# A Strong and Testable Threshold Multi-Authority Access Regulation System in Public Cloud Storage

**Md Ateeq Ur Rahman[1] and L Shah Sarfaraz[2],**
[1]**Professor and Head, Dept. of CSE, SCET, Hyderabad**
mail_to_ateeq@yahoo.com
[2]**Research Scholar, Dept. of CSE, SCET, Hyderabad**
Shahsarfaraz1@gmail.com

**Abstract -**Attribute-based Cryptography (ABC) is found to be a promising cryptanalytic conducting tool to ensure information owners' direct management over their information in public cloud storage. Traditional ABC schemes involve just one authority to keep up the entire attribute set, which might bring a single-point bottleneck on each security and performance. Afterwards, some multi-authority schemes are projected, within which multiple authorities one by one maintain disjoint attribute subsets. However, the single-point bottleneck downside remains unsolved. In this paper, from another perspective, we tend to conduct a threshold multi-authority CP-ABC access management theme for public cloud storage, named TMACS, within which multiple authorities collectively manage a consistent attribute set. In TMACS, taking advantage of (t; n) threshold secret sharing, the master key is often shared among multiple authorities, and a legal user will generate his/her secret key by interacting with any t authorities. Security and performance analysis results show that TMACS isn't solely verifiable secure when less than t authorities are compromised, but additionally strong when no less than t authorities are within the system. Moreover, by expeditiously combining the normal multi-authority theme with TMACS, we tend to construct a hybrid one, that satisfies the situation of attributes returning from completely different authorities as well as achieving security and system-level robustness.

**Index Terms**—Hashtag graph, topic modeling, sparseness of short text, weakly-supervised learning

## I. INTRODUCTION

To satisfy needs of knowledge storage and high performance computation, cloud computing has drawn in depth attentions from each tutorial and trade. Cloud storage is a crucial service of cloud computing [1] that provides services for information house owners to source information to store in cloud via web. Despite several benefits of cloud storage, there still stay numerous difficult obstacles, among that, privacy and security of users' knowledge became major problems, particularly publicly cloud storage [2], [3]. Historically, information owner stores his/her data in sure servers that are usually controlled by a totally sure administrator. However, publicly cloud storage systems, the cloud is sometimes maintained and managed by a semi-trusted third party (the cloud provider). Knowledge is not any longer in information owner's sure domains and also the information owner cannot trust on the cloud server to conduct secure information access management. Therefore, the secure access management drawback has become a vital difficult issue publicly cloud storage, within which traditional security technologies can't be directly applied.

Attribute-based Cryptography (ABC) [4], [5], [6] is considered one in all the foremost appropriate schemes to conduct information access management in public clouds for it will guarantee information owners' direct management over their information and supply a fine-grained access management service. Till now, there are several ABCs schemes planned, which may be divided into 2 categories: Key-Policy Attribute-based Cryptography (KP-ABC), like, and Cipher text-Policy Attribute-based Cryptography (CP-ABC). In KP-ABC schemes, rewrite keys are related to access structures whereas ciphertexts are solely labeled with special attribute sets. On the contrary, in CP-ABC schemes, information homeowners will outline an access policy for every file supported users' attributes, which may guarantee owners' additional direct management over their information. Therefore, compared with KP-ABC, CP-ABC may be a most well-liked selection for planning access management for public cloud storage.

In most existing CP-ABC schemes there's just one authority chargeable for attribute management and key distribution [7]. This only-one-authority state of affairs will bring a single-point bottleneck on each security and performance. Once the authority is compromised, somebody will simply acquire the only-one-authority's passkey, and then he/she will generate non-public keys of any attribute set to rewrite the precise encrypted information. Moreover, once the only-one-authority is crashed, the system fully cannot work well. Therefore, these CP-ABC schemes are still off from being wide used for access management publically cloud storage. Though some multi-authority CP-ABC schemes are planned, they still cannot touch upon the matter of single-point bottleneck on each security and performance mentioned on top of. In these multi-authority CP-ABC schemes, the complete attribute set is split into multiple disjoint sets and every attribute subset continues to be maintained by just one authority. Though somebody cannot gain non-public keys of all attributes if he/she hasn't compromised all authorities, compromising one or additional authorities would create somebody have additional privileges than he/she ought to have. Moreover, when somebody acquire non-public keys of specific attributes by compromising specific one or additional authorities. Additionally, the only purpose bottleneck on performance isn't nevertheless resolved in these multi-authority

CP-ABC schemes [8]. Crash or offline of a particular authority can create that personal keys of all attributes in attribute set maintained by this authority can't be generated and distributed, which is able to still influence the complete system's effective operation[9].

In this paper, we have a tendency to propose a strong and verifiable threshold multi-authority CP-ABC access management theme, named TMACS, to touch upon the single-point bottleneck on each security and performance in most existing schemes[6]. In TMACS, multiple authorities collectively manage the complete attribute set however nobody has full management of any specific attribute. Since in CP-ABC schemes, there's perpetually a secret key (SK) wont to generate attribute non-public keys, we have a tendency to introduce (t; n) threshold secret sharing into our theme to share the key among authorities. In TMACS, we have a tendency to redefine the key within the ancient CP-ABC schemes as passkey. The introduction of (t; n) threshold secret sharing guarantees that the passkey can't be obtained by any authority alone. TMACS isn't solely verifiable secure once but authorities are compromised, however conjointly strong once no but t authorities are within the system. To the most effective of our information, this paper is that the 1st try and address the only purpose bottleneck on each security and performance in CPABC access management schemes publically cloud storage. Main contributions of this work may be summarized as follows:

- In existing access management systems for public cloud storage, there brings a single-point bottleneck on each security and performance against the only authority for any specific attribute. To the most effective of our information, we have a tendency to be the primary to style a multi authority access management design to touch upon the matter[10].
- By introducing the combining of (t; n) threshold secret sharing and multi-authority CP-ABC theme, we have a tendency to propose and notice a strong and verifiable multi authority access system publically cloud storage, during which multiple authorities collectively manage an identical attribute set.
- Furthermore, by with efficiency combining the standard multi-authority theme with ours, we have a tendency to construct a hybrid one, which may satisfy the state of affairs of attributes returning from completely different authorities still as achieving security and system-level robustness.

## II. RELATED WORKS

**Existing System**

There is just one authority accountable for attribute management and key distribution. This only-one-authority situation will bring a single-point bottleneck on each security and performance. Once the authority is compromised, Associate in nursing someone will simply acquire the only-one-authority's passkey, and then he/she will generate personal keys of any attribute set to decode the precise encrypted information. Crash or offline of a particular authority can build that non-public keys of all attributes in attribute set maintained by this authority can't be generated and distributed, which is able to still influence the complete system's effective operation.

**Advantages:**

This only-one-authority state of affairs will bring a single-point bottleneck on each security and performance. These CP-ABC schemes are still off from being wide used for access management publically cloud storage.

**Disadvantages:**

Crash or offline of a selected authority can build that non-public keys of all attributes in attribute set maintained by this authority cannot be generated and distributed, which is able to still influence the full system's effective operation.The access structure isn't versatile enough to satisfy advanced environments. Afterwards, abundant effort has been made to trot out the disadvantages within the early schemes.

## III. PROPOSED SYSTEM

In this section, we tend to first provide an outline of TMACS, as well as the theme structure and therefore the difficult problems within the style of TMACS. Within the following, we have a tendency to elaborate describe TMACS that chiefly consists of 4 phases: System data format, Secret Key Generation, Encryption, and coding.

To address the matter of single-point bottleneck, we have a tendency to introduce (t; n) threshold secret sharing, supported redundant multiple AAs, then propose a threshold multi-authority CPABC and therefore the relevant access management theme TMACS publically cloud storage.

In TMACS, the framework of the system is comparable to DAC-MACS projected by rule et al. the most distinction is: In DAC-MACS, the full attribute set is split into multiple disjoint sets and every one in all the multiple authorities maintains one attribute subset. In contrast, in TMACS, multiple authorities put together manage the full attribute set however nobody has full management of any specific attribute. In TMACS, a worldwide certificate authority is accountable for the development of the system that avoids the additional overhead caused by AAs' negotiation of system parameters. CA is additionally accountable for the registration of users that avoids AAs synchronic maintaining an inventory of users. However, CA isn't concerned in AAs' master key sharing and users' secret key generation, that avoids CA turning into the protection vulnerability and performance bottleneck.

The theme structure of TMACS also can be summarized. In TMACS, AAs should initially register to CA to achieve the corresponding identity and certificate (aid, aid.cert). Then AAs are going to be concerned within the construction of the system, aiding CA to end the institution of system parameters. CA accepts users' registration and problems the certificate (uid, uid.cert) to every legal user. With the certificate, the user will contract with any t AAs one by one to realize his/her secret key. Data owners who need share their knowledge within the cloud will gain the general public key from CA. Then the owner will cypher his/her knowledge underneath predefined access policy and transfer the ciphertext (CT) to the cloud server. User will freely transfer the ciphertexts that he/she is curious about from the cloud server. However, he/she can't decipher the ciphertext unless his/her attributes satisfy the access policy hidden within the ciphertexts.

One difficult issue in style of TMACS is reusing of the master key shared among multiple attribute authorities. In ancient (t; n) threshold secret sharing, once the key is reconstructed among multiple participants, somebody will truly gain its price. Similarly, in CP-ABC schemes, the only-one-authority is aware of the master key and uses it to come up with every user's secret key in line with a selected attribute set. During this case, if the AA is compromised by associate opposer, it'll become the protection vulnerability. To avoid this, by suggests that of (t; n) threshold secret sharing, the master key cannot be separately reconstructed and gained by any entity in TMACS. How to guarantee the pliability of the system in users' secret key generation is another difficult issue. In traditional (t; n) threshold secret sharing, the key will be reconstructed unless there are at least t participants cooperating with one another. this implies that, if simply merely introducing ancient (t; n) threshold secret sharing into our multi-authority CP-ABC style, the user ought to contact with t AAs throughout the key generation for every time, and therefore the chosen t AAs even have to contact with one another to implicitly reconstruct the passe-partout. This can bring an excessive amount of communication overhead, that isn't versatile for system acting.

To reduce the trivial communication overhead, in TMACS, instead of the master key, the whole secret key is reconstructed by grouping t secret key shares generated by AAs. Moreover, the reconstructed method will be done by the user instead of the particular t AAs. By this implies, the user will contact with the t AAs one by one, that is suit for real application situations, enhances the pliability of the system, avoids the additional communication overhead and synchronization problems among AAs.

## IV. SYSTEM ARCHITECTURE

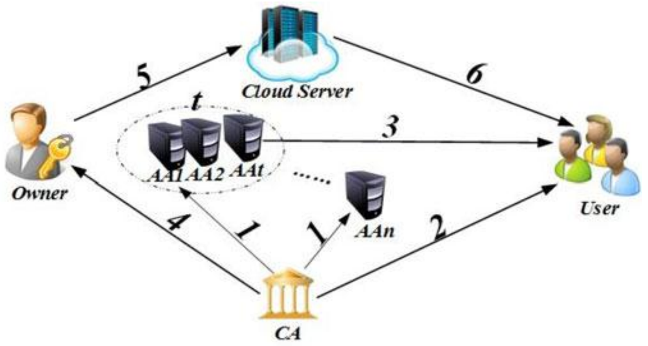


**Figure 1: System Architecture of the Proposed System**

In this section, we tend to offer the definitions of the system model in strong multi-authority public cloud storage systems. In strong multi-authority public cloud storage systems, there exist 5 entities: a worldwide certificate authority (CA), multiple attribute authorities (AAs), data owners (Owners), information consumers (Users), and also the cloud server.

1) The certificate authority may be an international trustworthy entity within the system that's accountable for the development of the system by fixing system parameters and attribute public key (PK) of every attribute within the whole attribute set. CA accepts users and AAs' registration requests by distribution a novel uid for every legal user and a novel aid for every AA. CA additionally decides the parameter t regarding the threshold of AAs that are concerned in users' secret key generation for every time. However, CA isn't concerned in AAs' master key sharing and users' secret key generation. Therefore, for instance, CA may be government organizations or enterprise departments that are accountable for the registration

2) The attribute authorities target the task of attribute management and key generation. Besides, AAs participate of the responsibility to construct the system, and that they may be the directors or the managers of the applying system. Totally different from different existing multi-authority CP-ABC systems, all AAs collectively manage the entire attribute set, however, anyone of AAs cannot assign users' secret keys alone for the passe-partout is shared by all AAs. All AAs work with one another to share the master key. By this means, every AA will gain a chunk of master key share as its personal key, then every AA sends its corresponding public key to CA to get one of the system public keys. Once it involves generate users' secret key, every AA solely ought to generate its corresponding secret key severally. That's to mention, no communication among AAs is required within the part of users' secret key generation.

3) The information owner (Owner) encrypts his/her file and defines access policy regarding who will get access to his/her information. First of all, every owner encrypts his/her information with a radially symmetrical encoding formula like AES and DES. Then the owner formulates access policy over associate attribute set and encrypts the radially symmetrical key underneath the policy consistent with attribute public keys gained from CA. Here, the symmetric key's the key employed in the previous method of symmetric encoding. After that, the owner sends the entire encrypted knowledge and therefore the encrypted radially symmetrical key to store within the cloud server. However, the owner doesn't consider the cloud server to conduct knowledge access management. Knowledge keep within the cloud server may be gained by any knowledge shopper. Despite all this, no information consumer will gain the plaintext while not the attribute set satisfying the access policy.

4) The information client (User) is allotted with a global user identity uid from CA, and applies for his/her secret keys from AAs with his/her identification. The user will freely get the ciphertexts that he/she is inquisitive about from the cloud server. He/She will decipher the encrypted information if and providing his/her attribute set satisfies the access policy hidden within the encrypted information.

5) The cloud server will nothing however give a platform for house owners storing and sharing their encrypted information. The cloud server doesn't conduct information access management for owners. The encrypted information keep

within the cloud server may be downloaded freely by any information client.

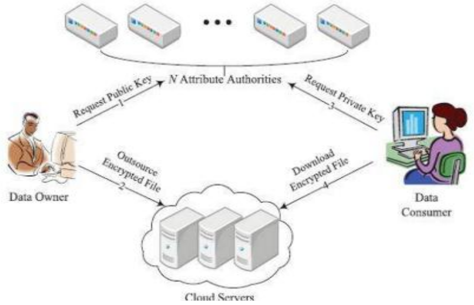


**Figure 2: Process flow of the proposed System**

**Module Description**

In this project, A Strong and Testable Threshold Multi-Authority Access Regulation System in Public Cloud Storage, we have three modules.

- User module
- Multi-authorityAccess control
- Public cloud storage.

**User Module:**

In this module, Users are having authentication and security to access the detail that is bestowed within the system. Before accessing or looking out the main points user ought to have the account therein otherwise they ought to register initially.

**Multi-authority Access control:**

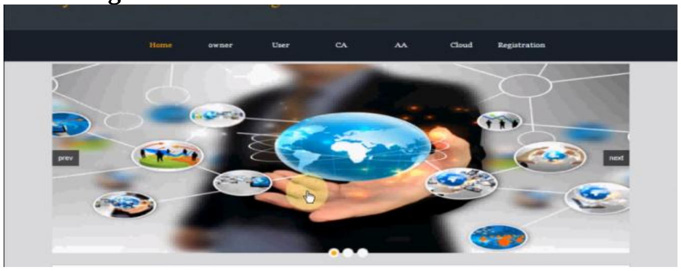
We conduct a threshold multi-authority CP-ABC access management theme for public cloud storage, named TMACS, during which multiple authorities collectively manage the same attribute set to the most effective of our data, we are the first to design a multi-authority access management design to affect the matter. To satisfy this hybrid situation, we tend to conduct a hybrid multi-authority access management theme, by combining the normal multi-authority theme with our planned TMACS.

**Public Cloud Storage:**

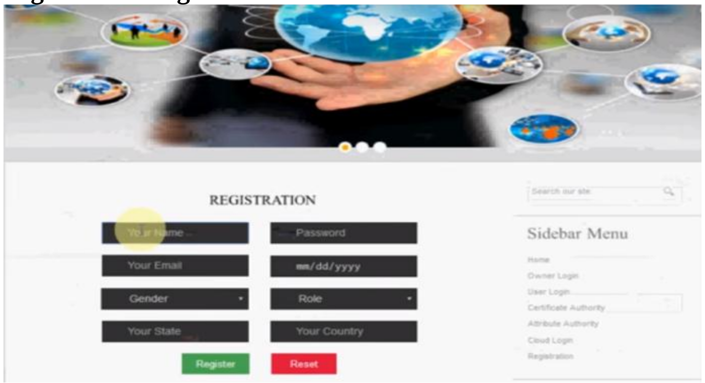
Cloud storage is a vital service of cloud computing that provides services for information owners to source information to store in cloud via web. The cloud server is often on-line and managed by the cloud provider. Usually, the cloud server and its provider are assumed “honest-but-curious”. The cloud server wills nothing but give a platform for owners storing and sharing their encrypted information. The cloud server doesn’t conduct information access management for owners.

## V. SAMPLE OUTPUT SCREENS

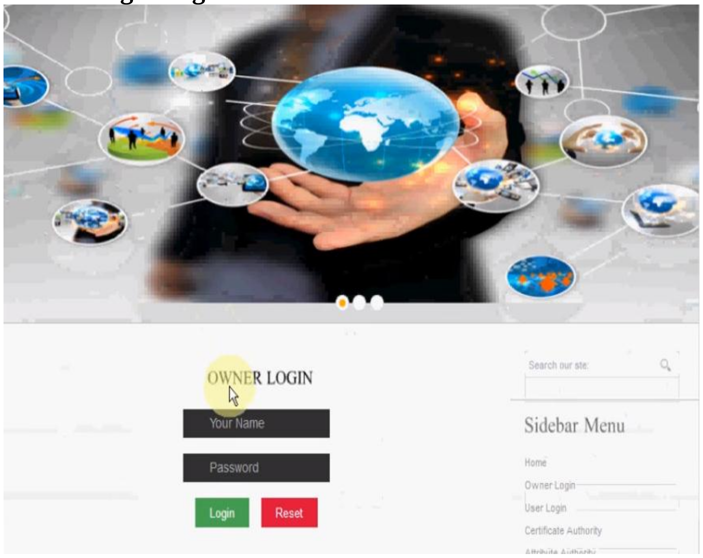
**Home Page**



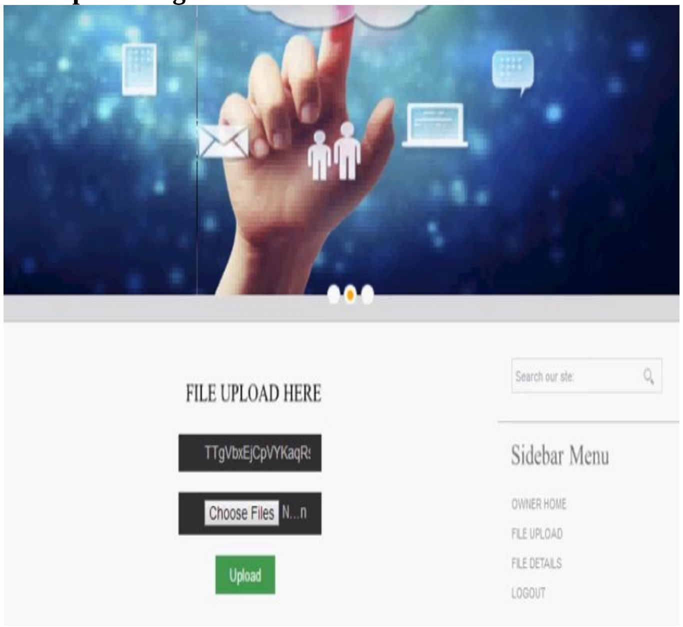
**Registration Page**



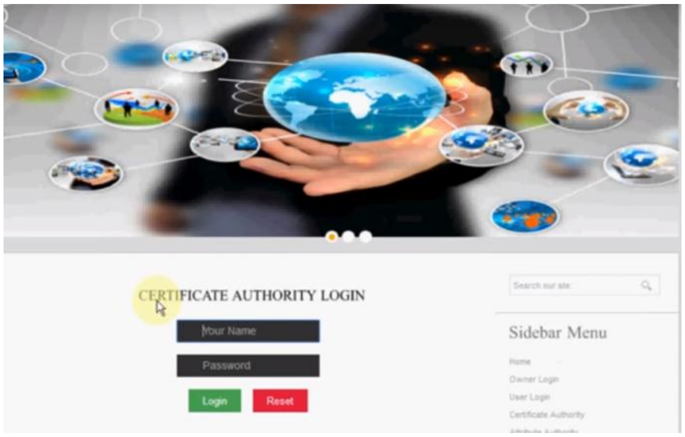
**Owner Login Page**



**File Upload Page**

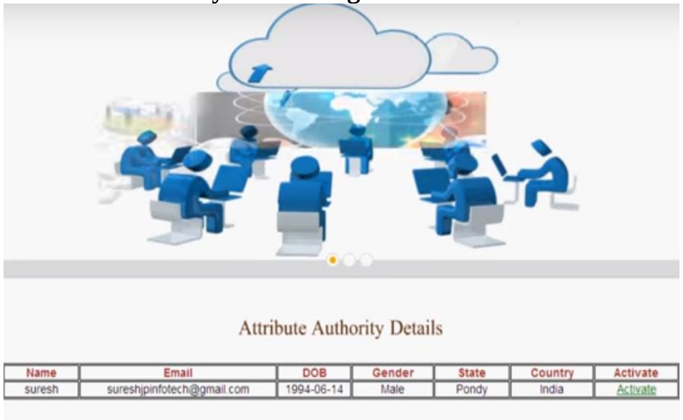


**Certificate Authority Login Page**



CERTIFICATE AUTHORITY LOGIN

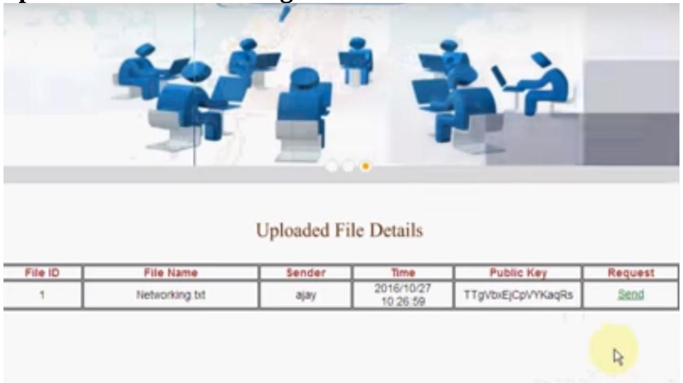
**Attribute Authority Details Page**



Attribute Authority Details

| Name | Email | DOB | Gender | State | Country | Activate |
|---|---|---|---|---|---|---|
| suresh | sureshjpinfotech@gmail.com | 1994-06-14 | Male | Pondy | India | Activate |

**Uploaded File Details Page**



Uploaded File Details

| File ID | File Name | Sender | Time | Public Key | Request |
|---|---|---|---|---|---|
| 1 | Networking.txt | ajay | 2016/10/27<br>10:26:59 | TTgVbxEjCpVYKaqRs | Send |

## VI. CONCLUSION

In this paper, we tend to propose a novel threshold multi-authority CP-ABC access management theme, named TMACS, in public cloud storage, within which all AAs collectively manage the full attribute set and share the master key a. Taking advantage of (t; n) threshold secret sharing, by interacting with any t AAs, a legal user will generate his/her secret key. Thus, TMACS avoids anyone AA being a single-point bottleneck on each security and performance. The analysis results show that our access management theme is powerful and secure. We can simply realize applicable values of (t; n) to create TMACS not solely secure once but t authorities are compromised, however conjointly strong once no but tauthorities are within the system. what is more, supported with efficiency combining the normal multi-authority theme with TMACS, we tend to additionally construct a hybrid theme that's additional appropriate for the important situation, during which attributes come back from completely different authority-sets and multiple authorities in an authority-set collectively maintain a set of the full attribute set. This better theme addresses not solely attributes coming back from completely different authorities but conjointly security and system- level robustness. The way to fairly choose the values of (t; n) in theory and design optimized interaction protocols are going to be addressed in our future work.

## REFERENCES

[1] P. Mell and T. Grance, “The NIST definition of cloud computing,” Nat. Instit. Standards Technol., vol. 53, no. 6, p. 50, 2009.

[2] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in Proc. 14th Financial Cryptography Data Security, 2010, pp. 136–149.

[3] K. Ren, C. Wang, and Q. Wang, “Security challenges for the public cloud,” IEEE Internet Comput., vol. 16, no. 1, pp. 69–73, Jan.-Feb. 2012.

[4] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Proc. 24th Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2005, pp. 457–473.

[5] R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-based encryption with non-monotonic access structures,” in Proc. 14th ACM Conf. Comput. Commun. Security, 2014, pp. 195–203.

[6] TMACS: A Robust and Verifiable ThresholdMulti-Authority Access Control Systemin Public Cloud Storage

[7] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters,“Fully secure functional encryption: Attribute-based encryptionand (hierarchical) inner product encryption,” in Proc. 29th Annu.Int. Conf. Theory Appl. Cryptographic Techn., 2010, pp. 62–91.

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-basedencryption for fine-grained access control of encrypted data,” inProc. 13th ACM Conf. Comput. Commun. Security, 2006, pp. 89–98.

[9] N. Attrapadung, B. Libert, and E. Panafieu, “Expressive keypolicyattribute-based encryption with constant-size ciphertexts,”in Proc. 14th Int. Conf. Practice Theory Public Key Cryptography,2011, pp. 90–108.

[10] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in Proc. IEEE Symp. Security Privacy, 2007,pp. 321–334.