ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201808.00072

A Traceable-Key Policy-Hiding Attribute-Based Encryption Scheme Postprint

Authors: Ou Yuyi, Liu Chunlong

Date: 2018-08-13T00:00:00+00:00

Abstract

Traditional attribute-based encryption schemes suffer from the issue where attributes contained in access policies may leak users' sensitive information, and malicious users who leak private keys to obtain illegal benefits cannot be held accountable. Meanwhile, private key length, ciphertext length, and decryption computational overhead all increase with the number of attributes, incurring substantial communication and computational overhead. To address these issues, we propose an attribute-based encryption scheme with traceability and hidden access structure. This scheme enhances the security of the encryption algorithm without affecting encryption/decryption efficiency, and employs a two-factor authentication mechanism to achieve more secure and efficient access control. Furthermore, a secure signature mechanism is introduced to support traceable keys for tracking malicious users. The scheme is based on the DBDH assumption and is proven secure in the standard model.

Full Text

Preamble

A Policy-Hidden Attribute-Based Encryption Scheme with Traceable Keys

Ou Yuyi, Liu Chunlong

(School of Computers, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Traditional attribute-based encryption schemes suffer from two critical issues: attributes contained in access policies may leak users' sensitive information, and malicious users can leak private keys for illegal profit without being held accountable. Additionally, private key length, ciphertext length,

and decryption computational overhead increase significantly with the number of attributes, incurring substantial communication and computational costs. To address these problems, this paper proposes an attribute-based encryption scheme that simultaneously achieves traceability and access structure hiding. The scheme enhances encryption algorithm security without affecting encryption/decryption efficiency, employs a two-factor authentication mechanism for more secure and efficient access control, and introduces a secure signature mechanism to support traceable keys for tracking malicious users. Based on the DBDH assumption, the scheme is proven secure in the standard model.

Keywords: attribute-based encryption; traceability; hidden policies; two-factor authentication

0 Introduction

Cloud storage, built upon distributed computing technology, provides powerful sharing and storage capabilities for users in open network environments. However, traditional encryption techniques can no longer meet users' demands for fine-grained access control [1]. To achieve fine-grained access control over encrypted data, Sahai et al. [1] first proposed a novel public-key encryption mechanism called Attribute-Based Encryption (ABE) in 2005, enabling one-to-many encryption in public-key cryptosystems. To represent more flexible access control policies, researchers subsequently proposed two categories of ABE mechanisms: Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [2]. In CP-ABE schemes, the encryptor encrypts messages using an access structure, and the decryptor obtains decryption keys from a trusted authority based on their attributes. If the decryptor' s attributes do not satisfy the access structure embedded in the ciphertext, they cannot decrypt the message.

However, in traditional CP-ABE schemes, the access policy and encrypted plaintext are often sent together to the decryptor, and the access structure may contain information related to the plaintext. Once the ciphertext is intercepted, the encryptor' s privacy may be compromised. To address this issue, numerous ciphertext-policy attribute-based encryption schemes with hidden access structures have been proposed. In 2007, Kapadia et al. [3] introduced a trusted third party and first proposed an anonymous ciphertext-policy attribute-based encryption scheme, but it could not resist collusion attacks. Subsequently, references [4, 5] implemented partially hidden policies in different forms, though [5] did not achieve key traceability. References [3, 6-8] all achieved fully hidden policies, but none provided key traceability, making it impossible to track users who maliciously leak information. Later, reference [9] proposed a highly expressive CP-ABE scheme with white-box traceability for malicious users, but it incurred significant performance overhead. Reference [10] presented a traceable scheme supporting large attribute universes. However, neither [9] nor [10] hid access policies. Reference [5] proposed a key-policy attribute-based encryption scheme that is traceable and partially hides attributes, dividing the attribute set into

three subsets: public normal attributes, hidden normal attributes, and hidden identity-related attributes, but the scheme could only hide partial attributes and suffered from long ciphertext and user public key lengths. In practical applications, excessively long ciphertexts incur substantial communication overhead. Reference [11] simultaneously achieved traceability and fully hidden policies, but its public key length increased.

In cloud storage environments, identity authentication serves as the first line of defense for access control and forms the foundation of security. Two-factor authentication is an enhanced network access control mechanism that adds an additional security layer to the login process. Therefore, this paper adopts the two-factor authentication mechanism from reference [8] to achieve more secure and efficient access control. Moreover, due to the characteristics of attribute-based encryption, user private keys are only associated with a set of attributes describing the user. In practice, legitimate users may maliciously leak their private keys to unauthorized users for profit without being held accountable. Therefore, adding identity authentication and designing encryption schemes that can track information-leaking users is practically significant.

This paper proposes an attribute-based encryption scheme with traceability and hidden access structures, building upon the system model of reference [7] while incorporating the signature mechanism from [11] and the two-factor authentication mechanism from [8]. The scheme first employs two-factor authentication to preliminarily verify user legitimacy, then embeds the access structure into ciphertexts using access trees converted from multi-value AND gates. Each attribute in the access structure can take multiple values, enhancing system flexibility, and the number of bilinear pairing operations is reduced to improve encryption and decryption efficiency. Based on the DBDH assumption, the scheme is proven secure in the standard model.

1.3 Security Model

The attacker model for user password authentication protocols has traditionally followed the classic Dolev-Yao model [18], where attackers can arbitrarily eavesdrop, intercept, insert, delete, or block messages flowing through public channels. In recent years, with the development of side-channel attack techniques (such as power analysis attacks, electromagnetic attacks, and timing attacks), attacker A can extract security parameters from smart cards, significantly enhancing attack capabilities. The capabilities of the attacker in this paper are summarized in Table 1 .

The proposed scheme achieves ciphertext indistinguishability under chosen plaintext attack in the selective attribute mode. The security model [19] is described through the following interactive game between attacker A and challenger B:

Initialization Phase: A provides B with two challenge access structures W and W . Challenger B selects a security parameter and runs the Setup algorithm to obtain public key PK and master secret key MSK. B sends PK to A

while retaining MSK.

Phase 1: Attacker A submits an attribute list L that does not satisfy either challenge access structure W or W . If this condition is met, challenger B runs the KeyGen algorithm and sends the private key SK to attacker A. A may make polynomially many such queries.

Challenge Phase: Attacker A submits two equal-length messages M and M to challenger B. B randomly selects a bit b {0,1} and encrypts M_b under W_b. B runs the encryption algorithm and returns the ciphertext CT to A.

Phase 2: Phase 1 is repeated, with A continuing to make private key queries for attribute lists that do not satisfy W or W .

Guess Phase: If A can correctly guess the value of b, the attacker wins the game; otherwise, A can only make a random guess. The advantage of attacker A in winning this game is defined as $Adv(A) = |Pr[b^* = b] - 1/2|$. If no polynomial-time attacker can win this game with non-negligible advantage, the scheme is considered secure.

1.4 Scheme Definition

The scheme consists of the following algorithms: Registration, Verification, Setup, KeyGen, Encrypt, Decrypt, and Trace. The scheme employs a two-factor authentication mechanism where each user has a unique ID and login password PW. Before decryption, users must log in, and the verification system performs the first layer of identity verification, increasing the difficulty for attackers to crack legitimate user credentials and impersonate authorized users. After successful authentication, users substitute their attribute private key components into the decryption calculation to verify whether their private key satisfies the access structure. If the access structure is satisfied, the ciphertext is decrypted.

- **Registration:** Registers users. The user inputs identity ID and password PW. The authority selects system parameters c, b, and the current registration time t to perform calculations and generate credentials stored in the smart card SC.
- **Verification:** Authenticates users. The user logs in with identity ID and password PW. Based on the verification algorithm, the system performs the first layer of identity verification to check legitimacy.
- **Setup:** Initialization algorithm performed by the authority. Takes security parameter as input, outputs master secret key MSK and public parameters PK, and generates a query list T containing (k, ID) pairs. Initially, T is empty.
- **KeyGen:** Key generation algorithm. Takes MSK, attribute list L, PK, and user identity ID as input, outputs private key SK for attribute list L, and stores the (k, ID) pair in query list T.

- **Encrypt:** Data encryption algorithm performed by the sender. Takes plaintext message M, access policy W, and PK as input, outputs ciphertext CT.
- **Decrypt:** Decryption algorithm. The receiver inputs PK, ciphertext CT with implicitly embedded access structure, and user private key SK containing attribute list L. If the attributes in the decryption key satisfy the access structure in the ciphertext, the message M is correctly decrypted; otherwise, it outputs .
- **Trace:** Trace algorithm. Takes user public key PK, private key SK, and query list T as input. First verifies whether SK is properly formed according to the standard definition. If verification succeeds, it outputs the user identity ID corresponding to k from the query list T; otherwise, it outputs the symbol , indicating that the user private key does not need to be traced.

2.1 User Registration

Two-factor authentication is implemented in two phases: registration and verification. For clarity, we define: S as the remote server, U as the user, and H() as a hash function. The authority S has private key x. The user inputs identity ID and password PW, and the client SC selects a random number c. The system performs hash computations on the credentials and stores the resulting parameters in the smart card SC. The authority S selects a random number b and uses the user registration time t to compute additional authentication parameters, which are also stored in SC.

2.2 Identity Verification

During verification, the user logs in by inputting identity ID* and password PW*. The smart card SC performs computations using the stored parameters and generates authentication messages. The authority S receives these messages and performs mutual authentication by verifying the correctness of the computed values. If verification succeeds, the user is authenticated and can proceed to the encryption algorithm; otherwise, the request is rejected. The protocol employs Diffie-Hellman key exchange techniques with ephemeral values to ensure forward security and protect user anonymity.

2.3 Encryption Algorithm

The encryption algorithm takes as input the system public key PK, plaintext M, and access structure W. The encryptor first converts the access structure expressed as multi-value AND gates into a corresponding access tree according to transformation rules. The encryptor selects a random value s and then selects random s_i for each child node i of the access tree. The ciphertext components are computed as:

$$C_0 = g^s$$

$$C_1 = e(g,g)^{ys} \cdot M$$

$$C_{i,j} = A_{i,j}^{s_i}$$

The final ciphertext is output as CT = {C , C , {C_{i,j}}}.

2.4 Decryption Algorithm

The decryption algorithm takes as input the system public key PK, ciphertext CT with implicitly embedded access structure, and user private key SK containing attribute list L. The decryptor computes:

$$M = \frac{C_1}{\prod e(C_0, D_{i,j})}$$

If the attribute list L satisfies the access structure W, the plaintext M is successfully recovered; otherwise, the algorithm outputs .

2.5 Trace Algorithm

The trace algorithm consists of verification and query phases, taking as input the user public key PK, private key SK, and query list T.

Verification Phase: The algorithm verifies whether SK is a properly formed key by checking the following equations:

$$\frac{e(k,g)}{e(g,g)^y} \neq 1$$

$$\prod e(A_{i,j}, D_{i,j}) = e(g,g)^{\Sigma}$$

If these verification equations hold, SK is a valid user private key.

Query Phase: If SK passes verification, the algorithm queries the value k in list T and outputs the corresponding user identity ID. If verification fails, it outputs the symbol .

3.1.1 Two-Factor Authentication Security Analysis

The scheme addresses the attacker capabilities outlined in Table 1 as follows:

C-00 Capability: The scheme employs public-key technology. Even if an attacker obtains a user' s ID and PW through offline exhaustive search, they cannot access encrypted files without the user' s attribute private key.

C-01 Capability: User anonymity is protected by hashing the user ID, preventing attackers from obtaining the user' s identity identifier.

C-1 Capability: The scheme combines two Diffie-Hellman key exchange operations with public-key cryptography. Even if attackers can extract parameters from smart cards and obtain public keys, they cannot break the untraceability of other users without the private key. Existing research has theoretically proven that public-key technology is necessary to achieve multi-factor security, user anonymity, resistance to offline password guessing attacks, and forward security [20].

C-3 Capability: Forward security is achieved through traditional Diffie-Hellman key exchange, where each verification uses different ephemeral values g^x and g^y, preventing attackers from using expired session keys to compromise future sessions.

3.1.2 Resistance to Chosen Plaintext Attack

Theorem 1. If no probabilistic polynomial-time attacker A can win the security game with non-negligible advantage, the proposed scheme is secure against chosen plaintext attacks.

Proof. If attacker A can break the scheme with non-negligible advantage , then there exists a challenger B that can solve the DBDH assumption with the same advantage . The challenger is given a DBDH tuple (g, g^a, g^b, g^c, Z), where Z is either e(g,g)^{abc} or a random element in G_T with equal probability.

Initialization Phase: A provides B with two challenge access structures W and W . B flips a coin to select b {0,1} and sets y = ab. B randomly selects parameters and sends the public key PK to A while keeping the master key MK.

Phase 1: Attacker A provides an attribute list L that does not satisfy W or W . For each attribute att_i L, if att_i W_b, B selects random r and computes D_{i,j} = g^{r}. Otherwise, B computes D_{i,j} = g^{r} · g^{ab}. B sends SK to A.

Challenge Phase: A submits two equal-length messages M and M . B randomly selects s and computes the challenge ciphertext CT. If Z = e(g,g)^{abc}, CT is a valid ciphertext; otherwise, CT is independent of the messages.

Phase 2: Phase 1 is repeated with continued private key queries.

Guess Phase: If A correctly guesses b, B outputs that Z = e(g,g)^{abc}; otherwise, B outputs that Z is random. Thus, B can solve the DBDH assumption with advantage /2, proving the scheme secure.

3.1.3 Resistance to User Collusion Attack

The scheme employs a two-factor authentication mechanism where each user has a unique ID and password. Users must authenticate before uploading encrypted data or downloading decrypted files, increasing the difficulty for attackers to impersonate legitimate users. After authentication, users substitute their attribute private key components into the decryption calculation to verify whether their private key satisfies the access structure. Users can only know whether they have access to secret files but cannot learn the specific attribute expressions that satisfy the access structure. This effectively prevents multiple illegal or corrupted users from colluding to combine attribute private keys and obtain shared files, or from low-privilege users illegally accessing high-level encrypted files.

3.1.4 White-Box Traceability

Before uploading data to the cloud server, the data owner first authenticates their identity. After successful authentication, they encrypt using access policy W, and only authorized users can decrypt the plaintext with the help of the decryption server. Users must also authenticate before decryption. During private key generation, a factor k is embedded in SK, and the (k, ID) pair is stored in query list T. To decrypt ciphertext CT, an attacker must possess the system public key PK, ciphertext CT, and private key SK, and the attribute set must satisfy the access control policy. The data provider can verify whether SK is properly formed using PK and SK. If verification succeeds, the user identity ID corresponding to factor k is obtained by querying list T; otherwise, the symbol is output.

3.2 Efficiency Analysis

The efficiency of attribute-based encryption mechanisms primarily concerns communication and computational overhead. Ciphertext length determines communication overhead, and reducing it can decrease communication costs. Since ABE uses one-to-many communication, decryption is a high-frequency operation compared to encryption, and bilinear pairing computations are less efficient than other operations. Reducing the number of bilinear pairing operations during decryption can effectively lower computational overhead. We compare our scheme with existing schemes in terms of encryption/decryption time, functional features, ciphertext length, and private key length. Let n be the number of attributes in the system, n_i be the number of possible values for the i-th attribute, N = Σ_{i=1}^n n_i be the total number of attribute values, |G|, |G_T|, and |Z_p| be the element sizes in groups G, G_T, and Z_p respec-

tively, T_G and T_{G_T} be computation times in G and G_T, and T_e be the time for bilinear pairing operations.

Table 2 compares functional features, showing that references [4-8] only achieve policy hiding, while [9,10] only achieve key traceability. Reference [5] achieves both but only partially hides policies. Reference [11] achieves both full policy hiding and traceability but has longer public keys. Our scheme achieves both traceability and policy hiding while using two-factor authentication to effectively resist user collusion attacks.

Table 3 compares parameter sizes. Our scheme has the same system public key and master key lengths as reference [7] but shorter than reference [11]. The user private key length matches reference [7] and is shorter than reference [11]. The ciphertext length is shorter than both [11] and [7].

Table 4 compares time overhead. Our encryption time matches reference [11], but decryption time is significantly shorter than [11] and shorter than [7] because the number of bilinear pairing operations is reduced by nearly half, thereby improving efficiency.

4 Conclusion

This paper proposes a ciphertext-policy attribute-based encryption scheme with traceable keys and hidden access structures. Without affecting encryption/decryption efficiency, the scheme combines two-factor authentication to achieve anonymous user authentication, addressing the risks of user sensitive information leakage and file theft. We prove that under the DBDH assumption, the scheme can resist chosen plaintext attacks and achieve ciphertext indistinguishability. While ensuring data security, the scheme significantly reduces communication and computational overhead by decreasing the number of bilinear pairing operations. The limitation of this scheme is that it only supports white-box traceability; therefore, future work will focus on achieving black-box traceability.

References

[1] Sahai A, Waters B. Fuzzy identity-based encryption [C]// Proc of International Conference on Theory and Applications of Cryptographic Techniques. [S. l.]: Springer-Verlag, 2005: 457-473.

[2] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data [C]// Proc of ACM Conference on Computer and Communications Security. [S. l.]: ACM Press, 2006: 89-98.

[3] Kapadia A, Tsang P P, Smith S W. Attribute-based publishing with hidden credentials and hidden policies [C]// Proc of Network and Distributed System Security Symposium. San Diego, California: [s. n.], 2007.

[4] Ying Zuobin, Ma Jianfeng, Cui Jiangtao. Partially policy hidden CP-ABE supporting dynamic policy updating [J]. Journal on Communications, 2015, 36(12): 178-189.

[5] Yu Shucheng, Ren Kui, Lou Wenjing, et al. Defending against key abuse attacks in KP-ABE enabled broadcast systems [M]// Security and Privacy in Communication Networks. 2009: 311-329.

[6] Liu Xueyan, Zheng Dengfeng. CP-ABE scheme based on prime group with fully hidden access structure [J]. Computer Engineering, 2016, 42(10): 140-145.

[7] Wang Haiping, Zhao Jingjing. Ciphertext-policy attribute-based encryption with anonymous access structure [J]. Computer Science, 2016, 43(2): 175-178.

[8] Shen Xueli, Lu Yingnan. Research on file hierarchy attribute encryption of hidden access structure [J/OL]. Application Research of Computers, 2019, 36(1): 1-2 [2018-05-17]. http://kns.cnki.net/kcms/detail/51.1196.TP.20180208.1714.084.html.

[9] Yadav U C. Ciphertext-policy attribute-based encryption with hiding access structure [C]// Advance Computing Conference. [S. l.]: IEEE Press, 2015: 6-10.

[10] Ning Jianting, Cao Zhenfu, Dong Xiaolei, et al. Large universe ciphertext-policy attribute-based encryption with white-box traceability [J]. 2014, 8713: 55-72.

[11] Wang Mei, Sun Lei. A secure and traceable attribute-based encryption scheme with hiding access structure [J]. Computer Applications and Software, 2017(2): 267-271.

[12] Ning Jianting, Cao Zhenfu, Dong Xiaolei, et al. White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes [J]. IEEE Trans on Information Forensics & Security, 2015, 10(6): 2015.

[13] Zhong Hong, Zhu Wenlong, Xu Yan, et al. Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage [J]. Soft Computing, 2016, 22(1): 1-9.

[14] Odelu V, Das A K, Rao Y S, et al. Pairing-based CP-ABE with constant-size ciphertexts and secret keys for cloud environment [J]. Computer Standards & Interfaces, 2017, 54(P1): 3-9.

[15] Qin Baodong, Deng Robert H, Liu Shengli, et al. Attribute-based encryption with efficient verifiable outsourced decryption [J]. IEEE Trans on Information Forensics and Security, 2015, 10(7): 1384-1393.

[16] Liu Zhen, Cao Zhenfu, Wong D S. White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures [J]. IEEE Trans on Information Forensics & Security, 2013, 8(1): 76-88.

[17] Chen Lu, Wang Wei. An access control scheme based on ATP-ABE [J/OL]. Computer Engineering and Applications, 1-9 [2018-05-17]. http://kns.cnki.net/kcms/detail/11.2127.TP.20180409.1448.016.html.

[18] Dolev D, Yao A. On the security of public key protocols [J]. IEEE Trans on Information Theory, 1983, 29(2): 198-208.

[19] Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures [M]// Applied cryptography and network security. Berlin: Springer, 2008: 111-129.

[20] Wang Ding, Li Wenting, Wang Ping. Crytanalysis of three anonymous authentication schemes for multi-server environment [J/OL]. Journal of Software, 1-16 [2018-06-25]. https://doi.org/10.13328/j.cnki.jos.005361.

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

Source: ChinaXiv —Machine translation. Verify with original.