona provides acceptable performance when browsing privacy-enhanced web pages, even on mobile devices.

In this paper, we construct an efficient “multi-receiver identity-based encryption scheme”. Our scheme only needs one (or none if precomputed and provided as a public parameter) pairing computation to encrypt a single message for n receivers, in contrast to the simple construction that re-encrypts a message n times using Boneh and Franklin’s identity-based encryption scheme, considered previously in the literature. We extend our scheme to give adaptive chosen ciphertext security. We support both schemes with security proofs under precisely defined formal security model. Finally, we discuss how our scheme can lead to a highly efficient public key broadcast

encryption scheme based on the “subset-cover” framework.

A pseudorandom function $F : K \times X \rightarrow Y$ is said to be key homomorphic if given $F(k_1, x)$ and $F(k_2, x)$ there is an efficient algorithm to compute $F(k_1 \oplus k_2, x)$, where \oplus denotes a group operation on k_1 and k_2 such as xor. Key homomorphic PRFs are natural objects to study and have several interesting applications: they can simplify the process of rotating encryption keys for encrypted data stored in the cloud, they give one round distributed PRFs, and they can be the basis of a symmetric-key proxy re-encryption scheme. Until now all known constructions for key homomorphic PRFs were only proven secure in the random oracle model. We construct the first provably secure key homomorphic PRFs in the standard model. Our main construction is based on the learning with errors (LWE) problem. In the proof of security we need a variant of LWE where query points are non-uniform and we show that this variant is as hard as the standard LWE. We also construct key homomorphic PRFs based on the decision linear assumption in groups with an ℓ -linear map. We leave as an open problem the

question of constructing standard model key homomorphic PRFs from more general assumptions.

We present new techniques for achieving adaptive security in broadcast encryption systems. Previous work on fully collusion resistant broadcast encryption with short ciphertexts was limited to considering only static security. First, we present a new definition of security that we call semi-static security and show a generic “two-key” transformation from semi-statically secure systems to adaptively secure systems that have comparable-size ciphertexts. Using bilinear maps, we then construct broadcast encryption systems that are semi-statically secure in the standard model and have constant size ciphertexts. Our semi-static constructions work when the number of indices or identifiers in the system is polynomial in the security parameter. For identity-based broadcast encryption, where the number of potential indices or identifiers may be exponential, we present the first adaptively secure system with sublinear ciphertexts. We prove security in the standard model.

III. PROPOSED SYSTEM

We introduce a system that offers a cryptographically enforced access control method on an un-trusted cloud storage. When a user uploads a file to a cloud storage service, our system takes over the file and encrypts it with a secret symmetric key (e.g., AES). It then encrypts the secret key separately using an ABE scheme. To maintain the integrity of the encrypted data, our system cryptographically signs both cipher texts. We call the encrypted file and the encrypted key data-block and meta-file, respectively. In our system, only users who satisfy the access policies in meta-file can decrypt the secret key. Hence, the access control policies can be enforced without any help from the cloud storage services, and the data file is cryptographically protected.

ADVANTAGES

1. Protecting the user’s data from untrusted cloud storage.
2. Users’ revocation will be done by the admin.

SYSTEM ARCHITECTURE

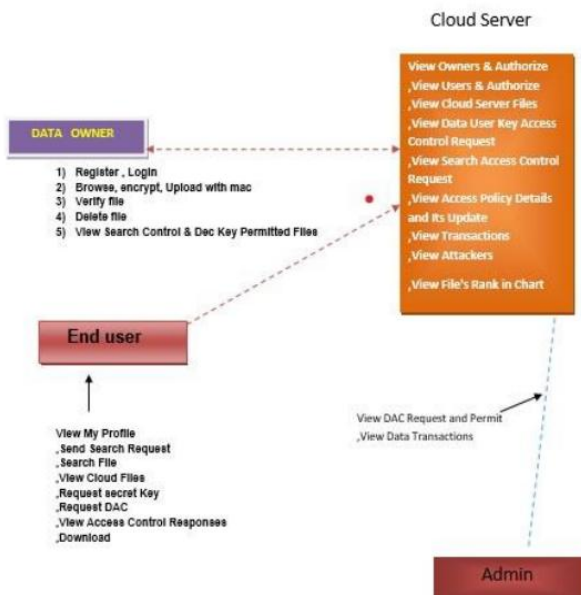


Fig.1 System architecture

secure management of data in cloud storage services, more specifically supporting multi-party sharing in the context of a collaboration, is a challenging problem. The problem is further exacerbated if the data owner does not have any trust on the cloud storage providers and the data need regular updates from collaborating parties. A number of cryptographically enforced secure cloud storage solutions have been proposed to address this problem. One of the key 10 issues with these solutions is the revocation of access to data for invalid users without moving the data (in the era of big data) and relying on the cloud service providers. The

system becomes inefficient as the time required for encryption/decryption increases proportionally to the number of revoked attributes.

Since this lookup work has already reached certain levels as witnessed from trendy works, few of the works had been studied and taken as an idea to construct a novel mannequin in this project. The strategies and effects in existing works have been noted in this file and properly stated as a token of appreciating the earlier works. Also, the code buildings acquired from the open supply boards are used solely for the academic purposes and no longer for any industrial benefits. Therefore, it is ensured that the occupation ethics have by no means been violated in this assignment for the duration of the undertaking tenure.

MODULES

1. User
2. Owner
3. Cloud server
4. Administrator

MODULAR DESCRIPTION

USER

The module should register with the application and the user should be authorized by the Cloud server. After the user login into the application then the user can perform such actions as view files, view writer list, download file.

OWNER

The owner also registers with the application and the owner should be authorized by the cloud server then the owner can upload the files into the cloud then view the files.

CLOUD SERVER

The cloud server can login directly with the application and the cloud performs some actions like view owners and authorize, view users and authorize, view files, and the cloud server view the request and generated the key.

ADMINISTRATOR

IV. RESULTS

OUTPUT SCREENS

The admin also can login directly with the application and the admin also perform some actions like view request, view file list, revoke list.


Introducing a system that offers a cryptographically enforced access control method on an un-trusted cloud storage. When a user uploads a file to a cloud storage service, our system takes over the file and encrypts it with a secret symmetric key (e.g., AES). It then encrypts the secret key separately using an ABE scheme. To maintain the integrity of the encrypted data, our system cryptographically signs both cipher texts. We call the encrypted file and the encrypted key data-block and metafile, respectively. In our system, only users who satisfy the access policies in meta-file can decrypt the secret key. Hence, the access control policies can be enforced without any help from the cloud storage services, and the data file is cryptographically protected

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/7.0.40

The Apache Software Foundation
http://www.apache.org/

If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:
[Security Considerations HOW-TO](#)
[Manager Application HOW-TO](#)
[Clustering/Session Replication HOW-TO](#)

Server Status
 Manager App
 Host Manager

Developer Quick Start
[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

Managing Tomcat
 For security, access to the [manager webpage](#) is restricted. Users are defined in:
 \$CATALINA_HOME/conf/tomcat-users.xml
 In Tomcat 7.0 access to the manager application is split between different users.

Documentation
[Tomcat 7.0 Documentation](#)
[Tomcat 7.0 Configuration](#)
[Tomcat Wiki](#)
 Find additional important configuration

Getting Help
[FAQ and Mailing Lists](#)
 The following mailing lists are available:
announce@tomcat.apache.org
 Important announcements, releases, security vulnerability notifications. (Low volume).

Fig.2 User Interface of Web Server Creation Application

Search our site:

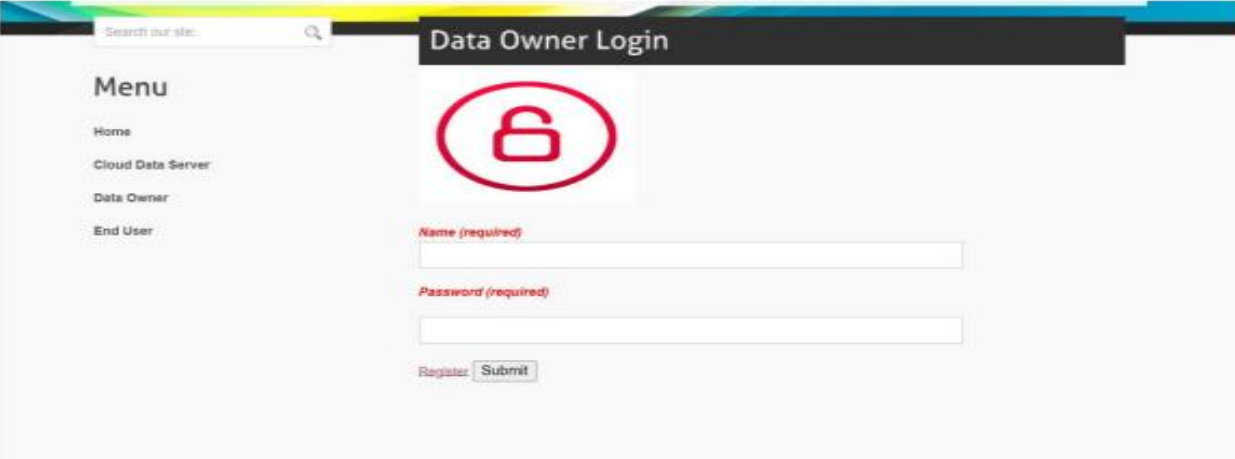
Secure File Storage With Hybrid Crypto Graphical And Fragmentation System



Cloud enabled data access control.

Enabling cryptographically enforced access controls for data hosted in untrusted cloud is attractive for many users and organizations. However, designing efficient cryptographically enforced dynamic access control system in the cloud is still challenging. In this paper, we propose Crypt-DAC, a system that provides practical cryptographic enforcement of dynamic access control. Crypt-DAC revokes access permissions by delegating the cloud to update encrypted data. In Crypt-DAC, a file is encrypted by a symmetric key list which records a file key and a sequence of revocation keys. In each revocation, a dedicated administrator uploads a new revocation key to the cloud and requests it to encrypt the file with a new layer of encryption and update the encrypted key list accordingly. Crypt-DAC proposes three key techniques to constrain the size of key list and encryption layers. As a result, Crypt-DAC enforces dynamic access control that provides efficiency, as it does not require expensive decryption/re-encryption and uploading/re-uploading of large data at the administrator side, and security, as it immediately revokes access permissions. We use formalization framework and

Figure 2: User Interface




Search our site:

Data Owner Login

Menu

- Home
- Cloud Data Server
- Data Owner
- End User



Name (required)

Password (required)

Register

Fig.3 Data Owner Login



Search our site:

Data User Register

Menu

- Home
- Cloud Data Server
- Data Owner
- End User



Name (required)

Password (required)

Email Address (required)

Mobile Number (required)

Your Address

DOB (required)

Gender (required)

Pincode

Fig.3 Data owner registration



Search our site:

Request Dec Key & Permission

Data User Menu
User Main
Log Out

Enter File Name :-	<input type="text" value="ob.txt"/>
Enter Owner Name :-	<input type="text" value="ansad"/>
<input type="button" value="Request"/>	

Fig. 4 Authorization of Reading and Writing the Data



Search our site:

Download Files

Data User Menu
User Main
Log Out

Enter File Name :-	<input type="text" value="ob.txt"/>
Enter Owner Name :-	<input type="text" value="ob"/>
Trapdoor :-	<input type="text" value="5050edba1c#00ba465ecta36edea2f8101bce4a"/>
Secret Key :-	<input type="text" value="!B@63aa16oe"/>
Status(Read)	<input type="text" value="Permitted"/>
Status(Write)	<input type="text" value="Permitted"/>
<input type="button" value="Download"/>	

[Back](#)

Fig.5 User download file



Search our site:

Download Files

Data User Menu
User Main
Log Out

```
hl hello 123 how are you?
```

[Back](#)

Fig.6 Downloading the Decrypted Data

V. CONCLUSION

In conclusion, the project "Secure File Storage with Hybrid Crypto Graphical And Fragmentation System" aims to provide an access control including both readers and writers control to the data stored in the cloud. Therefore, our system prevents unauthorized users including the cloud provider to read any plain texts of the files that are stored in the cloud. Authorized users can always access and decrypt the files. Though our system does not prevent the encrypted data in the cloud from being modified or replaced by malicious readers or cloud providers, it can detect such modification or replacement from unauthorized writers. We describe the security goals of our system more formally in terms of confidentiality, integrity and availability. Cloud storage services are widely used by individuals and organizations due to the inherent benefits offered by them, for example, affordability, availability, mobility. Globalization and outsourcing in modern organizations and the increasing adoption of the social web in

individual's life have increased the needs of sharing data in a collaborative environment. For example, cloud storage services such as Dropbox, Google Drive and Microsoft OneDrive have been widely used. In such services, users must rely on the service-level agreement (SLA) to entrust cloud service providers to provide a level of protection to their data. However, SLA-based data protection is vulnerable to different types of threats ranging from legal to ethical. Examples of such threats include data sovereignty, third-party data exploitation, and insider attack. This means users cannot trust these cloud service providers for their data. Hence, protecting data on such widely used cloud storage solutions remains a primary concern to their users. If the problem is left unaddressed, cloud storage providers might fail to keep users' data secure, and a large amount of private and sensitive data might be revealed with very high consequential financial, reputational and operational impact on both users and cloud storage providers.

REFERENCES

1. Attrapadung N, Hideki I (2009) Conjunctive broadcast and attribute-based encryption. In: Shacham H, Waters B (eds) Pairing, volume 5671 of LNCS. Springer, pp 248–265.
2. Attrapadung N, Libert B (2012) Functional encryption for public attribute inner products: achieving constant size ciphertexts with adaptive security or support for negation. *J Math Cryptol* 5(2):115–158.
3. Baden R, Bender A, Spring N, Bhattacharjee B, Starin D (2009) Persona: an online social network with user-defined privacy. In: Rodriguez P, Biersack EW, Papagiannaki K, Rizzo L (eds) ACM SIGCOMM. ACM, pp 135–146.
4. Baek J, Safavi-Naini R, Susilo W (2005) Efficient multi-receiver identitybased encryption and its application to broadcast encryption. In: *Public key cryptography*. pp 380–397.
5. Boneh D, Lewi K, Montgomery HW, Raghunathan A (2013) Key homomorphic prfs and their applications. In: Canetti R, Garay JA (eds) CRYPTO, volume 8042 of LNCS. Springer, pp 410–428.
6. Fiat A, Naor M (1993) Broadcast encryption. In: CRYPTO. pp 480–491.
7. Gentry C, Waters B (2009) Adaptive security in broadcast encryption systems (with short ciphertexts). In: EUROCRYPT. pp 171–188.
8. Goodrich MT, Sun JZ, Tamassia R (2004) Efficient tree-based revocation in groups of low-state devices. In: CRYPTO. pp 511–527.
9. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: Juels A, Wright RN, De Capitani di Vimercati S (eds) ACM conference on computer and 40 communications security. ACM, pp 89–98.
10. Hardt D (2012) The oauth 2.0 authorization framework. RFC 6749.
11. Afreen Bari, Dr. Prasadu Peddi. (2021). Review and Analysis Load Balancing Machine Learning Approach for Cloud Computing Environment. *Annals of the Romanian Society for Cell Biology*, 25(2), 1189–1195.