SECURING CLOUD DATA SHARING USING PAIRING-BASED IDENTITY BASED ENCRYPTION

Aishwarya Vidhya[1] | Rinesh S[2]
[1](Department of CSE, Karpagam University, Coimbatore, Tamil Nadu, INDIA, aishwaryavidhya89@gmail.com)
[2](Dept of Computer Science and Engg, Karpagam University, Coimbatore, Tamil Nadu, INDIA, rin.iimmba@gmail.com)

Abstract— *We describe the design of public key based protocols that allow authentication and key agreement between a data owner and a third party as well as between two users. The data owner is allowed to fully control the access policy associated with his/her data which is to be shared. Aside from the well- known vulnerabilities due to nature of cloud storage, it lack physical protection and are usually deployed in open for public use, which makes them vulnerable to be attacked. It is thus crucial to devise security solutions to this environment. The area of security and cryptography in cloud computing still has a number of open problems. On the other hand, the advent of Identity Based Encryption (IBE) has enabled a wide range of new cryptographic solutions. In this paper we describe that identity based encryption is ideal for securing data in cloud while sharing. Experimental results show that the proposed scheme well defend the threats during data sharing.*

Keywords—Attribute based Encryption, Identity based Encryption, Cloud, data sharing, Authentication

1. INTRODUCTION

In general, cloud users should be able to access data if they possess a certain set of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. Cipher text-policy attribute-based encryption (CP-ABE) is a very promising encryption technique for secure data sharing in the context of cloud computing. However, CP-ABE is limited to a potential security risk that is known as key escrow problem, whereby the secret keys of users have to be issued by a trusted key authority. Key escrow (also known as a "fair" cryptosystem) is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys. Key escrow systems provide a backup source for cryptographic keys. Escrow systems are somewhat risky because a third party is involved. Key escrow is a cryptographic key exchange process in which a key is held in escrow, or stored, by a third party. A key that is lost or compromised by its original user(s) may be used to decrypt encrypted material, allowing restoration of the original material to its unencrypted state. The basic definitions with respect to CP-ABE described in detail in [1][2] is as follows; Encrypt(PK,M, A). The encryption algorithm takes as input the public parameters PK, a message M, and an access structure A over the universe of attributes. The algorithm will encrypt M and produce a ciphertext CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains A. Key_ Generation(MK,S). The key generation algorithm takes as input the master key MK and a set of attributes S that describe the key. It outputs a private key SK. Decrypt(PK, CT, SK). The decryption algorithm takes as input the public parameters PK, a ciphertext CT, which contains an access policy A, and a private key SK, which is a private key for a set S of attributes. In CP-ABE, the cipher texts are identified with access structures and the private keys with attributes.

In this paper, we revisit attribute-based data sharing scheme in order to solve the key escrow issue but also improve the expressiveness of attribute, so that the resulting scheme is more friendly to cloud computing applications. We propose an improved two-party key issuing protocol that can guarantee that neither key authority nor cloud service provider can compromise the whole secret key of a user individually. Moreover, we introduce the concept of attribute with weight, provided to enhance the expression of attribute, which not only extend the expression from binary to arbitrary state, but also lighten the complexity of access policy. Therefore, both storage cost and encryption complexity for a cipher text are relieved.

2. RELATED WORKS

2.1 Attribute-based Encryption (ABE)

Attribute-based encryption (ABE) [3] is a relatively recent approach that reconsiders the concept of public-key cryptography. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver's public-key. Identity- based cryptography and in particular identity-based encryption (IBE) changed the traditional understanding of public-key cryptography by allowing the public-key to be an arbitrary string, e.g., the email address of the receiver. ABE goes one step further and defines the identity not atomic but as a set of attributes, e.g., roles, and messages can be encrypted with respect to subsets of attributes (key-policy ABE (KP-ABE)) or policies defined over a set of attributes (ciphertext-policy ABE – (CP-ABE)). The key issue is, that someone should only be able to decrypt a ciphertext if the

person holds a key for "matching attributes" (more below) where user keys are always issued by some trusted party.

2.2 Ciphertext-Policy ABE

In ciphertext-policy attribute-based encryption (CP-ABE) [5][6] a user's private-key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a ciphertext, if and only if his attributes satisfy the policy of the respective ciphertext. Policies may be defined over attributes using conjunctions, disjunctions and (k,n) - threshold gates, i.e., k out of n attributes have to be present (there may also be non-monotone access policies with additional negations and meanwhile there are also constructions for policies defined as arbitrary circuits). For instance, let us assume that the universe of attributes is defined to be {A,B,C,D} and user 1 receives a key to attributes {A,B} and user 2 to attribute {D}. If a ciphertext is encrypted with respect
to the policy (A∧ C)∨ D, then user 2 will be able to decrypt, while user 1 will not be able to decrypt.
CP-ABE thus allows to realize implicit authorization, i.e., authorization is included into the encrypted data and only people who satisfy the associated policy can decrypt data. Another nice feature is that users can obtain their private keys after data has been encrypted with respect to policies. So data can be encrypted without knowledge of the actual set of users that will be able to decrypt, but only specifying the policy which allows decrypting. Any future users that will be given a key with respect to attributes such that the policy can be satisfied will then be able to decrypt the data.

2.3 Key-Policy ABE

KP-ABE is the dual to CP-ABE in the sense that an access policy is encoded into the
users secret key, e.g., (A∧ C)∨ D(A∧ C)∨ D, and a ciphertext is computed with respect to a set of attributes, e.g., {A,B}{A,B}. In this example the user would not be able to decrypt the ciphertext but would for instance be able to decrypt a ciphertext with respect to {A,C}{A,C}.
An important property which has to be achieved by both, CP- and KP-ABE is called collusion resistance. This basically means that it should not be possible for distinct users to "pool" their secret keys such that they could together decrypt a ciphertext that neither of them could decrypt on their own (which is achieved by independently randomizing users' secret keys).

3. BASIC DEFINITIONS

3.1 Public key cryptosystem

In public key cryptosystem each user has a key pair (KU, KR), where KU is the public key and KR is the private key. To generate the key pair, one first chooses a private key KR and applies some one-way function to KR to obtain a random and uncontrollable KU. The main concern in a public key setting is the authenticity of the public key. If an attacker convinces a sender that a receiver's public key is some key of the attacker's choice instead of the correct public key, he can eavesdrop and decrypt messages intended for the receiver. This is the well-known man-in-the-middle attack. This authentication problem is typically resolved by the use of verifiable information called certificate, which is issued by a trusted third party consisting of the user name and his public key.

3.2 Identity based cryptography

The concept of identity based cryptography describes where the public key of a user can be derived from public information that uniquely identifies the user. For example, the public key of a user can be simply his/her email address or telephone number, and hence implicitly known to all other users. A major advantage of identity based cryptosystem is that no certificate is needed to bind user names with their public keys. The first complete identity based encryption scheme used a bilinear map (the Weil pairing) over elliptic curves to construct the encryption/decryption scheme. After that, the bilinear pairings have been used to design numerous identity based schemes, such as key exchange. In addition to the Weil pairing, there exists another bilinear map on the group of points on an elliptic curve, which is known as the Tate pairing. From a computational point of view, the Tate pairing can be done approximately twice as fast as the Weil pairing as it requires half the evaluations of rational functions in Weil pairing.
Identity based cryptosystem transparently provides security enhancement to the mobile applications without requiring the users to memorize extra public keys. For example, sending an identity based encrypted short message is exactly the same as sending a normal short message if the mobile phone number of the short message recipient is used as the public key. Therefore, the mobile user (the sender) does not need to memorize the public key of the receiver. This feature is especially desirable for mobile applications such as bank or stock transactions. However, in the existing identity based cryptosystem, the pairing computing has significant overhead. Therefore, efficient algorithm for identity based cryptosystem is essential in mobile devices with limited computing power.

3.3 Identity based Encryption

Identity Based Encryption (IBE) is an exception where known information that uniquely identifies users (e.g. IP or email address) [4][9]can be used as a public key and thus PKI is unnecessary. Although the notion of IBE, it only has become truly practical with the advent on Pairing Based Cryptography (PBC).
We first present some pairing concepts and then define the Tate pairing. In what follows, let E/Fq be an elliptic curve over a finite field Fq, E (Fq) be the group of points of this curve, and #E (Fq) be the group order.Bilinear pairing. Let n be a positive integer. Let G1 and G2 be additively-written groups of order n with identity 0, and let GT be a multiplicatively-written group of order n with identity 1. A bilinear pairing is a computable, non-degenerate function e:

G1 × G2 → GT. The most important property of pairings in cryptographic constructions is the bilinear, namely:

V P Є G1, VQ Є G2 and V a, b Є Z*, we have e ([a] P, [b] Q) = e (P, Q) ab.

Embedding degree. A subgroup G of E (Fq) is said to have an embedding degree k with respect to l if k is the smallest integer such that l | qk − 1.

Bilinear Diffie-Hellman Problem. Most of the PBC applications rely on the hardness of the following problem for their security: Given P, [a] P, [b] P, and [c] P for some a, b Є Z*, compute e (P, P) abc. This problem is known as the Bilinear Diffie-Hellman Problem [4]. The hardness of the Bilinear Diffie- Hellman Problem k depends on the hardness of the Diffie-Hellman problems both on E (Fq) and in Fq .So, for most PBC applications the parameters q, l, and k must satisfy the following

security requirements [2]:

1. l must be large enough so that solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) in an order n subgroup of E (Fq) is infeasible.
2. k must be large enough so that solving the Discrete Logarithm Problem (DLP) in

k

Fq is infeasible.

The Tate pairing. Let E (Fq) contain a subgroup of prime order l co-prime with q

and with embedding degree k. In most applications, l also is a large prime divisor of

#E (Fq). The Tate pairing is the bilinear pairing [11]:

$$\hat{e}: E(F_q^k)[l] \times E(F_q^k)/[l]E(F_q^k) \rightarrow F^*{}_q^k/(F^*{}_q^k)^l.$$

4. IBE ENABLED CLOUD DATA SHARING

Today, IBE seems to be the only truly practical mean of providing public key encryption in cloud environment. IBE would employ nodes identification (e.g., node IDs) as public keys and PKI's expensive operations would be thus unnecessary. IBE is not only ideal for cloud systems, but the converse is also true. For example, IBE schemes have strong requirements such as the existence of an unconditionally trusted entity that is responsible for issuing users private keys. Another requirement is that the keys must be delivered over confidential and authentic channels to users. In most of the cloud applications, however, nodes private keys can be distributed offline, i.e., they can be generated and preloaded directly into nodes prior to deployment.

In spite of all its advantages, IBE still is a public key cryptosystem and thus it is orders of magnitude more complex than symmetric cryptosystems. Because of this, IBE would only be used for setting up pair wise secret keys among cloud users.

4.1 Data Transfer Algorithm

The following steps required for communication from sensor node X to sensor node Y.

- X encrypts message using Y's identity.
- X signs encrypted message using its private key.
- X sends encrypted message and digital signature to Y.
- Y decrypts message using private key.
- Y verifies digital signature of message using X's public key.

4.2 Security Primitives

We give background on well-studied cryptographic primitive commonly used to achieve our security goals. Message authentication code. A common solution for achieving message authenticity and integrity is to use a message authentication code (MAC) [5]. A MAC can be viewed as a cryptographically secure checksum of a message. Computing a MAC requires authorized senders and receivers to share a secret key, and this key is part of the input to a MAC computation. The sender computes a MAC over the packet with the secret key and includes the MAC with the packet. A receiver sharing the same secret key recomputed the MAC and compares it with the received MAC value. If they are equal, the receiver accepts the packet and rejects it otherwise. MAC must be hard to forge without the secret key. This implies if an adversary alters a valid message or injects a bogus message, he/she cannot compute the corresponding MAC value, and authorized receivers will reject these messages.

4.3 Key Distribution Protocol

We show how IBE can be used to establish secret keys among data owner, third party and users. The protocol works as follows. Prior to deployment. Each node X is assigned the following information: the node's ID idX, the node's IBE private key SX, and a function Ø that takes an ID (e.g., idY) as input and outputs the corresponding IBE public key to the ID (e.g. PY).

After deployment.

1)Each node broadcasts its ID and a nonce.

2)Neighboring nodes thus use the function Ø together with the received ID to derive the corresponding public key. After that, neighboring nodes generate a secret key and respond to the original node by including this key in the message.

3)i) The transmission of the message is protected by using IBE's public and private keys. To prevent replay attacks, the nonce from the original node's broadcast in Step 1 is also included in the message. Finally, subsequent communications among nodes are protected with MAC computed using the secret keys.

ii) A value computed from the nonce (nonce') is also included as input to the MAC to prevent replay. The value of the "freshness token" nonce' needs to be updated in each interaction between nodes.

1. IDs being broadcast by nodes (e.g. A and B): A ⟹ GA : idA, nonce

B ⟹ GB : idB, nonce

2.Neighboring nodes (e.g., M from GA and N from GB) use received IDs to derive public keys (e.g. PA and PB) and distribute secret keys:
M → A: idA, encPA(idM | idA | kM,A | nonce) N → B: idB, encPB(idN | idB | kN,B | nonce)
3.Secure exchange of information between neighboring nodes (e.g., A and M, and
N and B)
A → M : idA, idM, m, mackM,A (idA | idM | m | nonce')
N → B : idN, idB, m, mackN,B (idN | idB | m | nonce')
Symbols used are:
idX : Node X's ID
GX : Group of nodes in node X's neighborhood kX,Y : Secret key shared between nodes X and Y PX : Node X's public key
SX : Node X's private key
mack() : MAC computed using key k enck() : Encryption computed using key k m : Message information
=> : Broadcast
→ : unicast

5. Implementation Issues

The pairing. The two most important pairings in Elliptic Curve Cryptography are the Tate and the Weil pairings. The Tate pairing seems to be more efficient than the Weil pairing [7][8]. Therefore, the Tate pairing appears to be more adequate to Cloud Environment than the Weil pairing.
The field. Given a cryptosystem, the hardness of its underlying problem dictates the size of the security parameters. Specifically, the harder the problem, the smaller the parameter size. The parameter size, in turn, dictates the efficiency, i.e., the smaller the parameter size, the faster the computation time. The DLP in prime fields is considered to be harder than the DLP in binary fields and thus it seems that prime fields are more adequate to cloud data sharing.
Curve selection. Super singular curves have been shown empirically to be faster than non-super singular curves. Super singular curves seem to be more adequate to cloud data sharing.
Parameters q and l. The choice of the parameters q and l is a key factor in the efficiency of pairing computation, as curve operations are performed using arithmetic of the underlying field. In prime fields, by choosing q a Mersenne prime (i.e., a number of the form 2p − 1) helps in computing modular reduction operations efficiently. However, it has been shown recently that such technique also decreases the hardness of the DLP in Fq and is potentially unsafe in the context of PBC. For l, on the other hand, it is possible to choose a Solinas prime, which decreases the number of point additions and makes the pairing computation faster.
Embedding degree k. We have chosen k = 2 since it provides a number of benefits while computing pairings. For example, k = 2 allows the denominator elimination optimization and makes Fqk arithmetic easier to implement.
Parameter sizes. Parameter sizes often pose a tradeoff between security level and efficiency. For most PBC schemes including IBE, the security requirements described in pairing concepts can be satisfied by choosing l > 2160 and qk > 21024. However, security requirements in cloud data management are often relaxed to meet their needs for efficiency. This is possible because of their short lifetimes and because the goal is not to protect each node individually, but the network operation as a whole. Until now, the larger parameters sizes for which the ECDLP and the DLP in prime fields are known to be solved are 2109 and 2448, respectively. Therefore, it seems that l >= 2128 and qk >= 2512 are able to meet the current security requirements of Cloud data sharing.

Point coordinates. The two most common coordinate systems are the projective system (x, y, z) and the affine system (x,y). The affine system requires inversions while performing point addition or doubling operations. The inverse operation, in turn, is commonly expensive. The projective system, on the other hand, reduces the need for inverse and thus seems to be more adequate to our target system.

6. Results

We describe the results of an implementation of the Tate pairing for resource constrained nodes. We use the following parameters:
a)The Tate Pairing on elliptic curves defined over fields with a large prime characteristic;
b)The embedding degree k = 2, q is a 256-bit prime, and l a 128-bit Solinas prime;
c)Group field arithmetic uses projective coordinates. To be concrete, we use the curve E/Fq : y2 = x3 + x with the parameters:
q =37781606889598235856745576472658394721481625071533302983957476142038207746163;
l = 170141188531071632644604909702696927 233;
h = 222060320700642449943812747791145685 108;

The average execution time to compute a pairing is 30.21s. The costs concerning RAM memory is 1,831 bytes and ROM memory is 18,384 bytes. Since we use IBE only to distribute secret keys among neighboring nodes, the costs are not a heavy burden to the whole system.
Table 1. Costs to evaluate the Tate pairing

Tate Pairing			
Time (seconds)	RAM (bytes)	ROM(bytes)	30.21
	1,831	18,384	

7. Conclusion

This paper proposed an efficient identity based cryptography scheme for end to end cloud data sharing among users, data owner and trusted third party with notable improvements. In future work, we will consider other efficient pairings. Our approach provides an

appropriate identity based encryption solution for cloud data services where the cloud users have limited computing power. Acknowledgments. Authors thank Karpagam University for supporting this research.

REFERENCES

[1] John Bethencourt, Amit Sahai, Brent Waters, Ciphertext-Policy Attribute-Based Encryption, University of Texas, USA. IEEE Symposium on Security and Privacy, 2007 (321-334).

[2] Brent Waters, Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization, International Workshop on Public Key Cryptography, 53-70, 2011

[3] Ragesh ,G. K., and Baskaran, K., Cryptographically Enforced Data Access Control in Personal Health Record System. Procedia Technology Journal- Elsevier, volume 25, (2014) 473-480.

[4] Veningston. K, Shanmugalakshmi. R, Nirmala. V. Semantic Association Ranking Schemes for Information Retrieval Applications using Term Association Graph Representation. Sadhana - Academy Proceedings in Engineering Science, Special issue on Machine Learning for Big Data, Vol. 40, Part 6, pp. 1793–1819.

[5] Ragesh ,G. K., and Baskaran, K., Privacy Preserving Cipher text Policy Attribute Set Based Encryption (PP-CP-ASBE) scheme for Patient Centric Data Access Control in Cloud Assisted WBANs, Advances in Computing, Communication and Information Science (ACCIS-14), Elsevier, (2014) 325-333.

[6] Ragesh ,G. K., and Baskaran, K., Attribute Set Based Encryption for User Centric Data Security and Privacy in Cloud-Assisted WBANs. INFORMATION-An International Interdisciplinary Journal, Volume 17, Issue No. 6 (A), (2014) 2207-2216.

[7] Ronald Watro, Derrick Kong, Sue-fen Cuti, Charles Gardiner, Charles Lynn and Peter Kruus. TinyPK: Securing Sensor Networks with Public Key Technology, Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN'04), ACM Press, October 2004.