ABE ENFORCED TRIPLE DES WITH OUTSOURCED DECRYPTION IN CLOUD

Miruthuladevi.N[1], Priyanga.P[2], Ramya.R[3], M.Shobana[4]
[1], [2],[3] B.E. Department of Computer Science and Engineering
SNS College of Technology
[4]Assistant Professor
Department of Computer Science and Engineering
SNS College of Technology

Abstract—Attribute-based encryption allow user to encrypt data using public-key then it decrypt data using private-key in the cloud. ABE provide the public-key one-to-many encryption. Access polices and user attributes are associated with private keys and ciphertexts. The drawbacks of the existing ABE schemes are that decryption involves expensive pairing operations, the number of such operations grows with the complexity of the access policy and the transformations are performed only in the data. Recent ABE system with outsourced decryption largely eliminates the decryption overhead for users. Security of that system ensures that an adversary, it guarantee the correctness of the transformation done by the cloud. In this paper, we consider a new requirement of ABE with enforced triple DES. In our scheme the triple DES is applied to the cloud environment because it provides the additional security by applying symmetric key encryption to the key. This model of ABE with verifiable outsourced decryption and propose a concrete scheme. We prove that our new scheme is both secure and verifiable, without relying on random oracles. Finally, we show an implementation of our scheme and result of performance measurements.

Key Terms — Attribute-based encryption, outsourced decryption, verifiability, Triple DES.

I. INTRODUCTION

The cloud environment is a distributed computing over a network. In this environment an untrusted servers, such as the cloud server, many applications need mechanisms for complex access-control over encrypted data this issue is addressed as the notion of attribute-based encryption (ABE). ABE is a new public key based one-to-many encryption that enables access control over encrypted data using access policies and ascribed attributes associated with private keys and ciphertexts. There are two kinds of ABE schemes: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In a KP-ABE scheme enables senders to encrypt messages under a set of attributes and private keys are associated with access structures that specify which ciphertexts the key holder will be allowed to decrypt. In CP-ABE scheme a system for realizing complex access control on encrypted data by using this encrypted data can be kept confidential even if the storage server is untrusted.

One of the main efficiency drawbacks of the most existing ABE schemes is that decryption is expensive for resource-limited devices due to pairing operations, and the number of pairing operations required to decrypt a ciphertext grows with the complexity of the access policy. Recently proposed system eliminates this problem by introducing the notion of ABE with outsourced decryption, which largely eliminates the decryption overhead for users and provides an untrusted server, say a proxy operated by a cloud service provider, with a transformation key K that allows the latter to translate any ABE ciphertext ACT satisfied by that user's attributes or access policy into a simple ciphertext CT, and it only incurs a small overhead for the user to recover the plaintext from the transformed ciphertext CT.

The security property of the ABE scheme with outsourced decryption guarantees that an adversary (including the malicious cloud server) be not able to learn anything about the encrypted message and provides of the correctness of the transformation done by the cloud server. However in such system the transformation key is visible to the users. If, the third party identifies the transformation key K then there is a possibility of known the information. In order to avoid such situation our scheme proposes the triple DES concept in order to encrypt the key. In our system the key has been encrypted in the form of ciperkey CK.

II. CP - ABE WITH OUTSOURCED DECRYPTION

An existing CP-ABE scheme consists of the following four algorithms:

• *Setup(ζ,U)* takes as input a security parameter ζ and an attribute universe description *U*. It outputs the public parameters *PP* and a master secret key *SK*.

• *KeyGen(PP,S,A)* takes as input the public parameters *PP*, the master secret key *SK* and a set of attributes *A*. It outputs a private key *PK*.

• *Encrypt(PP.M,AS)* takes as input the public parameters *PP*, a message *M* and an access structure *AS*. It outputs a ciphertext *CT*.

• *Decrypt(PP,PK,CT)* takes as input the public parameters *PP*, a private key *PK* for user and a ciphertext *CT*. It outputs a message *M*.

Let $(PP,SK) \leftarrow Setup(1^{\zeta})$, $PK \leftarrow KeyGen(PP,SK,A)$. For correctness, we require the following to hold:

1) If the set *S* of attributes satisfies the access structure *AS*, then $M \leftarrow Decrypt(PP,PK,CT)$;

2) Otherwise, *Decrypt(PP,PK,CT)* outputs the error symbol $\oplus$.

We now give the definition of in-distinguish ability under chosen-ciphertext attack (CCA security) for CP-ABE scheme. This is described by a game between a challenger and an adversary *AD*.

The game proceeds as follows:

• *Setup* The challenger runs *Setup* algorithm to obtain the public parameters *PP* and a master secret key *SK*. It gives the public parameters *PP* to the adversary *AD* and keeps *SK* to itself.

• *Query phase 1* The challenger initializes an empty set φ .The adversary *AD* adaptively issues queries:

1) *Private key* query, on input a set of attributes *A*: The challenger runs $PK \leftarrow KeyGen(PP,SK,A)$ and sets $D=D\cup\{S\}$. It then returns to the adversary the private key *PK*.

2) *Decryption* query, on a set of attributes *S* and a ciphertext *CT*: The challenger runs $PK \leftarrow KeyGen(PP,SK,A)$ and $M \leftarrow Decrypt(PP,PK,CT)$. It then returns *M* to the adversary.

• *Challenge* The adversary *AD* submits two (equal length) messages M_1 and M_2 an access structure *AS*, subject to the restriction that, for all $S\in D$, *AD* cannot be satisfied by *S*. The challenger selects a random bit $B\in\{1,2\}$, sets $ACT=Encrypt(PP,M_B,A)$ and sends *ACT* to the adversary as its challenge ciphertext.

• *Query phase 2* The adversary continues to adaptively issue *Private key* and *Decryption* queries, as in Query phase 1, but with the restrictions that the adversary cannot

1) issue a *Private key* query that would result in a set of attributes *S* which satisfies the access structure *AS* being added to *D*.

2) issue a *Decryption* query on a set of attributes and a ciphertext *CT* such that *S* satisfies *A* and

CT=ACT.

• *Guess* The adversary *A* outputs its guess *B'=B* for *B* and wins the game if *B=B'*.

The advantage of the adversary in this game is defined as $|Pr[B=B']-1/2|$ where the probability is taken over the random bits used by the challenger and the adversary.

CPA Security: We say that a CP-ABE scheme is *CPA-secure* (or secure against chosen-plaintext attacks) if the adversary cannot make decryption queries.

Selective Security: We say that a CP-ABE scheme is *selectively secure* if we add an **Init** stage before **Setup** where the adversary commits to the challenge access structure *A*.

III. PROPOSED CP-ABE SCHEME WITH ENFORCED TRIPLE DES WITH OUTSOURCED DECRYPTION

CP-ABE scheme with outsourced decryption consists of five algorithms: *Setup, Encrypt, KeyGen*$_{out}$*, Transform* and *Decrypt*$_{out}$. A trusted party uses the algorithm *Setup* to generate the public parameters and a master secret key, and uses *KeyGen*$_{\text{out}}$ generate a private key and a transformation key for a user. Taking as input the transformation key given by a user and a ciphertext, the cloud can use the algorithm *Transform* to transform the ciphertext into a simple ciphertext if the user's attribute satisfies the access structure associated with the ciphertext; then the user uses the algorithm *Decrypt*$_{out}$ to recover the plaintext from the transformed ciphertext. Where *Decrypt*$_{out}$ includes only the private key of the user and the transformed ciphertext, but does not include the original ciphertext. Because of this omission of the original ciphertext, it is not possible to construct a CP-ABE scheme with verifiable outsourced decryption. A malicious cloud could replace the ciphertext it supposes to transform with a ciphertext of a different message, and then transform the latter into a simple ciphertext using its transformation key. Obviously, the user cannot detect this malicious behavior of the cloud since the input to the algorithm *Decrypt*$_{out}$ does not include the original ciphertext required to be transformed. In this scheme *GenKey*$_{out}$ the key has been encrypted into the cipherkey by using this user cannot identify the key. So, it will provide the more security into the system. A CP-ABE scheme with outsourced decryption consists of the following seven algorithms:

• *Setup(ζ,U)* takes as input a security parameter ζ and an attribute universe description *U*. It outputs the public parameters *PP* and a master secret key *SK*.

• *KeyGen(PP,S,A)* takes as input the public parameters *PP*, the master secret key *SK* and a set of attributes *A*. It outputs a private key *PK*.

• *Encrypt(PP.M,AS)* takes as input the public parameters *PP*, a message *M* and an access structure *AS*. It outputs a ciphertext *CT*.

• *Decrypt(PP,PK,CT)* takes as input the public parameters *PP*, a private key *PK* for user and a ciphertext *CT*. It outputs a message *M*.

• $GenTK_{out}(PP,PK)$ takes as input the public parameters *PP* and a private key *PK* for *S*. It outputs a transformation key *K* and the corresponding retrieving key *RK*.

• $GenKey_{out}(PP,K)$ takes as input the public parameters *PP* and a transformation key *K*. It outputs a decrypted transformation key *K'* and the corresponding retrieving key *RK*.

•$Transform_{out}(PP,CT,K')$ takes as input the public parameters *PP*, a ciphertext *CT* and a transformation key *K* for S . It outputs a partially decrypted ciphertext *ACT*.

•$Decrypt_{out}(PP,CT,ACT,RK)$ takes as input the public parameters *PP*, a ciphertext *CT*, a partially decrypted ciphertext *ACT* and a retrieving key *RK* for S . It outputs a message *M*.

Let$(PP,SK) \leftarrow Setup(1^{\zeta})$, $PK \leftarrow KeyGen(PP,SK,A)$, $CT \leftarrow Encrypt(PP,M,AS)$, $(K,RK) \leftarrow GenTK_{out}(PP,PK)$, $K' \leftarrow GenKey_{out}(PP,K)$ and $CT' \leftarrow Transform_{out}(PP,CT,K')$. For correctness, we require the following to hold:

1) If the set *S* of attributes satisfies the access structure *AS*, then $M \leftarrow Decrypt(PP,PK,CT)$ and $M \leftarrow Decrypt_{out}(PP,CT,ACT,RK)$;

2) Otherwise, $Decrypt(PP,PK,CT)$ and $Decrypt_{out}(PP,CT,ACT,RK)$ output the error symbol $\oplus$.

In our new model, the algorithms *Setup, KeyGen, Encrypt* and *Decrypt* constitute a traditional CP-ABE scheme. The input to the algorithm $Decrypt_{out}$ includes the original ciphertext and the transformed ciphertext. In fact, in our concrete scheme, a user only needs to know a small part of the original ciphertext to verify the correctness of the transformation done by the cloud in the algorithm $Decrypt_{out}$. In addition, in our model, using the algorithm $GenTK_{out}$ and his private key, the user generates the transformation key by himself, not by the trusted party. Having either the trusted party or the user generate the transformation key does not have an effect on the security of the scheme. However, it is more flexible if we let the user himself generate the transformation key. This transformation key has been encrypted into encrypted key by using the $GenKey_{out}$. Imagine that a user doesn't know whether he will outsource decryption of his stored files or not in the future. At the setup stage of our proposed ABE with verifiable outsourced decryption system, the user can just initialize an ordinary ABE system without outsourced decryption. Then, the user can generate the transformation key himself whenever he wants to outsource decryption, without having to resetup of the whole system. On the other hand, if the trusted party is responsible for the generation of transformation keys, the user is required to reinitialize the system for outsourced decryption.

Now, we formally describe the security and verifiability requirements of a CP-ABE scheme with outsourced decryption. Informally, security ensures that an adversary (including a malicious cloud) not be able to learn anything about the encrypted message and verifiability allows a user to check on the correctness of the transformation done by the cloud.

Security. Since the traditional notion of security against adaptive chosen-ciphertext attacks (CCA) does not allow any bit of the ciphertext to be altered called replayable CCA (RCCA) security, which allows modifications to the ciphertext provided they cannot change the underlying message in a meaningful way. The RCCA security for CP-ABE with outsourced decryption is described as a game between a challenger and an adversary. The RCCA security game proceeds as follows:

• *Setup* The challenger runs algorithm to obtain the public parameters *PP* and a master secret key SK. It gives the public parameters *PP* to the adversary *AD* and keeps to itself.

• *Query Phase:* The challenger initializes an empty table *T* and an empty set φ. The adversary *AD* adaptively issues queries:

1) *Private key* query, on input a set of attributes *S*: The challenger runs $PK \leftarrow KeyGen(PP,SK,A)$ and sets $D=D\cup\{S\}$. It then returns to the adversary *AD* the private key *PK*.

2) *Transformation key* query, on input a set of attributes *S*: The challenger searches the entry *(S,PK,K,RK)* in table *T* . If such entry exists, it returns the transformation key *K*. Otherwise, it runs $PK \leftarrow KeyGen(PP,SK,A)$, $CT \leftarrow Encrypt(PP,M,AS)$, $(K,RK) \leftarrow GenTK_{out}(PP,PK)$, $K' \leftarrow GenKey_{out}(PP,K)$ and $CT' \leftarrow Transform_{out}(PP,CT,K')$ stores in table the entry *T*. It then returns to the adversary the transformation key *K*.

Without of loss of generality, we assume that an adversary do not issue *Transformation key* query on a set of attributes *S*, if it has already issued a *Private key* query on the same set of attributes *S*. Since anyone can by himself generate a transformation key for a user using the algorithm $GenTK_{out}$ and the user's private key, our assumption is reasonable.

3) *Decryption* query, on input a set of attributes *S* and a ciphertext *CT*: The challenger runs *PK* ← *KeyGen(PP,SK,CT)* and *M* ← $Decrypt_{out}$*(PP,CT,ACT,RK)*. It then returns to the adversary *AD*.

4) $Decryption_{out}$ query, on input a set of attributes *S* and a pair of ciphertexts (*CT,ACT*): The challenger searches the entry *(S,PK,K,RK)* in table *T*. If such entry exists, it runs *M* ← $Decrypt_{out}$*(PP,CT,ACT,RK)* and returns to the adversary *AD*; otherwise, it returns ξ.

•*Challenge* The adversary *AD* submits two (equal length) messages M_1, M_2 and an access structure *AS*, subject to the restriction that, for all *S*∈*D*, *A* cannot be satisfied by *S*. The challenger selects a random bit *B*∈{1,2}, sets *ACT =Encrypt(PP,*M_B*,AS)* and sends *ACT* to the adversary as its challenge ciphertext.

•*Query Phase 2:* The adversary continues to adaptively issue *Private Key, Transformation Key, Decryption,* $Decryption_{out}$ queries, as in Query phase 1, but with the restrictions that the adversary cannot

1) issue a *Private key* query that would result in a set of attributes *S* which satisfies the access structure *AS* being added to *D*.

2) issue a trivial decryption query. That is, *Decryption* and $Decryption_{out}$ queries will be answered as in Query phase 1, except that if the response would be either M_1 or M_2, then the challenger responds with the error symbol ⊕.

•*Guess* The adversary *AD* outputs its guess *B'* ∈ {1,2} for *B* and wins the game if *B=B'*.

The advantage of the adversary in this game is defined as |*Pr[B=B']-1/2*| where the probability is taken over the random bits used by the challenger and the adversary.

Definition: A CP-ABE scheme with outsourced decryption is RCCA-secure if all polynomial time adversaries have at most a negligible advantage in this security game.

CPA Security: We say that a CP-ABE scheme with outsourced decryption is *CPA-secure* (or secure against chosen-plaintext attacks) if the adversary cannot make decryption queries.

Selective Security: We say that a CP-ABE scheme with outsourced decryption is *selectively secure* if we add an **Init** stage before **Setup** where the adversary commits to the challenge access

Structure *AS*.

Verifiability. Verifiability of CP-ABE with outsourced decryption is also described by a game between a challenger and an adversary. The game proceeds as follows:

• *Setup* The challenger runs *Setup* algorithm to obtain the public parameters *PP* and a master secret key *SK*. It gives the public parameters *PP* to the adversary *AD* and keeps *SK* to itself.

• *Query Phase 1* The challenger initializes an empty table *T*. The adversary *AD* adaptively issues queries:

1) *Private key* query, on input a set of attributes *S*: The challenger runs *PK* ← *KeyGen(PP,SK,A)* and returns to the adversary the private key *PK*.

2) *Transformation key* query, on input a set of attributes *S*: The challenger runs *PK* ← *KeyGen(PP,SK,A), CT* ← *Encrypt(PP,M,AS), (K,RK)* ← $GenTK_{out}$*(PP,PK)* , *K'* ← $GenKey_{out}$*(PP,K)* and *CT'* ← $Transform_{out}$*(PP,CT,K')* stores in table T the entry *(S,PK,K,RK)*. It then returns to the adversary the transformation key *K*.

Without of loss of generality, we assume that an adversary does not issue *Transformation key* query on a set of attributes *S*, if it has already issued a *Private key* query on the same set of attributes *S*. Since anyone can by himself generate a transformation key for a user using the algorithm and the user's private key, our assumption is reasonable.

3) *Decryption* query, on input a set of attributes *S* and a ciphertext *CT*: The challenger runs *PK* ← *KeyGen(PP,SK,A)* and *M* ← *Decrypt(PP,SK,CT)*. It then returns to the adversary *M*.

4)$Decryption_{out}$ query, on input a set of attributes *S* and a pair of ciphertexts *(CT,ACT)*: The challenger searches the entry *(S,PK,K,RK)* in table *T*. If such entry exists, it runs *M* ← *Decrypt(PP,SK,CT)* and returns *M* to the adversary; otherwise, it returns ξ.

• *Challenge* The adversary *AD* submits a message *M'* and an access structure *AS*. The challenger sets *ACT=Encrypt(PP,M',AS)* and sends *ACT* to the adversary.

• *Query Phase 2* The adversary continues to adaptively issue *Private key, Transformation key, Decryption* and $Decryption_{out}$ queries, as in Query phase 1.

•*Output* The adversary *AD* outputs a set of attributes *S* and a transformed ciphertext *ACT* . We assume that entry *(S',PP,K,RK)* exists in table (If not, the challenger can generate the entry as in the response of *Transformation key* query). The adversary wins the game if $Decrypt_{out}$*(PP,ACT,CT,RK)* ∉{*M',* ξ}.

The advantage of the adversary in this game is defined as *Pr [*AD_{wins}*]* where the probability is taken over the random bits used by the challenger and the adversary.

Definition: A CP-ABE scheme with outsourced decryption is verifiable if all polynomial time adversaries have at most a negligible advantage in the above game.

One stronger notion of verifiability is that, even if the trusted party who setups the system is malicious, a user still can check on the correctness of the transformation done by the cloud. That is, the adversary generates the system's public parameters and master secret key by himself in the above game. Nevertheless, it is difficult to construct a CP-ABE scheme with outsourced decryption which is verifiable in such stronger model, since most existing techniques of provable security need to generate the system's public parameters *elaborately* by the challenger. This challenging problem will be left as one of our future research topics.

IV. RELATED WORKS

As referred by the paper [4] it states, the existing ABE schemes are that decryption involves expensive pairing operations. The number of such operations grows with the complexity of the access policy and the encryption and decryption is done in the same cloud and the paper [5] the encryption and decryption message is based on user attributes. The size of the cipher text is directly proportional to the complexity and the time required to decrypt grows with the complexity of the access formula. The paper [3] provides an idea that, access control not only fine-grained and it also full delegation and high performance. To efficiently revoke access rights from users. In paper [6] has been invokes an idea that, Secure multi-owner data sharing scheme for dynamic groups in the cloud. It states that, the dynamic broadcast encryption techniques, any cloud user data can anonymously share data with others and the Sharing data while preserving data and identity privacy is a challenging issue. The paper [1] proposes that Secure multi-owner data sharing scheme for dynamic groups in the cloud. To perform dynamic broadcast encryption techniques, any cloud user data can anonymously share data with others and Sharing data while preserving data and identity privacy is a challenging issue. The paper [2] propose the ideas in Secure multi-owner data sharing scheme for dynamic groups in the cloud by dynamic broadcast encryption techniques, any cloud user data can anonymously share data with others and Sharing data while preserving data and identity privacy is a challenging issue.

V. CONCLUSION

In this paper, we considered a new requirement of ABE by enforcing the Triple DES with outsourced decryption in the cloud. We modified the original model of ABE with outsourced decryption. We also proposed a concrete ABE scheme with triple DES outsourced decryption and proved that it is secure and verifiable. Our scheme does not rely on random oracles. To assess the practicability of our scheme, we implemented it and conducted experiments in a simulated outsourcing environment. As expected, the scheme substantially reduced the computation time required for resource-limited devices to recover plaintexts. Thus the proposed Triple DES with Attribute Based Encryption will ensure security and privacy of data and also security will be enhanced by attribute based encryption with verifiable decryption.

VI.REFERENCES

[1] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, And Provably Secure Realization"
[2] B.Wang, Sherman S.M. Chow, M. Li, And H. Li, "Storing Shared Data On The Cloud Via Security-Mediator," In Proc. IEEE 2013
[3] G.Wang, Q.Liu, J.Wu, and M.Guo, "Hierarchical Attribute-Based Encryption And Scalable User Revocation For Sharing Data In Cloud Servers"
[4] Junzuo Lai, Robert H. Deng, Chaowen Guan, and Jian Weng,"Attribute-Based Encryption With Verifiable Outsourced Decryption", vol.8, no.8, august 2013
[5] M. Green, S. Hohenberger, and B.Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. USENIX Security Symp.*, San Francisco, CA, USA, 2011.
[6] X.Liu, Y.Zhang, B.Wang, and J.Yan, "Mona: Secure Multi-Owner Data Sharing For Dynamic Groups In The Cloud," In Proc. IEEE Transactions, June 2013