Postprint: Revocable Location-Based Hierarchical Attribute-Based Encryption

Authors: Shen Xueli, Cui Haiyun, Chen Xintong

Date: 2018-11-29T00:00:00+00:00

Abstract

Attribute-based encryption-based location hierarchical access schemes enable users to flexibly configure their location access information according to their own circumstances, which not only addresses the location sharing problem in social networks but also enhances decryption efficiency through algorithmic improvements. However, during system operation, there exists the need for users to correct their attribute information or the possibility that partial private keys may be compromised, making revocation support essential for system security. Based on this consideration, we propose a revocation-supporting location hierarchical attribute-based encryption scheme that outsources partial decryption operations to a decryption server and incorporates a two-factor authentication method. This scheme reduces users' computational overhead while enhancing algorithmic security.

Full Text

Preamble

Research on Location-Hierarchical Attribute-Based Encryption Supporting Revocation

Shen Xueli, Cui Haiyun, Chen Xintong

School of Electronics & Information Engineering, Liaoning Technical University, Huludao, Liaoning 125105, China

Abstract: The location-hierarchical access control scheme based on attribute-based encryption allows users to flexibly configure their location access information according to their own circumstances. It not only solves the location sharing problem in social networks but also improves decryption efficiency through algorithmic enhancements. However, during system operation, users may need to correct their attribute information, and there exists the possibility that some

private keys could be leaked, making revocation support essential for system security. This paper proposes a location-hierarchical attribute-based encryption scheme supporting revocation that outsources partial decryption operations to a decryption server and incorporates a two-factor identity authentication method. The scheme reduces user computational costs while enhancing algorithmic security.

Keywords: attribute-based encryption; revocation; hierarchical access control; outsourced decryption; two-factor authentication

0 Introduction

With the rapid development of the Internet, cloud storage has become a hot research topic in the field of information technology. In recent years, various location-based services have proliferated. In distributed open network environments, users in social networks can enjoy diverse location services such as vehicle navigation and hotel search services. However, while enjoying the multifunctionality of these services, users have also noticed security issues in social software. To improve the security of their location information, relevant location protection methods have received increasing attention.

Current methods for protecting personal location information can be broadly divided into two categories: spatial cloaking techniques [**?**] and dummy location techniques [**?**]. Both methods protect information by obfuscating user location data or generating false information to confuse attackers. However, they do not allow users to perform fine-grained access control, cannot provide the most concise location information according to specific user needs, and can easily lead to information leakage.

To address these issues, Lin et al. [**?**] proposed a new attribute-based encryption algorithm based on traditional attribute-based encryption schemes [**?**] and designed a location-hierarchical access control scheme by drawing on linear secret sharing schemes from references [**?**, **?**]. This hybrid approach uses both attribute-based encryption and symmetric encryption to encrypt user location information, allowing users to set location information access policies according to their needs. However, user permissions change as users modify their attributes, and the scheme proposed by Lin et al. cannot achieve timely attribute revocation. Pirretti et al. [**?**] proposed an attribute revocation scheme for ciphertext-policy attribute-based encryption in 2006, where the central authority periodically updates attribute versions with set validity periods, thereby revoking specified attributes to achieve user revocation. Hur et al. [**?**] proposed an attribute-based encryption scheme supporting revocation in 2011, which can achieve fine-grained access control in outsourced environments but cannot resist collusion attacks. Zhao et al. [**?**] proposed a revocable CP-ABE scheme based on two-party computation, which requires splitting the attribute center into an attribute authority and a central controller.

In distributed open network environments, user identity authentication is the foundation and key to ensuring security. Two-factor identity authentication technology compensates for the shortcomings of traditional password authentication. In 1999, Yang et al. [**?**] introduced the first password authentication scheme based on smart cards, which did not maintain a table of sensitive information—a key advantage over traditional schemes. To protect static user-related ID information, Das et al. [**?**] proposed a feasible method using "dynamic ID technology" to solve information attack problems. In 2015, after cryptanalyzing two important anonymous two-factor schemes [**?**, **?**], Wang et al. proposed a two-factor identity authentication scheme for distributed systems [**?**], which was shown through analysis and verification to have stronger security.

To solve the aforementioned problems, drawing on existing revocable CP-ABE decryption outsourcing schemes [**?**], this paper proposes a revocable location-hierarchical attribute-based encryption scheme. The scheme supports fine-grained attribute revocation and user revocation, improving system security. To reduce user computational burden, complex decryption calculations are outsourced to agents, making the system more flexible. Additionally, by incorporating existing anonymous two-factor authentication mechanisms [**?**], the scheme further protects user privacy and data security.

1.1 Bilinear Maps

Let G_1 and G_2 be multiplicative cyclic groups of prime order p. A bilinear map $e : G_1 \times G_1 \to G_2$ satisfies the following properties:

1) **Bilinearity:** For all $u, v \in G_1$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2) **Non-degeneracy:** There exist $u, v \in G_1$ such that $e(u, v) \neq 1$.
3) **Computability:** For all $u, v \in G_1$, $e(u, v)$ can be computed efficiently.

1.2 Linear Secret Sharing

Let P be a set of participants. A secret sharing scheme on P is linear if it satisfies the following two conditions:

a) The secret value for each participant is represented as a vector over $\mathbb{Z}_p$.
b) There exists an $l \times h$ matrix M where a mapping function ρ associates each row of M with an attribute value, with row i corresponding to the i-th attribute. A random column vector $\vec{v} = (s, r_2, \ldots, r_n)$ is selected, where s represents the secret to be shared and $r_2, \ldots, r_n$ are random values from $\mathbb{Z}_p$. The i-th share is $\lambda_i = M_i \cdot \vec{v}$, where M_i denotes the i-th row of M.

Every linear secret sharing scheme satisfies the property of linear reconstruction: Suppose $\mathbb{A}$ is a linear secret sharing scheme for access structure A, and S is any authorized set in A. If $\{\lambda_i\}$ are valid shares of secret s, there must exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ such that $\sum_{i\in I}\omega_i\lambda_i = s$, where $I = \{i : \rho(i) \in S\}$.

1.3 Diffie-Hellman DBDH Assumption

Based on the system security parameter, randomly select $a, b, c, t \in \mathbb{Z}_p$, a bilinear map $e : G_1 \times G_1 \to G_2$, and generator g of G_1, where prime p is the order of groups G_1 and G_2. The adversary' s advantage is defined as $\left|\Pr[b = b'] - \frac{1}{2}\right|$. The adversary cannot obtain $(g^a, g^b, g^c, e(g,g)^{abc})$ from $(g^a, g^b, g^c, e(g,g)^t)$ with non-negligible advantage.

1.5 Security Model

This scheme describes the security model for location-hierarchical encryption supporting revocation through a game between a challenger and an adversary. The specific process is as follows:

Setup(): The challenger runs the Setup algorithm, inputs public parameters, and outputs public key PK and attribute public key PK_{att}. The challenger sends PK and PK_{att} to the adversary.

Phase 1: The adversary can repeatedly make the following queries to the challenger:

- **Transformation key query:** The challenger selects $k = k + 1$ and runs the outsourcing key generation algorithm, returning the corresponding transformation key TK to the adversary.
- **Attribute key query:** The adversary submits an access policy corresponding to I for query, and the challenger generates the corresponding attribute key SK and sends it to the adversary.

Challenge: The adversary submits two equal-length messages M_0 and M_1 and I^*, where the decryption key corresponding to I^* cannot satisfy the access policy of I^*. The challenger randomly selects $b \in \{0, 1\}$, encrypts M_b with I^*, and sends the encrypted ciphertext CT^* to the adversary.

Phase 2: Similar to Phase 1, but the adversary cannot directly query the transformation key that can decrypt CT^*.

Guess: The adversary outputs a guess b' for b. If $b' = b$, the adversary succeeds. The adversary' s advantage in this experiment is $\Pr[b' = b] - \frac{1}{2}$.

Definition 1: A scheme is secure against chosen-plaintext attacks if no polynomial-time adversary can win the security game with non-negligible

advantage.

2 Scheme Design

Based on the location-hierarchical access policy using attribute-based encryption [**?**], this paper introduces outsourced decryption and combines it with a two-factor identity authentication method to propose a revocable CP-ABE encryption scheme.

2.1 User Registration and System Initialization

a) The server S selects an elliptic curve $E : y^2 = x^3 + ax + b \mod p$, where G is a base point of order n, and $P_{pub} = s \times G$ is the system public key. User U_i inputs their ID_i and PW_i. The server sends security parameters to the user.

b) **Setup(1^λ):** Define a hash function $H : \{0,1\}^* \to G_1$ and valid bilinear groups G_1 and G_2 of prime order p, where g is a generator of G_1. Let $A = \{h_1, h_2, \dots, h_u\}$ be the main attribute set and $A' = \{h'_1, h'_2, \dots, h'_u\}$ be the secondary attribute set. For each attribute in the system, set main attribute numbers $\{n_1, n_2, \dots, n_u\}$ and secondary attribute numbers $\{n'_1, n'_2, \dots, n'_u\}$.

Subsequently, select random numbers $\alpha, \beta, \alpha_0 \in \mathbb{Z}_p$, compute:

$$PK = \langle g, e(g,g)^\alpha, g^\beta, g^{\alpha_0}, g^{\alpha_1}, \dots, g^{\alpha_n}, g^{h_1}, \dots, g^{h_u}, g^{h'_1}, \dots, g^{h'_u}, H \rangle$$

and public attribute keys $PK_{att} = g^{\alpha_0}$ and $PK'_{att} = g^{\alpha_0}$. Set the master key $MSK = \{\alpha, \beta, \alpha_0, \alpha_1, \dots, \alpha_n\}$, which AA secretly retains. Finally, send PK and PK_{att} to the agent.

KeyGen(MSK, u): User u inputs the master key MSK. AA runs the key generation algorithm, randomly selects a unique $t \in \mathbb{Z}_p$, and combines the main attribute set and secondary attribute set to generate the transformation key for legitimate authorized user u:

$$TK = \langle K = g^{\alpha/\beta} \cdot g^{t \cdot H(x)}, K_x = g^t \cdot g^{\beta \cdot H(x)} \rangle$$

The transformation key TK is sent to the agent, and the user private key $SK = \{K, K_x\}$ is sent to the user via a secure channel.

2.2 Encryption Algorithm

The data owner generates three symmetric keys k_1, k_2, k_3 using a symmetric encryption algorithm (denoted as E) based on the user' s location information hierarchical policy, directly defining the symmetric keys k_1, k_2, k_3 as ciphertexts. Randomly encrypt location information m_1, m_2, m_3 to obtain encrypted ciphertexts C_1, C_2, C_3.

The data owner further encrypts the symmetric keys k_1, k_2, k_3 and constructs an attribute-based access policy using the LSSS matrix to represent the main policy P_1. M represents an $l \times h$ matrix, and ρ is the mapping function. Randomly generate column vector $\vec{v} = (s, v_2, \ldots, v_h)$, where s represents the secret to be shared. Randomly select $r_1, r_2, \ldots, r_l \in \mathbb{Z}_p$, then compute:

$$C = k_1 \cdot e(g,g)^{\alpha s}, \quad C' = g^s$$

$$C_i = g^{\alpha_0 \lambda_i} \cdot g^{-\beta r_i}, \quad D_i = g^{r_i} \quad \text{for } i = 1, 2, \ldots, l$$

In the process of encrypting k_2, k_3, use g^{α_0} and g^{α_1} from the system public key PK as the first stage of encryption. Output symmetric key ciphertexts C_{k_2}, C_{k_3} to complete the second stage of encryption.

2.3 Identity Authentication and Private Key Generation

The user selects random number n_i, computes $Q_i = H(PW_i \| n_i) \oplus H(ID_i \| x \| T)$, and sends $M_1 = \{ID_i, Q_i\}$ to S.

The server performs the following calculations: selects random number $c \in \mathbb{Z}_p$, computes $a = c \times P$, $e = g^c$, and $M_2 = \{a, e, H(c)\}$, then sends the information back to the user.

After receiving S' s reply, the user computes:

$$M_3 = H(ID_i \| x \| T) \oplus H(PW_i \| n_i) \oplus H(a)$$

and further calculates $M_4 = H(M_3 \| e)$. The server, upon receiving M_4, performs the operation: if $M_4 \neq H(SK \| e)$, user authentication fails; if $M_4 = H(SK \| e)$, user authentication succeeds and the private key algorithm can continue.

2.4 Decryption Algorithm

Upon receiving the user' s request, the agent inputs the corresponding transformation key TK and ciphertext CT, runs the transformation algorithm to determine whether the user' s attributes satisfy the main access policy. If satisfied, a set of constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ can be computed in polynomial time such that $\sum_{i \in I} \omega_i \lambda_i = s$ holds. Compute partial decryption ciphertext:

$$CT' = \prod_{i \in I} \left(\frac{e(C_i, K)}{e(D_i, K_x)} \right)^{\omega_i} = e(g,g)^{\alpha s}$$

Continue to determine whether user attributes match the secondary access policy. Using the previous stage' s calculation, similar to the first decryption stage, compute the remaining partial decryption ciphertexts CT'_2, CT'_3. Send the partial decryption ciphertexts CT'_1, CT'_2, CT'_3 to the user. The user computes:

$$k_1 = \frac{C}{CT'_1}, \quad k_2 = \frac{C_2}{CT'_2}, \quad k_3 = \frac{C_3}{CT'_3}$$

Finally, obtain the decrypted plaintext information through symmetric keys k_1, k_2, k_3:

$$m_1 = D_{k_1}(C_1), \quad m_2 = D_{k_2}(C_2), \quad m_3 = D_{k_3}(C_3)$$

2.5 Attribute Revocation

This phase' s revocation scheme is divided into user revocation and attribute revocation.

User Revocation: When AA revokes a specified user, that user cannot decrypt any information in the data server after leaving the system, and user u' s access rights are revoked. AA maintains a revocation list $R = \bigcup RT_u$. Users on the list will be denied partial decryption ciphertexts by the agent, thereby achieving user revocation.

Attribute Revocation: When revoking a user' s main attribute att_1, input the master key MSK and attribute key PK_{att_1}. Randomly select a new attribute number V'_{att_1} for this attribute. The attribute authority computes and outputs the updated public attribute key $PK'_{att_1} = g^{\alpha_0/V'_{att_1}}$ and publishes an announcement that the public attribute key has been updated to PK'_{att_1}. Through a secure channel, send the update key $UK_{att_1} = V_{att_1}/V'_{att_1}$ to the agent. For users who still possess the revoked attribute, the agent updates their transformation key as:

$$TK' = \langle K, K_x, \{K_{att_1} = g^{t/V'_{att_1}}\}\rangle$$

To prevent new users from being unable to access previous ciphertexts when they satisfy the access policy, all ciphertexts associated with the revoked attribute att_1 in their access policies need to be updated. AA generates the ciphertext update key $CUK_{att_1} = (V_{att_1} - V'_{att_1})$ and sends it to the agent, who updates the ciphertext to:

$$C'_{att_1} = C_{att_1} \cdot g^{CUK_{att_1}}$$

The process for revoking secondary attribute att_2 is similar to the above main attribute revocation process. The agent updates the ciphertext and transformation key TK using the new transformation key generated by the attribute authority.

3.1 Chosen Plaintext Attack

Setup: The challenger initializes the system, outputs public parameters PPA, public key PK, and attribute public key PK_{att}.

Phase 1: a) **Transformation key query:** The adversary can repeatedly query transformation keys for attributes they control. The challenger runs the outsourced key algorithm $KeyGen_{out}(MSK, I)$ to obtain transformation key

TK and private key SK. According to the DBDH assumption, the adversary' s advantage is ϵ, and they cannot obtain the private key SK corresponding to the plaintext.

b) **User private key query:** Transformation key TK corresponds to private key SK. Since neither TK nor its corresponding main attribute set can match the access policy of ciphertext M_b' s I^*, the adversary attempts to combine different transformation key TK access policies to attack the location information m_1 access policy. However, in this scheme, all attribute access policies contain random parameter t. According to the transformation algorithm, computing $K_{att} = g^{t/V_{att}}$ means the adversary cannot calculate $e(g,g)^{\alpha s}$. Based on the DBDH assumption, the adversary cannot compute the secret value S at this point. If users 1 and 2 input parameters t_1, t_2 and attempt to collude and expand their attribute sets to attack, they would need to compute $e(g,g)^{\alpha s}$. According to the DBDH assumption, the adversary' s advantage is negligible, and SK is sent to the adversary.

Challenge: The adversary submits two plaintext messages M_0 and M_1, and additionally generates I^*, where all decryption keys for I^* cannot match the access policy corresponding to I^*, meaning they cannot decrypt. The challenger randomly selects $b \in \{0,1\}$, encrypts plaintext M_b, and sends ciphertext CT^* to the adversary.

Phase 2: Repeat the query steps from Phase 1.

Guess: Output the adversary' s guess b' for b. Further determine whether it matches the encrypted ciphertext of M_0 and M_1, i.e., whether b' equals b. If $b' = b$, the adversary succeeds. The adversary' s advantage in the above process is $\Pr[b' = b] - \frac{1}{2}$.

3.2 Data Confidentiality

Data confidentiality is defined as: only legitimate users and data owners can decrypt and obtain plaintext messages; illegal users and revoked users cannot access data. In this scheme, the security of the second encryption stage relies on the NPC problem of discrete logarithm decomposition in the Elgamal algorithm. Assuming an unauthorized user has satisfied the first-stage access policy and decrypted to obtain symmetric key k_1, if this user wants to continue attacking to obtain keys k_2, k_3, they must be able to obtain random numbers n_3, n_4. However, calculating the values of n_3 and n_4 from $g^{\alpha_0}, g^{\alpha_1}, g^{\alpha_2}$ in the system public key PK is computationally infeasible. This computational problem belongs to the NPC problem of discrete logarithm decomposition, thereby verifying data confidentiality.

3.3 Resistance to Collusion Attacks

In the first decryption stage, users must first obtain $e(g,g)^{\alpha s}$ to recover the symmetric encryption key k_1. Normally, if two users cannot satisfy the main policy, they cannot recover the secret value s from $e(g,g)^{\alpha s}$. However, these two users might satisfy parts of the main attribute policy and could potentially extend their permissions by exchanging main attribute parameters. To prevent this, random number t is introduced into transformation key TK. In this scheme, all K_x contain parameter t, so colluding users cannot decrypt by combining their private keys.

In the second stage, the user' s goal is to obtain keys k_2, k_3. If the user has already obtained $e(g,g)^{\alpha s}$ from the first-stage decryption process, they still need n_3 and n_3+n_4 to decrypt. This system does not directly release the parameters $n_1, n_2, \ldots, n_u$ that users desire but instead releases secondary attribute parameters, preventing users from decrypting. Further discussion: when two users cannot satisfy the secondary attribute policy, they cannot access n_3, n_3+n_4 contained in the private key and thus cannot decrypt. Users who partially match the secondary attribute policy could exchange their parameters $n_1, n_2, \ldots, n_u$ to launch a coordinated attack. Using the same approach as in the first stage, random number t is also employed here, preventing colluding users from achieving decryption by combining private key parameters. Verification shows that this scheme can resist collusion attacks.

3.4 Forward Security and Backward Security

Forward Security: Users whose attributes have been revoked cannot decrypt updated ciphertexts associated with those attributes, even if they possess the attributes. Taking secondary attribute revocation as an example, when attribute revocation occurs, AA generates a new attribute number V'_{att} for the revoked attribute att and upgrades the key TK for users who have not revoked this attribute. Users who have revoked this attribute cannot upgrade their keys, while AA simultaneously upgrades related ciphertexts. If a user with revoked attribute att attempts to decrypt the updated ciphertext using their previous key, according to the data decryption algorithm, the final decryption result would be $e(g,g)^{\alpha s \cdot (V_{att}/V'_{att})}$, while the adversary cannot compute $e(g,g)^{\alpha s \cdot (V_{att}/V'_{att})}$, thus cannot decrypt to obtain k_2.

Backward Security: Newly joined users can still decrypt previous ciphertexts after AA updates the re-encrypted ciphertexts. For new users, their private keys are generated from the updated attribute key CUK. AA sends the updated attribute key to the agent, who is responsible for upgrading related ciphertexts, so new users can decrypt previous ciphertexts, ensuring backward security of the scheme.

---

4 Scheme Analysis

This chapter provides a comparison of the proposed scheme with related schemes [?, ?, ?, ?, ?] in terms of functionality and computational efficiency.

4.1 Functionality Comparison

As shown in Table 1 , reference [?] uses hierarchical attribute-based encryption and optimizes the algorithm, but its security and computational cost reduction need improvement. Reference [?] achieves immediate attribute-level revocation capability but does not implement fine-grained access control. Reference [?] proposes a key-escrow-free revocable attribute-based scheme that achieves fine-grained access control but requires two-party computation between a central controller and authorization institution, which does not integrate well with hierarchical attribute-based encryption. Based on these works, our scheme supports both user revocation and attribute revocation, outsources extensive decryption computations to agents, and reduces user computational burden.

Table 1. Comparison of Scheme Functions

Scheme	Access Structure Type	Revocation Mechanism	Revocation Granularity	Supports Location Hierarchy	Decryption Outsourcing
[?]	Access tree	-	-	No	No
[?]	LSSS	-	-	No	No
[?]	LSSS	-	-	Yes	No
[?]	Access tree	Revocable	Attribute revocation	No	No
[?]	LSSS	Revocable	Attribute/user revocation	No	No
Our scheme	LSSS	Revocable	Attribute/user revocation	Yes	Yes

4.2 Efficiency Analysis

This paper addresses the security and computational issues in reference [?] by proposing a scheme combining revocation support and decryption outsourcing while ensuring the original scheme' s location-hierarchical multi-encryption advantages continue without impacting encryption/decryption efficiency. Therefore, this paper analyzes the efficiency of two-stage and three-stage attribute encryption. As shown in Tables 2 and 3 , adding revocation functionality does not affect the original scheme' s encryption efficiency, and the introduced decryption outsourcing reduces user decryption burden.

Symbol Definitions: p denotes bilinear pairing operations, e denotes modular exponentiation operations, and a, b represent the number of main and secondary attributes, respectively.

Table 2. Efficiency Analysis of Two-Stage Attribute Encryption

Scheme	Structure Type	Related Attributes	Encryption Cost	Decryption Cost
[?]	Access tree	-	$(4a+4b+2)e+2p$	$(2a+2b)e+(2a+2b+4)p$
[?]	LSSS	-	$(6a+6b+2)e+2p$	$(2a+2b)e+(4a+4b+2)p$
[?]	LSSS	-	$(3a+1)e+2p$	$(a+1)e+(2a+2)p$
Our scheme	LSSS	-	$(3a+1)e+2p$	$(a+1)e+(2a+2)p$

Table 3. Efficiency Analysis of Three-Stage Attribute Encryption

Scheme	Structure Type	Related Attributes	Encryption Cost	Decryption Cost
[?]	Access tree	-	$(6a+6b+3)e+3p$	$(3a+3b)e+(3a+3b+6)p$
[?]	LSSS	-	$(9a+9b+3)e+3p$	$(3a+3b)e+(6a+6b+3)p$
[?]	LSSS	-	$(3a+1)e+3p$	$(a+2)e+(2a+2)p$
Our scheme	LSSS	-	$(3a+1)e+3p$	$(a+2)e+(2a+2)p$

5 Conclusion

This paper proposes a location-hierarchical attribute-based encryption scheme supporting revocation. Without affecting encryption efficiency and hierarchical functionality, the scheme combines a two-factor identity authentication mechanism to enhance system security, improve decryption efficiency, and reduce user computational load. The scheme also offers advantages in data confidentiality and resistance to user collusion attacks. Through functional comparison and efficiency analysis, results demonstrate that the proposed scheme achieves high decryption efficiency and certain security guarantees.

References

[1] Kido H, Yanagisawa Y, Satoh T. An anonymous communication technique

using dummies for location-based services [C]// Proc of International Conference, Pervasive Services. 2005: 88-97.

[2] Yiu Manlung, Christian S, Jensen, et al. SpaceTwist: managing the tradeoffs among location privacy, query performance and query accuracy in mobile services [C]// Proc of the 24th IEEE International Conference on Data Engineering. Washington DC: IEEE Computer Society, 2008: 366-375.

[3] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption [C]// Proc of IEEE Symposium on Security and Privacy. Washington DC: IEEE Computer Society. 2007: 321-334.

[4] Beimel A. Secure schemes for secret sharing and key distribution [D]. Israel: Israel Institute of Technology. 1996.

[5] Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization [J]. Public Key Cryptography, 2011: 53-70.

[6] Lin Xi, Han Yiliang, et al. Location hierarchical access control scheme based on attribute encryption [C]// Proc of the 36th Chinese Control Conference.

[7] Pirretti M, Traynor P, Mcdaniel P, et al. Secure attribute-based systems [C]// Proc of the 13th ACM Conference on Computer and Communications Security. 2006: 99-112.

[8] Hur J, Noh D K. Attribute-based access control with efficient revocation in data outsourcing systems [J]. IEEE Trans on Parallel and Distributed Systems, 2011, 22 (7): 1214-1221.

[9] Zhao Zhiyuan, Zhu Zhiqiang, Wang Jianping, et al. Revocable attribute-based encryption scheme with escrow-free in cloud storage [J]. Journal of Electronics & Information Technology, 2018, 40 (1): 1-10.

[10] Yang Wenher, Shieh Shiuhping. A Password authentication schemes with smart cards [J]. Computers & Security, 1999, 18 (8): 727-733.

[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] Tsai Jialun, Lo Naiwei, Wu Tzongchen. Novel anonymous authentication scheme using smart cards [J]. IEEE Trans on Ind. Inform, 2013, 9 (4): 2004-2013.

[13] Li Chunta. A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card [J]. IET Inform. Security, 2013, 7 (1): 3-10.

[14] 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.

[15] Li Yong, Zeng Zhenyu, Zhang Xiaofei. Outsourced decryption scheme supporting attribute revocation [J]. J Tsinghua University: Sci&Technol, 2013, 53 (12): 1664-1669.

[16] Chi Shuiming, Chen Qin, Dang Zhengqin. An attribute-based encryption scheme with attribute revocation and key delegation based on policy control [J]. Computing Engineering & Science, 2013, 35 (9): 94-98.

[17] Yan Xixi, Meng Hui. Ciphertext policy attribute based encryption scheme supporting direct revocation [J]. Journal on Communications, 2016, 37 (5): 44-50.

[18] Huang Qinlong, Ma Zhaofeng, Yang Y, et al. Attribute-based secure data sharing with efficient revocation in cloud computing [J]. Chinese Journal of Electronics, 2015, 24 (4): 862-868.

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