Postprint: Research on Hierarchical File Attribute-Based Encryption with Hidden Access Structure

Authors: Shen Xueli, Lü Yingnan

Date: 2018-05-20T00:00:00+00:00

Abstract

Attribute-based encryption schemes based on file hierarchical structures are highly efficient and require low storage overhead in cloud storage environments. However, the access structure itself contains sensitive information, which creates risks of user information leakage and file theft. To address this problem, we propose a file hierarchical attribute-based encryption scheme that conceals the access structure. This scheme enhances the security of the encryption algorithm without compromising encryption and decryption efficiency, and employs a two-factor authentication mechanism to achieve more secure and efficient access control. The security of this research result is proven in the standard model under the Decisional Bilinear Diffie-Hellman assumption.

Full Text

Preamble

Title: Research on File Hierarchy Attribute Encryption with Hidden Access Structure

Authors: Shen Xueli , Lyu Yingnan
School of Electronics & Information Engineering; School of Graduate Studies, Liaoning Technical University, Huludao, Liaoning 125105, China

Abstract: Attribute-based encryption schemes based on file hierarchy are efficient and storage-saving in cloud storage environments. However, the access structure itself contains sensitive information, posing risks of user information leakage and file theft. To address this problem, this paper proposes a file hierarchy attribute encryption scheme with hidden access structure. This scheme improves the security of the encryption algorithm without affecting encryption and decryption efficiency, and adopts a two-factor authentication mechanism to achieve more secure and efficient access control. The research results are proven

secure under the standard model based on the Decisional Bilinear Diffie-Hellman (DBDH) assumption.

Keywords: cloud storage; access structure; file hierarchy; attribute encryption; two-factor authentication

0 Introduction

Cloud storage, built upon distributed computing technology, provides users with powerful sharing and storage capabilities in open network environments. However, traditional encryption techniques can no longer meet users' requirements for fine-grained access control [1]. Therefore, Waters et al. proposed a ciphertext-policy attribute-based encryption (CP-ABE) scheme [2] to achieve fine-grained access control. Many scholars have proposed CP-ABE schemes [3–7] that are costly and inefficient when encrypting files with hierarchical structures. To address this, Wang et al. [8] proposed a file hierarchy attribute-based encryption access control scheme (FH-CP-ABE), which solves multi-level file sharing problems through a hierarchical access structure model. Files are encrypted using an integrated access structure to reduce storage costs and computational complexity. However, since the access structure itself contains sensitive information, attacks on the access structure can easily lead to user information leakage and file theft risks.

In cloud storage environments, attribute-based encryption (ABE) schemes can achieve flexible user access control, where identity authentication is the first line of defense for access control and the foundation of cloud storage security. Two-factor authentication is a strengthened network access control mechanism that adds an additional security layer to the login process. In 1999, Yang et al. [9] first proposed a password authentication scheme using smart cards, which, unlike conventional password schemes, adopted a two-factor authentication strategy for the first time. Subsequently, scholars proposed numerous two-factor authentication schemes based on this foundation, such as Fan et al. [10] who proposed a robust remote authentication scheme with smart cards, and Das et al. [11] who proposed a dynamic ID-based remote user authentication scheme to solve security problems of static user IDs being attacked. In 2015, Wang et al. [12] proposed an anonymous two-factor authentication mechanism in distributed systems, which is both efficient and secure.

To solve the aforementioned problems, this paper draws on existing encryption models with hidden access policies [13–16] and proposes a hidden access structure file hierarchy attribute encryption scheme (HASFH-CP-ABE). By partially hiding the access structure, the scheme prevents sensitive information leakage from the access structure that could cause user information disclosure and file theft risks. Simultaneously, it adopts the two-factor authentication mechanism from [12] to achieve more secure and efficient access control. The scheme is proven secure under the standard model based on the DBDH assumption.

1.1 Bilinear Maps

Let G and G be two multiplicative cyclic groups of prime order p, and g be a generator of G . A bilinear map e: G × G → G satisfies the following properties:

(a) Bilinearity: For all u, v G and a, b , e(u , v) = e(u, v)

(b) Non-degeneracy: There exist u, v G such that e(u, v) 1

(c) Computability: For all u, v G , there exists an efficient algorithm to compute e(u, v)

1.2 Complexity Assumption: DBDH

Under the system security parameter, the challenger selects a group G of prime order p, where g is a generator of G. The Decisional Bilinear Diffie-Hellman (DBDH) problem is defined as: given (g, g , g , g , T), determine whether T = e(g, g) .

Given an algorithm B, if:

|Pr[B(g, g , g , g , e(g, g)) = 0] - Pr[B(g, g , g , g , e(g, g)) = 0]|

then B is said to solve the DBDH problem with advantage .

Definition 1: If no polynomial-time algorithm has non-negligible advantage in solving the DBDH problem, the DBDH assumption holds.

1.3 Access Structure

Let P = {p , p , ···, p } be a set of users. For a collection A 2 { }, if for any B, C: whenever B A and B C, then C A, then A is called monotone. Sets contained in the access structure A are called authorized sets, while sets not contained in A are called unauthorized sets.

1.4 Access Structure Tree

An access structure tree T represents an access structure. This paper assumes that files to be shared are divided into k access levels, thus M = {m , m , ··· , m }, where m represents the highest level and m the lowest level. Nodes in the access tree are denoted by (x, y). Leaf nodes represent attributes, while non-leaf nodes represent thresholds. For convenience, this paper provides the following definitions:

- num , : represents the number of child nodes of node (x, y)
- k , : represents the threshold value of the node. For leaf nodes, k , = 1; for non-leaf nodes: k , = 1 represents OR, k , = num , represents

AND

- parent(x, y): represents the parent node of node (x, y)
- index(x, y): represents the unique value arbitrarily assigned from 1 to num , for each child node of the access tree
- child(x, y): represents the child node set of node (x, y)
- att(x, y): represents the attribute associated with node (x, y)
- (x , y): represents level nodes, where each level node is the root of an integrated access tree, with (x , y) representing the highest level and levels decreasing from top to bottom

2 HASFH-CP-ABE Scheme Construction

Based on [8], this paper proposes an HASFH-CP-ABE encryption scheme using a two-factor authentication mechanism.

2.1 User Registration

The implementation of two-factor authentication requires two phases: registration and verification. For convenience, we define S as the remote server, U as the user, H (i = 1, 2, 3) as hash functions, as a secure channel, and → as a normal channel. x is the private key of S, and y = g mod p is the system public key.

Registration Phase:
User U selects a random string a and inputs identity ID and password PW . The user calculates M = H (H(ID) H(a || PW)) and sends {ID , PW , a, M } to authority S through a secure channel. S selects a random number b and calculates verification parameters based on user registration time t:

N = H (H(a || PW) H(x || ID || t))
S stores {ID , t, a, HoneyList = NULL} in the user database and stores {N , M , y, H (·), H (·), H (·)} in smart card SC.

Verification Phase:
When user U inserts the smart card into the card reader and inputs identity information ID ' and password information PW ' , SC performs the following operations:
SC selects a random number d and calculates:
Y = g mod p, Y = y mod p
K = H (x || ID || t) = H (H(a || PW) H(x || ID || t))
CID = H (ID ID ' H(Y || Y))
CMK = H (K b H(Y || Y))
SC sends {Y , Y , CID, CMK, M ' } to S through a normal channel, where M '

= H (H(ID ') H(a || PW ')). If M ' = M , S accepts and calculates:
CID' = H (ID ID ' H(Y || Y))
CMK' = H (K b H(Y || Y))
If CID = CID' and CMK = CMK' , the verification is complete. S and user U authenticate each other, and SK = Y mod p serves as the session key.

2.2 Encryption Algorithm

Define a bilinear map e: G × G → G , where G is a multiplicative cyclic group of prime order p, and g is a generator. Let A = {A , A , ···, A } represent the attribute set in the system, where each attribute A has m possible values, i.e., S = {t , t , ···, t } for 1 i n, 1 m.

Let L = {l , l , ···, l } be the user' s attribute list, where L S and A A. Define two hash functions H: {0, 1}* → G and G: {0, 1}* → {0, 1} .

Setup( ): System initialization takes security parameter as input. For any attribute value in the system, randomly select u . *Randomly select* , and compute Y = e(g, g) . The system public key is PK = {G , g, Y, {H = g } , }, and the master secret key is MSK = { , , {u } , }.

Encrypt(PK, M, T): For each node (x, y), select a polynomial q , . The polynomial selection rules are as follows: Starting from the root node R, for each node (x, y), the degree of polynomial q , is d , = k , - 1. The level coordinate of root node R is (x , y). Randomly select s * and set q (0) = s. The values of the polynomial at other d , points are randomly selected to fully define it. For other nodes in the access tree, set q , (0) = q , (index(x, y)), and randomly select the remaining d , points.

Let Y be the set of leaf nodes in the access structure tree. For any (x, y) Y, set i = att(x, y). The ciphertext is output as:

CT = {T, C' = M · Y , C'' = g , (x, y) Y: C , = H , C' , = g , }

2.3 Authentication and Private Key Generation

(1) User Login: The user inputs identity information ID ' and login password PW ' . If M ' = H (H(ID ') H(a || PW ')) holds, the smart card SC randomly selects d and calculates Y = g mod p, Y = y mod p. SC sends {Y , Y , CID, CMK, M ' } to the remote server S. S verifies whether M ' = M . If it holds, S randomly generates e and computes the temporary key K = H (x || ID || t). S sends {CID, CMK} to SC. SC calculates CID' = H (ID ID ' H(Y || Y)) and CMK' = H (K b H(Y || Y)). If CID' = CID and CMK' = CMK hold, the verification passes and the user private key generation algorithm is executed; otherwise, it returns .

(2) KeyGen(PK, MSK, L): The authorization center runs the key generation algorithm. For each attribute, it randomly selects r * and defines the private

key components. For each attribute value l L, randomly select r ' * and compute:

D = g^(/(+ r)) · H(l)^(r ')
D ' = g^(r ')
D '' = g^(r)

The private key is SK = {D , D ' , D '' } for all attributes.

2.4 Decryption Algorithm

(1) Verification Check: For access tree structure T, this paper achieves access structure hiding by erasing subtrees that do not contain level nodes under each level node. The erased access structure is represented by T. The system assigns an attribute component to each user' s attribute. During decryption, each level node (x , y) is the root of the original integrated access structure tree, and each level node corresponds to a level of ciphertext. The user substitutes their attribute components into the calculation to verify parameter values. If the verification fails, the system returns error identifier .

(2) Decrypt(CT, SK, (x, y)): If the verification passes, when node (x, y) is a leaf node, let i = att(x, y). If i S, then Decrypt(CT, SK, (x, y)) = . If i S, the decryption algorithm is defined as:

Decrypt(CT, SK, (x, y)) = e(D , C ,) / e(D ' , C' ,) = e(g, g)^(s/(+ r))

When (x, y) is a non-leaf node, let Q be the set of all child nodes of (x, y). For nodes in set Q, perform decryption operations and store the results in F , . Let S' , be an arbitrary threshold-sized subset of child node set Q. Calculate:

F , = _{z S' , } F_z^(Δ , ' , (0))

where Δ is the Lagrange coefficient. Based on the hierarchy of the access tree, if a user' s attribute set contains low-level authorization nodes, the value of each authorization node can be recursively calculated using:

F , = _{i S' , } (e(g, g)^(q (0)/(+ r)))^(Δ , ' , (0))

After obtaining F_R, use the following formula to get the authorized file:

M = C' / F_R

3.1 Resistance to Chosen Plaintext Attack

Theorem 1: If a probabilistic polynomial-time adversary B has no non-negligible advantage in winning the security game under selective plaintext attack, then the scheme is secure.

Proof: Construct a simulator B that can distinguish DBDH tuples from random tuples with advantage . Define a valid computable bilinear map e: G × G → G . Randomly select p, l, m, n *, and let g be a generator of G . The challenger defines an access structure tree T.

(1) **Initialization:** The attacker A selects an access structure A' and sends it to challenger B.

(2) **Setup:** Challenger B runs the initialization algorithm, randomly selects , *, and sends the public key PK to attacker A.

(3) **Phase 1:** Attacker A can adaptively query private keys for attribute sets S_A' that cannot satisfy access structure A' . For each attribute, B randomly selects r ' * and computes private key components.

(4) **Challenge:** Attacker A provides two equal-length ciphertext messages m and m . Challenger B randomly selects u {0, 1}, and under access structure A' , runs the encryption algorithm to generate challenge ciphertext CT.

(5) **Phase 2:** Repeat Phase 1.

(6) **Guess:** Attacker A outputs a guess u' for u. Challenger B outputs 1 if u' = u, otherwise 0.

The attacker' s advantage is defined as Adv(A) = |Pr[u' = u] - 1/2|. From the attacker' s perspective, the ciphertext component is completely random, so the probability of u' = u is exactly 1/2. Therefore, the advantage of B in the selective plaintext attack security game is . Under the DBDH assumption, the proposed data sharing scheme is proven secure.

3.2 Resistance to Collusion Attack

This scheme adopts a two-factor authentication mechanism where each user has a unique ID and login password PW. During decryption, user login is performed first, and the authentication system provides the first layer of judgment on user identity, increasing the difficulty for attackers to crack legitimate user credentials and impersonate authorized users. After passing identity verification, users substitute their attribute private key components into the decryption calculation to verify whether their private keys satisfy the access structure. In this scheme, users can only know whether they have the conditions to access secret files, but cannot obtain the specific attributes that satisfy the access structure. This effectively prevents collusion among multiple illegal or corrupted users to obtain decryption keys and shared files, or prevents low-authorization users from overstepping their authority to steal high-level encrypted files.

Experimental Results

The experimental code is verified based on the pbc-0.5.14 library [17] and the CP-ABE toolkit. The experimental results are shown in [Figure 1: see original paper] and [Figure 2: see original paper]. Figure 1 shows the relationship

between encryption/decryption time and the number of attributes when the number of shared files is fixed (k = 4). Figure 2 shows the relationship between encryption/decryption time and the number of files when the number of attributes is fixed (N = 30). The experimental simulation data values for varying attribute numbers and file numbers are N = {10, 15, 20, 25, 30, 35, 40} and k = {2, 4, 6, 8} respectively.

Experimental verification shows that when the number of shared files is fixed, as the number of attributes increases, this scheme is superior to the original scheme in encryption but less effective in decryption. When the number of attributes is fixed, as the number of files increases, the efficiency of this scheme in encryption and decryption is not significantly different from the original scheme. Therefore, the improved scheme enhances security without significantly impacting encryption and decryption efficiency.

5 Conclusion

Through experimental verification, this scheme addresses the risks of user information leakage and file theft without affecting encryption and decryption efficiency. Combined with the two-factor authentication mechanism, it achieves anonymous authentication of user identities, making the scheme more efficient and improving the security of the encryption algorithm. The research results are proven secure under the DBDH assumption.

References

[1] Khalil I M, Khreishah A, Azeem M. Cloud computing security: A survey [J]. IEEE Computers, 2014, 3(1): 1-35.
[2] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption [J]. IEEE Symposium on Security & Privacy, 2007, 2008(4): 321-334.
[3] Li J, Wang Q, Wang C, et al. Enhancing attribute-based encryption with attribute hierarchy [J]. Mobile Networks and Applications, 2011, 16(5): 553-561.
[4] Wang G J, Liu Q, Wu J, et al. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers [J]. Computers & Security, Advances in Network and System Security, 2011, 30(5): 320-331.
[5] Hur J. Improving security and efficiency in attribute-based data sharing [J]. IEEE Trans on Knowledge and Data Engineering, 2013, 25(10): 2271-2282.
[6] Green M, Hohenberger S, Waters B. Outsourcing the decryption of ABE ciphertexts [J]. Usenix Conference on Security, 2011, 49(3-4): 34-34.
[7] Lai L, Deng H R, Guan C, et al. Attribute-based encryption with verifiable outsourced decryption [J]. IEEE Trans on Information Forensics and Security, 2013, 8(8): 1343-1354.
[8] Wang Shulan, Zhou Junwei, Yu Jianping, et al. A novel file hierarchy access control scheme using attribute-based encryption [J]. Applied Mechanics and Materials, 2015, 701-702: 911-918.

[9] Yang W, Shieh S P. Password authentication schemes with smart cards [J]. Computers & Security, 1999, 18(8): 727-733.
[10] Fan C, Chan Y, Zhang Z. Robust remote authentication scheme with smart cards [J]. Elsevier Advanced Technology Publications, 2005, 24(8): 619-626.
[11] Das M, Saxena A, Gulati V. A dynamic ID-based remote user authentication scheme [J]. IEEE Trans on Consumer Electronics, 2004, 50(2): 629-631.
[12] Wang Ding, He Debiao, Wang Ping, et al. Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment [J]. IEEE Trans on Dependable & Secure Computing, 2015, 12(4): 428-442.
[13] Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures [M]. Berlin: Springer, 2008: 111-129.
[14] Xie L, Ren Y L. Efficient attribute-based encryption scheme with hidden access structure [J]. Journal of Xidian University, 2015, 42(3): 97-102.
[15] Song Y, Qin Z, Liu F M, et al. Policy-hidden attribute-based encryption scheme based on access tree [J]. Journal on Communications, 2015, 36(9): 119-126.
[16] Li X, Peng C G, Niu C C. Attribute-based encryption scheme with hidden tree access structure [J]. Journal of Cryptology, 2016, 3(5): 471-479.
[17] Lynn B. The pairing-based cryptography (PBC) library [EB/OL]. http://crypto.stanford.edu/pbc/.

Note: Figure translations are in progress. See original paper for figures.