POSTER: Ciphertext-Policy Attribute-Based Encryption Method with Secure Decryption Key Generation and Outsourcing Decryption of ABE Ciphertexts

Yuejian Fang(✉), Zilong Wen, Qingni Shen, Yahui Yang, and Zhonghai Wu

School of Software & Microelectronics, Peking University, Beijing, China
{fangyj,qingnishen,yhyang}@ss.pku.edu.cn,
{zlwen,wuzh}@pku.edu.cn

Abstract. Attribute-based encryption (ABE) allows user to encrypt and decrypt data based on user attributes, and can be applied in some promising area such as mobile cloud storage. Since these are massive users in these applications, secure online transmission of decryption key is necessary. In this paper, a ciphertext-policy attribute-based encryption (CP-ABE) method with secure decryption key generation and outsourcing decryption of ABE ciphertexts is proposed. In the method, a user's public key information is embedded into his decryption key in the key generation algorithm. Both the user's decryption key and private key are needed to decrypt a ciphertext. With only the decryption key, a ciphertext cannot be decrypted, so the decryption key is secure and can be directly transmitted online. This saves some costs comparing to other transmission approaches, such as Secure Sockets Layer (SSL). Furthermore, the method supports outsourcing the decryption of ABE ciphertexts. Our analysis and experiment results prove that our method is more efficient than the existing outsourcing methods which generally use key transformation technique.

Keywords: CP-ABE · Secure decryption key generation · Outsourcing · Mobile cloud storage

1 Introduction

Attribute-based encryption provides a solution for a user to specify access control policy without prior knowledge of who will receive the use's messages. ABE can be applied in some new promising areas, such as mobile cloud storage [1]. Since these are massive users in these applications, secure online transmission of decryption key is necessary. An existing solution is to use SSL. There are much costs of SSL including setup, identification and key exchange, data encryption/decryption, etc.

In mobile cloud storage, since the size of ciphertext and the decryption time grow with the complexity of the access formula in ABE, the decryption process becomes a burden for mobile devices with limited computation ability. Some research works provide methods for outsourcing the decryption of ABE ciphertext [2-4]. The disadvantage of the existing outsourcing method is that the key transformation time grows linearly with the number of attributes, and this cost is not negligible for mobile devices in mobile cloud storage applications.

B. Thuraisingham et al. (Eds.): SecureComm 2015, LNICST 164, pp. 585–589, 2015.
DOI: 10.1007/978-3-319-28865-9_36

Our Contribution. In this paper, we propose a simple and efficient ciphertext-policy attribute-based encryption method with secure decryption key generation and outsourcing decryption of ABE ciphertexts.

In the set up algorithm of the method, each user establishes his public/private key. A user's public key information is embedded into his decryption key in the decryption key generation algorithm. In the decryption algorithm, both a user's decryption key and private key are needed to decrypt a ciphertext. With only the decryption key, ciphertext cannot be decrypted, so the decryption key can be directly transmitted online. It is secure from online attack, such as stealing by attackers. In existing secure transmission approaches, such as SSL, the user's key is regarded as structureless data bytes. Comparing to our method, SSL incurs extra costs, including extra costs data encryption/decryption.

The decryption algorithm of the method is divided into two stages. In the first stage, only the user's decryption key is used and a middle result is obtained. If the user decides to outsource the first stage computation to a third party, such as a cloud proxy, he sends his decryption key to the cloud. The cloud proxy gets the ciphertext and computes a middle result with the decryption key. The middle result is an ElGamal type ciphertext, and the cloud proxy can't further decrypt it. In the second stage, a user uses his private key, and uses part of his decryption key if needed, to get the final decrypted message. The advantage of our method is that the decryption key can be directly sent to the third party, while in the existing outsourcing methods, the user needs to use a secret key to turn his decryption key into a single transformation key, then he sends the transformation key to the third party. So our method is more efficient than the existing methods with outsourcing of ABE ciphertexts.

2 Our Construction

In this chapter, we give our new construction of CP-ABE algorithms that apply our method in research work [5]. The detailed description is given below.

Set up (λ) → GP. The setup algorithm takes as input a security parameter λ, it first generates $(q, \mathbb{G}, \mathbb{G}_T, e)$, where q is a λ-bit prime, $\mathbb{G}_0$ and $\mathbb{G}_1$ are two multiplicative cyclic groups with prime order q, Let g be a generator of $\mathbb{G}$ and e is the bilinear pairing $e: \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$. Next it chooses $H: \{0,1\}^* \to \mathbb{G}$.

The authority chooses random exponent $\alpha, \beta \in \mathbb{Z}_p$ as its master key($MK_A = \{\alpha, \beta\}$, his public parameters are: $PK_A = \{e(g,g)^{\alpha}, h = g^{\beta}, f = g^{1/\beta}\}$.

Each user j chooses a random exponent $\alpha_j \in \mathbb{Z}_p$ as his private key, and computes the corresponding public key $PubK_j$ as g^{α_j}.

Message Encryption(PK_A, M, $\mathbb{A}$). A user encrypts a message M under tree access structure $\mathcal{T}$ as follows:

The algorithm first chooses a tree access structure $\mathcal{T}$ the same as in [5], and then the cipher-text is then constructed by giving the tree access structure $\mathcal{T}$ as follows:

$$CT = (\mathcal{T}, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s, \forall y \in Y: C_y = g^{q_y(0)},$$

$$C'_y = H(att(y))^{q_y(0)})$$

KeyGen($MK_A, S, PubK_j$). Suppose user j with public key g^{α_j} holds a set of attribute S. The authority generates user j's decryption key. The algorithm chooses random $r \in \mathbb{Z}_p$, and then chooses random $r_k \in \mathbb{Z}_p$ for each attribute $k \in S$. Then it computes the decryption key as

$$SK_j = (D = g^{\frac{\alpha_j(\alpha+r)}{\beta}}, \forall k \in S: D_k = g^{\alpha_j r} \cdot H(k)^{r_k}, D_j' = g^{r_k})$$

FirstStageDecrypt (CT, SK_j). If the user decides to outsource the first stage computation to a third party, such as a cloud proxy, then the cloud proxy performs all the computation of this stage. The detailed process is described is as follows.

If the node x is a leaf node then we let $k = att(x)$, and define as follows: If $k \in S$, then

$$DecryptNode(CT, SK_j, x) = \frac{e(D_k, C_x)}{e(D_k', C_x')} = \frac{e(g^{\alpha_j r} \cdot H(k)^{r_k}, g^{q_x(0)})}{e(g^{r_k}, H(k)^{q_x(0)})}$$
$$= \frac{e(g^{\alpha_j r}, g^{q_x(0)}) e(H(k)^{r_k}, g^{q_x(0)})}{e(g^{r_k}, H(k)^{q_x(0)})} = e(g,g)^{\alpha_j r q_x(0)}$$

If $i \notin S$, then we define $DecryptNode(CT, DK_j, x) = \perp$.We now consider the recursive case when x is a non-leaf node. Then let S_x be an arbitrary k_x-sized set of child nodes z such that $F_z \neq \perp$. If no such set exists then the node was not satisfied and the function returns $\perp$.Otherwise, we compute

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S_x'}(0)}, i = index(x), S_x' = \{index(z): z \in S_x\}$$
$$= \prod_{z \in S_x} (e(g,g)^{\alpha_j r \cdot q_x(i)})^{\Delta_{i,S_x'}(0)} = e(g,g)^{\alpha_j r \cdot q_x(0)}$$

If the tree is satisfied by S, we set $CT_P = DecryptNode(CT, SK_j) = e(g,g)^{\alpha_j r \cdot q_R(0)} = e(g,g)^{\alpha_j r \cdot s}$。

SecondStageDecrypt ($\tilde{C}, CT_P, SK_j, PriK_j$). User j now decrypts $\tilde{C}$ with part of his decryption key SK_j and his private key $PriK_j$ by computing:

$$\frac{\tilde{C}}{(e(C,D)/CT_P^{\frac{1}{\alpha_j}})} = \frac{Me(g,g)^{\alpha s}}{(e(h^s, g^{\frac{\alpha_j(\alpha+r)}{\beta}})/(e(g,g)^{\alpha_j r \cdot s}))^{\frac{1}{\alpha_j}}} = M$$

The security proof of the construction is omitted for sake of space.

3 Performance

We implemented our new scheme of CP-ABE with access tree structure, and evaluated the performance. We made comparison with the existing method with key transformation technique [2] (shown with "-T" in the figure). We ran the tests on two hardware platforms: a 3.3 GHz Intel Core Duo platform with 4 GB RAM running Linux Kernel version 3.2.0, and a Google Nexus one mobile phone with 1 GHz Qualcomm Snapdragon (QSD) single core processor, 512 MB ARM running Android 2.3. We generated a collection of 100 distinct ciphertext policies of the form (A_1 AND A_2 AND … AND A_N), where A_i is an attribute. Each experiment was repeated vast times and averaged to obtain our decryption timings. The results are shown in Fig. 1.

Comparing the existing transmission approach such as SSL using HTTPS protocol, the decryption key generated in our method can be directly transmitted online using HTTP protocol. The key generation time of the method (SDKeyGen) with key transformation technique (KeyGen-T) need much more time. The main advantage of our method is that our method needs no key transformation, so the key generation time of our method is much smaller than the corresponding time of the method with key transformation technique.

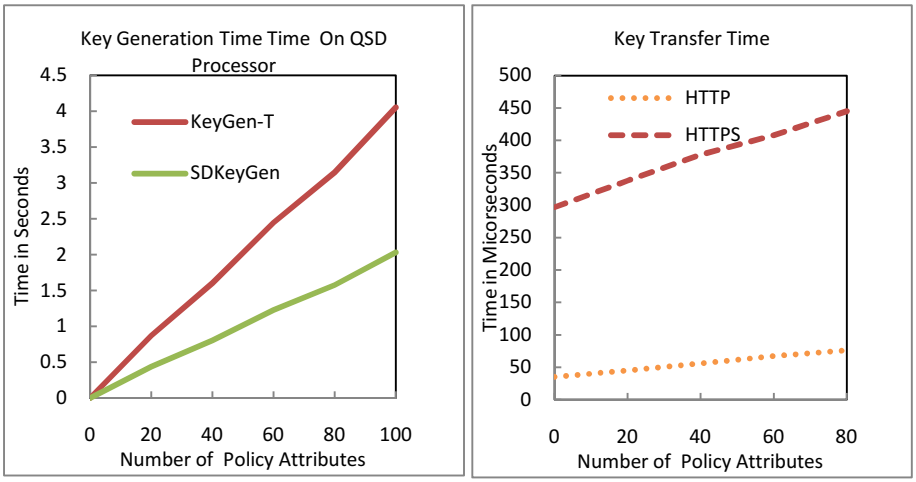

Fig. 1. Results with our CP-ABE scheme

4 Conclusion

In this paper, we propose a simple and efficient ciphertext-policy attribute-based encryption method with secure decryption key generation and outsourcing decryption of ABE ciphertexts. The analysis and experiments show that our method is more efficient than SSL and existing methods with outsourcing of ABE ciphertexts.

Acknowledgments. This work is supported by the National High Technology Research and Development Program ("863" Program) of China under Grant No. 2015AA016009, the National Natural Science Foundation of China under Grant No. 61232005, and the Science and Technology Program of Shen Zhen, China under Grant No. JSGG20140516162852628.

References

1. Zhou, Z., Huang, D.: Efficient and secure data storage operations for mobile cloud computing. In: Proceedings of the 8th International Conference on Network and Service Management, pp. 37–45 (2012)
2. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of ABE ciphertexts. In: Proceedings of the 20th USENIX Conference on Security, pp. 34–34 (2011)
3. Li, J., Huang, X., Li, J., Chen, X.: Securely Outsourcing Attribute-based Encryption with Checkability. IEEE Transactions on Parallel and Distributed Systems **25**(8), 2201–2220 (2014)
4. Lai, J., Deng, R.H., Guan, C., Weng, J.: Attribute-Based Encryption With Veriable Outsourced Decryption. IEEE Transactions on Information Forensics and Security **8**, 1343–1354 (2013)
5. Bethencourt, J., Sahai, A., Waters, B.: Cipheretxt-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)