Verifying Outsourced Decryption of CP-ABE with Signature

Hongwei Liu[1, a], Xueyuan Wang[1, b] and Peng Zhang[1, c]

[1]Faculty of Information Engineering, ShenZhen University, Shenzhen 518000, China.

[a]liuhw@szu.edu.cn, [b]1083275807@qq.com, [c]zhangp@szu.edu.cn

Keywords: Attribute-based encryption, Outsourced Decryption, Verification, Short Signature.

Abstract. Attribute-based encryption enables users to encrypt and decrypt messages based on user attributes. The computational complexities of decryption are very high. By Outsourcing decryption to the third party, it can address the problem. But the verifiability of results returned from the third party has yet to be addressed. In this paper, we introduce short signature and give a new model of verifiable outsourced decryption of ABE cipertexts. By using signature, we achieve the verification and provide good performance and propose a concrete scheme. We prove that our scheme is both secure and verifiable. Moreover, we show an implementation of our scheme and result of performance measurements.

1. Introduction

In 2005, Sahai and Waters [1] introduced the notion of attribute-based encryption (ABE). In 2007, Bethencourt et al. [2] introduced a varient of ABE: ciphertext policy ABE (CP-ABE). In a CP-ABE scheme, a ciphertext is associated with an access formula, and a user's private key is associated with attributes. The data owner can determine who can decrypt. But one of the main efficiency drawbacks of the existing CP-ABE schemes is that decryption is very expensive. Beacause the number of pairing operations grow with the complexity of the access formula.

Green et al. [3] proposed a solution to this problem by introducing the notion of ABE with outsourced decryption in 2011,which largely eliminates the decryption overhead for users. Green et al.also presented concrete CP-ABE schemes with outsourced decryption. The security property of the CP-ABE scheme with outsourced decryption guarantees that an adversary be not able to learn anything about the encrypted message. However, the scheme provides no guarantee on the correctness of the transformation done by the cloud server.

Junzuo et al. [4] study ABE with verifiable outsourced decryption in 2013. They first modify the original model of ABE with outsourced decryption to allow for verifiability of the transformations. After describing the formal definition of verifiability, they propose a new ABE model and based on this new model construct a concrete ABE scheme with verifiable outsourced decryption. But their scheme's performance is less than Green et al.

In 2014, Jin et al. [5] propose a new Secure Outsourced ABE system, which supports both secure outsourced key-issuing and decryption. Their method offloads all access policy and attribute related operations in the key-issuing process or decryption to a Key Generation Service Provider and a Decryption Service Provider, respectively, leaving only a constant number of simple operations for the attribute authority and eligible users to perform locally.But their scheme provide no guarantee that an adversary be not able to learn anything about the encrypted message.

1.1 Contribution.

Signature is an important means to achieve verification. The cost of signature is very small. In this paper, we give new methods for outsourcing decryption of ABE ciphertexts. We propose the first construction of CP-ABE associated with short signature [6, 7] to address the problem in the verifiable outsourced decryption of ABE cipertexts. In our scheme, encrypt algorithm produce a ciphertext which consist of the signature for this plaintext. The signature exists in the entire life cycle of the ciphertext. In addition, keygen algorithm produce a private key PK and a transform key TK. The TK can transform the ciphertext to simple ciphertext, and the PK is used to decrypt ciphertext. We show that the cost of signing is very low. As a result, we acheive both verification and good performance.

1.2 Organization.

The rest of the paper is organized as follows.In Section 2, we review some standard notations and cryptographic definitions. In Section 3, we describe our system model and security model of CP-ABE scheme. We describe our new model of CP-ABE with verifiable outsourced decryption in Section 4. We present security analysis in section 5. In section 6, we shows the experimental results on the performance of our proposed scheme. To the end, we state our conclusion in Section 7.

2. Background

2.1 Access Structure.

Definition1(access structure [8]). Let $\{P_1, P_2, ..., P_n\}$ be a set of parties. A collection $\mathbb{A} \in 2^{\{P_1, P_2, ..., P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (resp., monotone collection) of non-empty subsets of $\{P_1, P_2, ..., P_n\}$, i.e., $\mathbb{A} \in 2^{\{P_1, P_2, ..., P_n\}} \setminus \{0\}$.The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

Definition 2(access tree [2]). Let $\mathbb{T}$ be a tree representing an access structure. Each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value. If num_x is the number of children of a node *x* and k_x is its threshold value, then $0 < k_x \le num_x$. When $k_x = 1$, the threshold gate is an OR gate and when $k_x = num_x$, it is an AND gate. Each leaf node *x* of the tree is described by an attribute and a threshold value $k_x = 1$.

To facilitate working with the access trees, we define a few functions. We denote the parent of the node *x* in the tree by $parent(x)$. The function $att(x)$ is defined only if *x* is a leaf node and denotes the attribute associated with the leaf node *x* in the tree. The access tree $\mathbb{T}$ also defines an ordering between the children of every node, that is, the children of a node are numbered from 1 to *num*. The function $index(x)$ returns such a number associated with the node *x*. Where the index values are uniquely assigned to nodes in the access structure for a given key in an arbitrary manner.

Satisfying an access tree. Let $\mathbb{T}$ be an access tree with root *r*. Denote by $\mathbb{T}_x$ the subtree of $\mathbb{T}$ rooted at the node *x*. Hence $\mathbb{T}$ is the same as $\mathbb{T}_r$. If a set of attributes γ satisfies the access tree $\mathbb{T}_x$, we denote it as $\mathbb{T}_x(\gamma) = 1$. We compute $\mathbb{T}_x(\gamma)$ recursively as follows. If *x* is a non-leaf node, evaluate $\mathbb{T}_{x'}(\gamma)$ for all children x' of node *x*. $\mathbb{T}_x(\gamma)$ returns 1 if and only if at least k_x children return 1. If *x* is a leaf node, then $\mathbb{T}_x(\gamma)$ returns 1 if and only if $att(x) \in \gamma$.

2.2 Bilinear Maps.

Definition 3(bilinear maps [2]). Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order *p*. Let g_1 be a generator of $\mathbb{G}_1$, g_2 be a generator of $\mathbb{G}_2$ and *e* be a bilinear map, $e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. The bilinear map *e* has the following properties:

1. Bilinearity: for all $u \in \mathbb{G}_1, v \in \mathbb{G}_2$, $v \in \mathbb{G}_2$ and $a, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.

2. Non-degeneracy: $e(g_1, g_2) \neq 1$.

We say that $\mathbb{G}_1, \mathbb{G}_2$ is respectively a bilinear group if the group operation in $\mathbb{G}_1, \mathbb{G}_2$ and the bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ are both efficiently computable.

2.3 k-CAA.

Definition 4(K-CAA [9]). For an integer k, and $x \in \mathbb{Z}_p$, $g \in \mathbb{G}_1$, $h, h_1, h_2, ..., h_k \in \mathbb{Z}_p$, given $\{P, Q = g^x, g^{\frac{1}{h_1+x}}, ..., g^{\frac{1}{h_k+x}}\}$, to compute $g^{\frac{1}{h+x}}$ for some $h \notin \{h_1, ..., h_k\}$.

We say that the k-CAA is (t, ε) if for all t-time adversaries $\mathbb{A}$, we have Advk-CAA$_A$=

$$\Pr\left[\begin{array}{l}\mathbb{A}(g, Q = g^x, g^{\frac{1}{h_1+x}}, g^{\frac{1}{h_2+x}}, ..., g^{\frac{1}{h_k+x}}) = g^{\frac{1}{h+x}} \\ | \; x \in \mathbb{Z}_p, g \in \mathbb{G}_1\end{array}\right] < \varepsilon$$

2.4 Short Signature.

The short signature scheme is based on [10]. Let g_1, g_2 be a generator of G_1, G_2 with order *p*, the bilinear pairing is given by $e: G_1 \times G_2 \to G_T$. Define a cryptographic hash function $H: \{0,1\}^* \to \{0,1\}^\lambda$, where $|p| \geq \lambda \geq 160$. A signature scheme consists of the following four algorithms:a parameter generation algorithm ParamGen, a key generation algorithm KeyGen, a signature generation algorithm Sign and a signature verification algorithm Verify. We describe the signature scheme as follows:

1. *ParamGen.* The system parameters are $\{G_1, G_2, G_T, g_1, g_2, p, e, H\}$.

2. *KeyGen.* Randomly selects $x \in Z_p$, and computes $P_{pub} = g_1^x$. The public key is P_{pub}. The secret key is *x*.

3. *Sign.* Given a secret key *x*, and a message m, computes $S = g_2^{\frac{1}{H(m)+x}}$. The signature is S.

4. *Verify.* Given a public key P_{pub}, a message *m*, and a signature S, verify if

$$e(g_1^{H(m)} \cdot P_{pub}, S) = e(g_1, g_2).$$

The verification works because of the following equations:

$$\begin{aligned} e(g_1^{H(m)} \cdot P_{pub}, S) &= e(g_1^{H(m)+x}, g_2^{\frac{1}{H(m)+x}}) \\ &= e(g_1, g_2)^{(H(m)+x)\cdot(H(m)+x)^{-1}} \\ &= e(g_1, g_2) \end{aligned}$$

3. CP-ABE scheme and Security game

3.1 Ciphertext-policy ABE.

CP-ABE scheme consists of five fundamental algorithms: *Setup, Encrypt, KeyGen, Transform* and *Decrypt*.

Setup. The setup algorithm takes no input other than the implicit security parameter. It outputs the public key PK, a master key MK, a signing public key SPK and a signing secret key SSK.

Encrypt(PK,SSK,M,A). The encryption algorithm takes as input the public key PK and the signing secret key SSK, 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 and verify the message.Besides, CT implicitly contains the signature σ.

KeyGen(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 and a transform key TK.

Transform(TK,CT). The transform algorithm take as input TK and CT, and output the ElGamal ciphertext CT'. We specify our decryption procedure as a recursive algorithm.

Decrypt(SK,CT',SPK). The decrypt algorithm take as input SK, CT', SPK, and if the verification success, output message and if not, output $\perp$.

3.2 Security game.

Suppose the CP-ABE of [2] is a selectively CPA-secure CP-ABE scheme. Then the outsourcing scheme above is selectively CPA-secure in the random oracle model.

Setup. The challenger runs the Setup algorithm and gives the public parameters PK and sign public parameters SPK to the adversary.

Phase 1. The adversary makes repeated transformative keys and private keys corresponding to sets of attributes $s_1,...,s_{q_n}$. The TK and SK is given to the adversary.

Challenge. The adversary submits two equal length messages M_0 and M_1. In addition the adversary gives a challenge access structure A^* such that none of the sets $s_1,...,s_{q_n}$ from Phase 1 satisfy the access structure. The challenger flips a random coin b, and encrypts M_b under A^*. The ciphertext CT^* is given to the adversary.

Phase 2. Phase 1 is repeated with the restriction that none of sets of attributes $s_{q_{n+1}},...,s_q$ satisfy the access structure corresponding to the challenge. The TK and SK is given to the adversary.
Guess. The adversary outputs a guess b' of b.

The advantage of an adversary A in this game is defined as $Pr[b' = b]-1/2$. We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2.

4. A new CP-ABE with outsourcing and verification

4.1 Our construction.

Our CP-ABE construction is based on the construction of [2]. To enable outsourcing and verifing we modify the KeyGen algorithm to output a transformation key. Moreover, we define a new Transform algorithm, and modify the decryption algorithm to handle outputs of Encrypt as well as Transform. Now we describe our construction.

Setup(1^λ). The setup algorithm takes as input a security parameter λ.The security parameter is implict.The setup algorithm choose two bilinear group $\mathsf{G}_1,\mathsf{G}_2$ of prime order p with generator g_1, g_2. Next it will choose α, $\beta \in \mathsf{Z}_p$.The public key is PK=($g_1, h = g_1^{\beta}, e(g_1,g_2)^{\alpha}$) and the master key is MSK=(β, g_2^{α}). Then it will choose $x \in Z_p$, the signature public key is $SPK = (g_1^{x})$ and the signature private key is $SSK = (x)$.

Encrypt(PK,SSK,M,A). The encryption algorithm encrypts a message M under the tree access structure A. The algorithm first chooses a polynnomial q_n for each node n in the tree A. These polynomials are chosen in the following way in a topdown manner, starting from the root node R.For each node n in the tree, set the degree d_n of the polynomial q_n to be one less than the threshold value k_n of that node, that is, $d_n = k_n$-1.

Starting with the root node R the algorithm chooses a random $s \in \mathsf{Z}_p$ and sets $q_R(0) = s$. Then, it chooses d_R other points of the polynomial q_R randomly to define it completely. For any other node n, it sets $q_n(0) = q_{parent(n)}(index(n))$ and chooses d_n other points randomly to completely define q_n.

Let,J be the set of leaf nodes in T. The ciphertext is then constructed by giving the tree access structure T and computing

$$\text{CT}=(T, \tilde{C} = Me(g_1,g_2)^{\alpha s}, C = h^s; \forall j \in J : C_j = g_1^{q_j(0)}, C_j' = H(att(j))^{q_j(0)}).$$

Moreover, given the SSK = x, a message M, the signature is $\sigma = g_2^{\frac{1}{H(M)+x}}$.

KeyGen(MSK,S). The key generation algorithm will take as input a set of attributes *S* and output a key that identifies with that set. The algorithm first chooses a random $r, z \in \mathsf{Z}_P$, and then random $r_k \in \mathsf{Z}_p$ for each attribute $k \in S$. Then it computes the transformation key as

TK=($D = g_2^{(\alpha+r)/\beta z}; \forall k \in S : D_k = g_2^{r/z} \cdot H(k)^{r_k/z}, D_k' = g_2^{r_k/z}$).The SK is (TK,z).

Transform(TK,CT). The transform algorithm take as input TK and CT, and output the ElGamal ciphertext. We specify our decryption procedure as a recursive algorithm.

We first define a recursive algorithm DecryptNode(CT,TK, n) that takes as input a ciphertext CT, a transformation key TK which is associated with a set S of attributes, and a node n from T. If the node n is a leaf node then we let $i = att(n)$ and define as follows: If $i \in S$, then

$$\text{DecryptNode(CT, TK, } n\text{)}=\frac{e(D_i,C_n)}{e(D_i',C_n')} = e(g_1,g_2)^{rq_n(0)/z}.$$

If $i \notin S$,then we define DecryptNode(CT, TK, n)= $\perp$.

We now consider the recursive case when n is a nonleaf node. The algorithm

DecryptNode(CT,TK,*n*) then proceeds as follows: For all nodes e that are children of *n*, it calls DecryptNode(CT,TK,*e*) and stores the output as F_e. Let S_n be an arbitrary k_n-sized set of child nodes e such that $F_e \neq \perp$. We compute $F_n = e(g_1, g_2)^{rq_n(0)/z}$.

We define the transform algorithm. If the tree is satisfied by S, we set A=DecryptNode(CT, TK, *n*)= $e(g_1, g_2)^{rq_n(0)/z} = e(g_1, g_2)^{rs/z}$, compute $e(C, D)/A = e(g_1, g_2)^{\alpha s/z}$.

Finally, the transform algorithm output the partially decrypted ciphertext

$$CT' = (T_0, T_1) = (\tilde{C}, e(C, D)/A) = (Me(g_1, g_2)^{\alpha s}, e(g_1, g_2)^{\alpha s/z})$$

Decrypt(SK,CT',SPK). The decrypt algorithm take as SK, CT', SPK. Firstly, it use SK and CT' to compute

$$T_0 / T_1^z = M . \quad (1)$$

Next it compute $e(g_1^{H(M)} \cdot SPK, \sigma)$. If

$$e(g_1^{H(M)} \cdot SPK, \sigma) = e(g_1, g_2), \quad (2)$$

the decrypt algorithm output M. Otherwise, it output $\perp$.

4.2 Correctness.

The Decryption works because of the following equations:

Expand (1) as follows:

$$T_0 / T_1^z = \frac{\tilde{C}}{[e(C, D)/A]^z} = \frac{M \cdot e(g_1, g_2)^{\alpha s}}{[e(g_1^{\beta s}, g_2^{(r+\alpha)/\beta z}) / e(g_1, g_2)^{\alpha s/z}]^z} = M$$

The verification works because of the following equations:

Expand (2) as follows:

$$\begin{aligned} e(g_1^{H(M)} \cdot SPK, \sigma) &= e(g_1^{H(M)+x}, g_2^{\frac{1}{H(M)+x}}) \\ &= e(g_1, g_2)^{(H(M)+x)\cdot(H(M)+x)^{-1}} \\ &= e(g_1, g_2) \end{aligned}$$

5. Security

Theorem 5.1. Suppose the construction of [2] is a selectively CPA-secure CP-ABE scheme. Then the outsourcing scheme above is selectively RCCA-secure in the random oracle model for large message spaces.

Proof. Our outsourcing scheme is similar to [3]. Please refer to this proof in [3, Appendix A].

Theorem 5.2. If there exists a $(t, q_H, q_S, \varepsilon)$-forger F using adaptive chosen message attack for the used signature scheme, then there exists a (t', ε')-algorithm A solving q_S-CAA, where $t' = t, \varepsilon' \geq (\frac{q_S}{q_H})^{q_S} \cdot \varepsilon$.

Proof. Please refer to this proof in [10].

6. Performance

This section will give the performance of the proposed scheme.

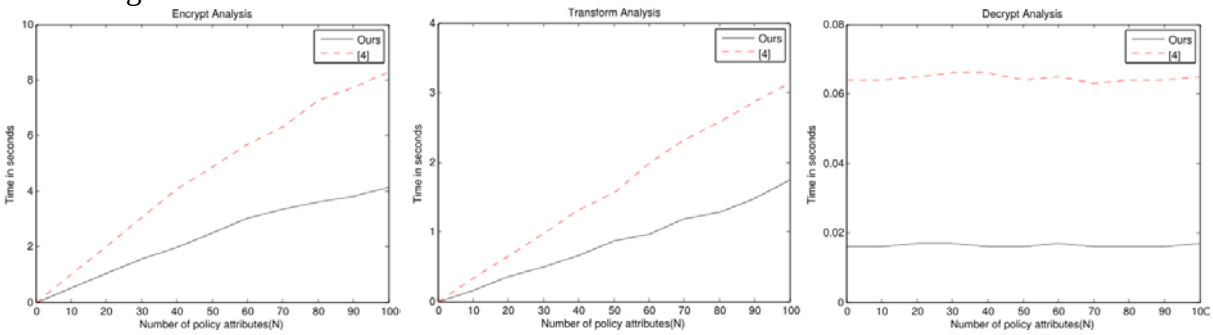
In order to evaluate the performance of our CP-ABE scheme with verifiable outsourced decryption, we compare with [4], as shown in Table 1. We denote *Pm* the point mutiplication on G_1, *Pa* the pairing operation, *n* the number of policy attributes, H the hash operation.

Table 1. Comparison of our scheme and the Junzuo scheme

System	Encrypt	Transform	Decrypt
Ours	*(n+1)Pm+nPa*	*nPa*	*Pa+Pm+H*
[4]	*2nPm+2nPa*	*2nPa*	*4Pm+2H*

We implement our scheme in software based on Stanford Java Pairing-Based Crypto library. All

implementitions run on an hardware platform: a 2.9GHz Intel Core i7-3520M CPU with 4GB of RAM running 64-bit Windows 8.



(a) (b) (c)

Fig. 1 Performance of our CP-ABE scheme with verifiable outsourced decryption

In a CP-ABE scheme, the complexity of ciphertext policy impacts both the encryption time and the decryption time. For each ciphertext policy, we repeat our experiment 100 times. In Fig. 1, we show the encryption time and the transformation time, the decryption time on ours and [4]. In Fig. 1 (a), we show that [4]'s encryption time is twice as ours. Beacause [4]'s Encrypt algorithm need to encrypt the message and a random message for verification. In Fig. 1 (b), we show that our scheme's transformation time is half of [4]. And we show that [4]'s decryption time is fourth of ours in Fig. 1 (c). In general, our scheme realize the outsourced decryption and verification, and guarantee system's performance.

7. Conclusion

In this paper, we give new methods for efficiently and securely outsourcing decryption of ABE ciphertexts. We provide the first construction of CP-ABE associated with Short Signature to address the problem in the verifiable outsourced decryption of ABE ciphertexts. We change the original ABE scheme with outsourced decryption proposed by Green et al. to include verifiability. We proved that it is secure and verifiable. To prove the performance of our scheme, we carry out it and perform experiments in a simulated outsourcing environment. As expected, the scheme largely reduced the computation time for decryption and realize the verifiable outsourced decryption.

Acknowledgments

This work is supported by the Science & Technology Innovation Projects of Shenzhen (JCYJ20140418095735596).

References

[1]. A. Saihai and B. Waters. Fuzzy identity-based encryption. In Proc. EUROCRYPT. 2005, p. 457–473.

[2]. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In IEEE Symposium on Security and Privacy, 2007, p. 321–334.

[3]. M. Green, S. Hohenberger, and B. Waters. Outsourcingthedecryption of ABE ciphertexts. In Proc. USENIX Security Symposium. San Francisco, CA, USA, 2011.

[4]. Junzuo Lai, Robert H. Deng, Chaowen Guan, and Jian Weng. Attribute-Based Encryption With Verifiable Outsourced Decryption. In IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. Vol. 8 (2013) , No. 8, p. 1343-1354.

[5]. Jin Li, Xinyi Huang, Jingwei Li, Xiaofeng Chen, and Yang Xiang. Securely Outsourcing Attribute-Based Encryption with Checkability. In IEEE TRANSACTIONS ON PARALLEL AND DISTTRIBUTED SYSTEMS. Vol. 25 (2014) , No. 8, p.2201-2210.

[6]. Dan Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In Proc. ASIACRYPT. 2001, p. 514-532.

[7]. Dan Boneh, Xavier Boyen. Short Signatures Without Random Oracles. In Proc. EUROCRYPT. 2004, p. 56-72.

[8]. A. Beimel. Secure Schemes for Secret Sharing and Key Distribution. PhD thesis, Israel Institute of Technology, Technion. Haifa, Israel, 1996.

[9]. S. Mitsunari, R. Sakai and M. Kasahara. A new traitor tracing. IEICE TRANSACTION. Vol. E85-A (2002) , No. 2, p.481-484.

[10]. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In Proc. PKC. Berlin, 2004, p. 277–290.