Attribute Based Encryption Schemes for
Efficient Verifiable Data Sharing

V. Agalya,
PG Scholar,
Department of computer science and Engineering
Dr. Sivanthi Aditanar college of Engineering
Tiruchendur, India

D. Kesavaraja,
Assistant Professor,
Department of computer science and Engineering
Dr. Sivanthi Aditanar college of Engineering
Tiruchendur, India

Abstract- Attribute-based encryption (ABE) with outsourced decryption not only enables fine-grained sharing of encrypted data, but also overcomes the effective drawback of the standard ABE schemes. An Attribute based encryption scheme with outsourced decryption allows a third party (e.g., a cloud server) to renovate an ABE cipher text into a (short) El Gamal-type cipher text using a public transformation key provided by a user so that the last can be decrypted much more efficiently than the former by the user. However, a shortcoming of the original outsourced ABE scheme is that the correctness of the cloud server's transformation cannot be verified by the user. That is, an end user could be cheated into accepting a wrong or maliciously transformed output. A security model of ABE with verifiable outsourced decryption is formalized by introducing a verification key in the output of the encryption algorithm. An approach to convert any ABE scheme with outsourced decryption into an ABE scheme with verifiable outsourced decryption is presented. The new approach is simple, general, and almost optimal. Compared with the original outsourced ABE, verifiable outsourced ABE neither increases the users and the cloud server's computation costs except some non-dominant operations (e.g., hash computations), nor expands the cipher text size except adding a hash value (which is <20 byte for 80-bit security level).

Index Terms— Attribute based encryption, data sharing, decryption outsourcing, verifiability.

I. INTRODUCTION
The enlarged use of computer and communication systems by industry has increased the risk of theft of proprietary information. Access controls to data operate on the assumption that data servers can be trusted to keep data confidential and enforce access control policies correctly. However, this assumption is no longer true today since services are increasingly storing data across many servers that are shared with other data owners. An example of this is cloud data storage where cloud service providers are not in the same trusted domains as end users, and hardware platforms are not under the direct control of data owners. To mitigate users’ privacy concerns about their data, a common solution is to store data in encrypted form so that it will remain private, even if data servers or storage devices are not trusted or compromised. The encrypted data, however, must be amenable to sharing and access control. Data encryption using symmetric or public key cryptography is not amenable to scalable access control. A promising approach to address this issue is attribute-based encryption (ABE). ABE schemes can be divided into two categories: Cipher text- Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE), depending on the access policy is embedded into the cipher text or the user’s private key. In CP-ABE, an access policy A is embedded in a cipher text CT and a user’s private key SK are associated with a set of attributes. The cipher text CT can be decrypted by SK if and only if \( f(A, S) = 1 \) for some predicate function \( f \), meaning that \( S \in A \). In KP-ABE, every cipher text is associated with a set of attributes, and every user’s private key is associated with an access policy on attributes. A user is able to decrypt a cipher text only if the set of attributes associated with the cipher text satisfies the access policy associated with the user’s private key. Both CP-ABE and KP-ABE can prevent any unauthorized users from accessing data, even if the user stores data in an untrusted server. Such properties of ABE schemes are very attractive in the area of cloud data storage. However, a drawback of the standard ABE schemes is their relatively large cipher text size and high decryption cost, and this problem is especially acute for resource limited devices such as mobile devices. In an ABE scheme, the size of the cipher text and the cost
of decryption grow with the complexity of the access structures/policies. Moreover, current constructions of ABE schemes rely on pairing-based groups and require many pairing operations in decryption. though exist ABE schemes with constant cipher text size and/or constant number of pairing operations in decryption, their access structures are restricted to AND gates or threshold gates which severely limit their practical applications. The user private key is split into a “transformation key” (denoted by TK), and an El Gamal-type secret key (denoted by DK). The transformation key can be publicly shared with a proxy, called Cipher text Transformation Server (CTS), while the secret key DK must be kept private by the user. ABE cipher texts are stored in a Cloud Storage Server (CSS). A cipher text CT stored in the CSS is first submitted to the CTS which use the key TK to transform CT into a simple and short El Gamal-type cipher text CT′ of the same message, instead of being decrypted by the user directly. From CT′, the user is able to recover the message using the secret key DK with just one exponentiation operation. The user’s transformation key can transform any ABE cipher text satisfied by user’s attributes, without revealing any information of the underlying message to malicious CTS.

II. PRELIMINARIES

A. ACCESS STRUCTURE

Let \{P1, P2, . . . , Pn\} be a set of parties. A collection A \notin \{P1, P2, . . . , Pn\} is monotone if \forall B, C: if B \in A and B \subseteq C then C \subseteq A. An access structure (respectively, monotone access structure) is a collection A of non empty subsets of \{P1, P2, . . . , Pn\}. The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets. The role of the parties is taken by attributes. Thus, the access structure A will contain the authorized sets of attributes. However, it is also possible to (inefficiently) realize general access by having the NOT of an attribute as a separate attribute.

B. RANDOM EXTRACTOR

Random extractor is a family of independent pairwise hash functions. These hash functions hold two important properties such that no two hash functions produce the same result for two different inputs and it must not be able to recover the original data from its hash value.

III. METHODOLOGY

A. SYMMETRIC ENCRYPTION

A symmetric encryption (SE) scheme with key space K consists of two probabilistic polynomial time algorithms: SE.Enc(K,m), mapping a key K \in K and a message m \in \{0,1\}∗ to a cipher text C, and SE.Dec(K, c) recovering m from C using K. The encryption of the symmetric encryption algorithm takes the message and the key as inputs and produces the cipher text as its output. It can also be said as the mapping of keys and the message. The decryption process considers the key and the cipher text and results in original message.

B. ATTRIBUTE BASED ENCRYPTION

Let A denote an access structure and S denote a set of attributes. For generality, we will define lenc and lkey as the inputs to the encryption algorithm and the key generation algorithm, respectively.
Concretely, we have \((I_{enc}, I_{key}) = (A, S)\) in a CP-ABE setting; \((S,A)\) in a KP-ABE setting. We also denote by \(f(I_{key}, I_{enc}) = 1\) the case \(S \in A\) and by \(f(I_{key}, I_{enc}) = 0\) otherwise.

1) Syntax of VO-ABE: An ABE system with privately verifiable outsourced decryption (shorted as VO-ABE system) consists of the following PPT algorithms:

- \((MPK, MSK) \leftarrow \text{Setup}(\kappa, U)\): The setup algorithm takes as input a security parameter \(\kappa\) and an attribute universe description \(U\). It outputs a master public key \(MPK\) (which defines a message space \(M\)) and a master secret key \(MSK\).
- \((CT_{I_{enc}}, VKM) \leftarrow \text{Encrypt}(MPK, M, I_{enc})\): The encryption algorithm takes as input \(MPK\), \(M\) and \(I_{enc}\). It outputs a cipher text \(CT_{I_{enc}}\) and a verification key \(VKM\).
- \((TK_{I_{key}}, DK_{I_{key}}) \leftarrow \text{KeyGen}(MSK, I_{key})\): The transformation key generation algorithm takes as input \(MSK\) and \(I_{key}\). It outputs a transformation key \(TK_{I_{key}}\) and a decryption key \(DK_{I_{key}}\).
- \(CT_{out} \leftarrow \text{Transform}(TK_{I_{key}}, CT_{I_{enc}})\): The cipher text transformation algorithm takes as input \(TK_{I_{key}}\) and \(CT_{I_{enc}}\). It outputs a partially decrypted cipher text \(CT_{out}\).
- \(M/ \perp \leftarrow \text{Decrypt}(DK_{I_{key}}, VKM, CT_{out})\): The message recovering algorithm takes as input \(VKM\), \(CT_{out}\) and \(DK_{I_{key}}\). It outputs a message \(M \not\in M \cup \{\perp\}\). Here the special symbol \(\perp\) indicates that the partially decrypted cipher text is invalid.

C. VERIFICATION

Verification in this construction is obtained through two consecutive hash functions. Hash function when applied produces the result to be of equal length. The result of hash function at both the sender and receiver must be the same or equal. If they do not match then it indicates the modification or deletion of the original message. Applying the hash functions is very efficient and cost effective.

IV. EXPERIMENTAL RESULTS

The implementation of the ABE schemes is done by using 224-bit MNT elliptic curve from the Stanford Pairing-Based Crypto (PBC) library. For \(\kappa = 80\) bits security parameter, and encapsulate a random 128-bit symmetric key \(\text{key}_{\text{sym}}\). In our scheme, we first hash the element \(CT\) of group \(GT\) to a random seed and then apply a pseudorandom number generator (e.g., AES scheme) to extend it to a 512-bit key \(K\).

The security analysis of both ECC and RSA reveals that ECC to be more secured than that of RSA as small number of bits can be recovered over a large number of years.

<table>
<thead>
<tr>
<th>TIME TO BREAK</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA (bits required)</td>
</tr>
<tr>
<td>(10^4)</td>
</tr>
<tr>
<td>(10^{11})</td>
</tr>
<tr>
<td>(10^{20})</td>
</tr>
<tr>
<td>(10^{40})</td>
</tr>
</tbody>
</table>

Fig 2. Comparative analysis of ECC and RSA

V. CONCLUSION

In this paper, a simple and generic method to convert any ABE scheme with non-verifiable outsourced decryption to an ABE scheme with verifiable outsourced decryption in the standard model is proposed. To concretely assess the performance of the new method, we presented an instantiation of our generic method based on Green et al.’s outsourced CP-ABE scheme without verifiability. We implemented our instantiation, verifiable outsourced scheme on PC. Experiment results showed that our method is nearly optimal in the sense that it introduces minimal overhead in exchange for verifiability.
REFERENCES