Revocable Attribute-Based Encryption Scheme with Cryptographic Reverse Firewalls

Yang Zhao[1,2], Xing-Yu Ke[1,2], Yu-Wei Pang[1,2], Hu Xiong[1,2], Guo-Bin Zhu[1,2], and Kuo-Hui Yeh[3](✉)

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

[2] Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu 610054, China

[3] Department of Information Management, National Dong Hwa University, Hualien, Taiwan

khyeh@gms.ndhu.edu.tw

Abstract. With the prevalence of information sharing, preserving the confidentiality of sensitive data has become paramount. Attribute-based encryption (ABE) has become a viable option to tackle this problem. Using a set of attributes, data owners can encrypt data with ABE, and data is only accessible by users with the required attributes and authorization. However, there are various limitations associated with the traditional CP-ABE scheme, such as embedding user-sensitive information in the access structures without any hidden operations, an inability to effectively address the issue of user attribute changes, and vulnerability to internal attacks from cryptography devices. To address these limitations, researchers have proposed various enhanced ABE schemes. Mironov presented a concept of cryptographic reverse firewall (CRF) in Eurocrypt 2015, which could resist certain compromised machines from leaking secret information. The CRF has been deployed in many cryptographic systems, but its application in the ABE field has been relatively limited. This paper presents a novel attribute-based encryption scheme which incorporates attribute revocation, hidden policy components, and CRF mechanism to prevent attackers from internal attacks on cryptography devices. This scheme is applicable in various applications, such as cloud computing, where secure data sharing is required.

Keywords: Attribute-based Encryption · Attributes revocation · Partial Hidden policy · Cryptographic Reverse Firewalls · Data sharing

1 Introduction

While traditional encryption methods suffer from three major drawbacks: (1) To ensure secure encryption, resource providers need the user's genuine public key certificate before proceeding with encryption. (2) messages must be encrypted

Z. Tan et al. (Eds.): BDTA 2023, LNICST 555, pp. 79–94, 2024.
https://doi.org/10.1007/978-3-031-52265-9_6

individually using the public key of each user, resulting in high processing overheads and bandwidth consumption issues; and (3) broadcast encryption technology, while partially solving the efficiency problem [5], requires resource providers to obtain the user list before encryption, creating two secondary problems: distributed applications cannot obtain the size of the receiving group at once, and listing user identities may compromise user privacy.

Shamir and Boneh et al. [4,19] introduced identity-based encryption (IBE) mechanisms to address the issue that resource providers are required to acquire a user's public key certificate, while Sahai and Waters presented attribute-based encryption (ABE) mechanisms built upon IBE technology to solve issues of high processing overhead and bandwidth consumption [18]. ABE mechanisms have four key features that make them promising for fine-grained access control [9, 22], targeted broadcasting [9], group key management [7,8], privacy protection [22,23], and other fields. These features include the ability for resource providers to encrypt messages based on attributes, without paying attention to group size or identity, ensuring only group members with required attributes can decrypt messages, preventing collusion attacks by users, and supporting flexible access control policies based on attribute conjunctions, disjunctions, negations, and threshold operations.

In 2013, Edward Snowden released a large number of documents to the media revealing various surveillance programs of the US government. These programs allowed the government to monitor user privacy on a large scale by obtaining data from super-large internet companies such as Microsoft and Google. Additionally, the NSA installed backdoor programs in widely used public encryption standards and intercepted hardware sent to users to tamper with programs for monitoring purposes [10]. Bellare et al. [2] presented an algorithm replacement attack. Mironov and Stephons-Davidowitz [15] introduced the CRFs in 2015 that intercepts and modifies both inbound and outbound messages to enhance security protection. However, few papers have proposed a cryptographic reverse firewall suitable for ABE, and its addition may increase time overhead and require further study on algorithm efficiency.

1.1 Related Work

ABE [18] is an encryption scheme that allows access control based on user attributes rather than their identity. ABE is commonly categorized into key-policy ABE [9] and ciphertext-policy ABE [3]. KP-ABE embeds access policy into secret key, and attributes set into ciphertext. On the contrary, CP-ABE embeds attribute set into the key, while access policy into ciphertext. In order to successfully decrypt and access a secret message, their attribute set must satisfy the access policy requirements.

Within context of attribute-based encryption (ABE), user attributes may change frequently, leading to the revocation of certain attributes. Two types of revocation can be implemented: user revocation and attribute revocation. Revoking user entails invalidating all attributes that have been assigned to a particular user. The term attribute revocation, on the other hand, refers to the situation where a user's access rights are restricted because a particular attribute has been revoked. To achieve attribute revocation, Pirretti et al. [17] presented a method in 2010 that utilizes a timed update key mechanism. Subsequently, Wang et al. [20] introduced group key forms and binary trees to implement attribute revocation in their respective schemes in 2018. Notably, none of these schemes support decryption testing. Zhang [25] presented a different method that reduces the reliance on bilinear pairing. However, this scheme lacks support for attribute revocation. In 2021, a scheme was proposed by Zeng et al. [24] that can handle large attribute domains, but lacks support for attribute revocation.

Chen et al. [6] put forward cryptographic reverse firewall (CRF) and suggested a smooth projective hash function (SPHF) as a technique for building CRFs. However, their CRF construction is not applicable to attribute-based encryption due to its complexity. To address this limitation, Yuyang Zhou [26] presented a CRFs method for certificateless public key encryption, while Mengdi Ouyang [16] presented a non-monotonic access structure-based scheme with CRFs for identity-based signature. In addition, BO HONG et al. [11] presented a Multi-Authority KP-ABE scheme with Cryptographic Reverse Firewalls, and Hui Ma [14] designed a CRFs scheme based on online/offline CP-ABE. Although various attempts have been made, a CRF scheme appropriate for CP-ABE that includes attribute revocation and partial policy hiding has yet to be proposed. Xiong et al. [21]introduced a secure efficient revocable PRS (R-PRS) scheme

Therefore, there is still a need to develop a CRFs scheme that can address the challenges associated with CP-ABE, such as attribute revocation and policy partial hiding. This will be an great improvement for CP-ABE and enhancing the security of attribute-based encryption schemes.

1.2 Our Contribution

To overcome these challenges, this paper presents a novel ABE scheme with reverse firewall, called RH-CPABE-CRF. Our proposed scheme provides a robust security framework for data sharing in dynamic environments by supporting attribute revocation and hidden policy delegation. The cryptography reverse firewall ensures that our scheme can support protection from malicious attacks from within the device.

Resist Internal Attacks. We extended the CRFs insecure for the base scheme which are used to recalculate important parameters to against the leakage of inner attack and also keep functionality.

Partial Policy Hiding: We have implemented partial policy hiding in our scheme by only hiding attribute values, while keeping attribute names visible. This ensures efficient implementation while maintaining a level of security.

Attribute Revocation: Our scheme supports attribute-level revocation for users. The CA creates a group key for each user attribute, which are organized in a binary tree structure. This enables efficient storage and retrieval of user attributes, while also allowing for easy revocation of specific attributes as needed.

Large Attribute Universe: Our scheme has public parameters of constant size, ensuring that its performance remains efficient and scalable even with an increasing number of user attributes.

2 Preliminary

2.1 Bilinear Groups

Assume $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ denotes two cyclic groups of prime order p. Bilinear pairing e between these groups is rigorously defined when certain conditions are met:

1. Bilinearity: $\forall g_1, g_2 \in \mathbb{G}$, $\forall m, n \in \mathbb{Z}_p^*$, such that $e\left(g_1^m, g_2^n\right) = e(g_1, g_2)^{mn}$.
2. Non-degeneracy: e is non-degenerate meaning that $P \in \mathbb{G}$ and $e(P, P) \neq 1$.
3. Computability: $\forall g_1, g_2 \in \mathbb{G}$, $e(g_1, g_2)$ must be efficiently computable.

2.2 Access Structures

The collection of participants $\mathcal{T} = \{P_1, \cdots, P_n\}$ is defined in our scheme. Access structure is then defined as $B \subseteq 2^{\mathcal{T}}$ of non-empty subsets of $\mathcal{T}$ with monotonicity, meaning that if $E \in B$ and $E \subseteq R$, then $R \in B$. This access structure is used to determine which participants have access to encrypted data.

2.3 Linear Secret Sharing Schemes

Definition 1. *Linear Secret Sharing Scheme (LSSS): To qualify as a LSSS over the field $\mathbb{Z}_p$, certain requirements must be met:*

1. *Each participant is assigned a vector over $\mathbb{Z}_p$ for their share.*
2. *Let M be a $d \times n$ matrix that serves as the shared generating matrix. For each $i \in 2, \ldots, d$, let $\rho(i)$ denote the party labeling row i. Each column vector $v = (s, r_2, \ldots, r_n)$, where s is the secret to be shared and r_2 to r_n are randomly selected variables in $\mathbb{Z}_p$, produces a vector M_v of d shares of the secret $s.s$*

In [1], the authors proved that a LSSS satisfying definition 1 has the linear reconstruction property. Specifically, given an access structure $\mathbb{A}$ corresponding to an LSSS scheme, and an authorized set $S \in \mathbb{A}$, one can find constants $\omega_i \in \mathbb{Z}_p, i \in I$ in polynomial time, where $I = i : \rho(i) \in S$, such that λ_i is a valid share and $\sum_{i \in I} \omega_i \lambda_i = s$. This means that the authorized parties are able to rebuild s using their shares, while unauthorized parties cannot.

3 System Model

Our proposed Revocable CP-ABE Scheme with CRFs (RH-CPABE-CRF) consists of six entities (Fig. 1):

Central Authority (CA): The Central Authority creates and manages the global public parameters used in the cryptographic system, as well as generating secret keys and sent to users.

Cloud Service Provider (CSP): CSP stores and manages access to data and also provides assistance with the re-encryption or updating of ciphertext when attribute revocation occurs.

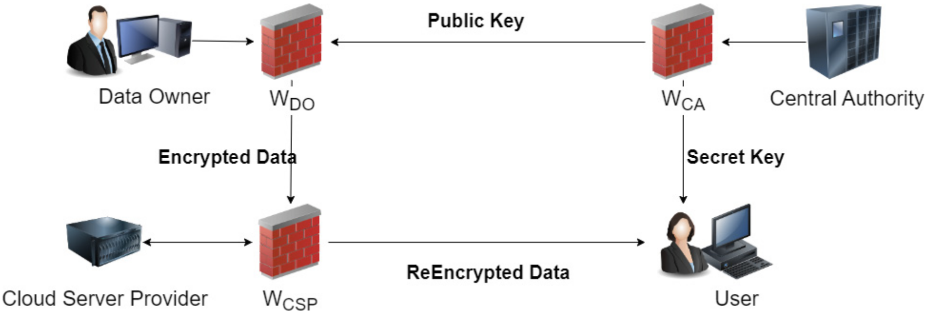


Fig. 1. System Model

Data Owner (DO): DO selects access policy for attributes and performs encryption accordingly, without relying on the CSP for access control. User decryption privileges based on attributes are used for access control within the cryptography.

User: Users are assigned with attributes. Decryption is only allowed when attributes meet access policy, and this can be done using their corresponding secret keys.

Firewall of CA (W_{CA}): W_{CA} intercepts the public parameters published by the CA, modifies a portion of them, and then republishes them. When a user obtains their secret key, W_{CA} also scrambles the user's secret key.

Firewall of CSP (W_{CSP}): W_{CSP} intercepts the encrypted message sent by the DO, performs pre-re-encryption processing, and then forwards it to the CSP for re-encryption. Once the CSP completes re-encryption, W_{CSP} performs post-re-encryption processing on the message before forwarding it to the user.

Firewall of DO (W_{DO}): After the DO encrypts the message, W_{DO} re-randomizes the encrypted message before sending it to the CSP.

3.1 Security Model

Our security model involves a challenger $\mathcal{C}$ and an adversary $\mathcal{D}$.

1. SetUp: Upon execution of SetUp, the party $\mathcal{C}$ obtains public parameters PK and master key MSK. PK are disclosed to $\mathcal{D}$.
2. Query phase 1: $\mathcal{D}$ can issue two kind of queries.
   (a) Secret keys query: $\mathcal{C}$ generates the secret key $sk_{id,\mathcal{S}}$ using the KeyGen algorithm with the given id and attribute set $\mathcal{S}$ provided by $\mathcal{D}$, and then sends $sk_{id,\mathcal{S}}$ back to $\mathcal{D}$ in response to a secret key query.
   (b) Decryption query: $\mathcal{C}$ can run the KeyGen and Decrypt algorithms on the ciphertext provided by $\mathcal{D}$ to obtain the corresponding plaintext M, which it then sends to $\mathcal{D}$ as a response.
3. Challenge: $\mathcal{D}$ sends two messages S_0 and S_1 of same length, along with two access structures $\mathbb{A}_0$ and $\mathbb{A}_1$ to $\mathcal{C}$. $\mathcal{C}$ picks $\omega \overset{R}{\leftarrow} \{0,1\}$, then runs encryption algorithm and re-encryption algorithm to obtain the encrypted data CT'_ω. CT'_ω is returned to $\mathcal{D}$ such that either $\mathbb{A}_0$ or $\mathbb{A}_1$ cannot be fulfilled by any subset in $\mathcal{S}$).
4. Query phase 2: the adversary $\mathcal{D}$ is restricted from issuing secret key and decryption queries using attribute sets $\mathcal{S}'$ that can satisfy either of the access structures $\mathbb{A}_0$ or $\mathbb{A}_1$, and the ciphertext used in these queries cannot be the same as the challenge ciphertext CT'_ω.
5. Guess: $\mathcal{D}$'s guess bit ω' is checked against the randomly selected bit ω by the challenger. If $\omega' = \omega$, then $\mathcal{D}$ wins the game, otherwise, the challenger declares the game a failure for $\mathcal{D}$.

If $\mathcal{D}$ can correctly guess the value of ω with a significant advantage, then the security of this scheme is compromised (Fig. 2).

4 Our Construction

- **Setup**: It produced bilinear pairing $(\mathbb{G}, \mathbb{G}_T, e)$ over a composite order $N = p_1 p_2 p_3 p_4$, given security parameter λ as input. It initializes the attribute universe $\mathcal{U}$ as $\mathbb{Z}_N$, then it selects random values $\beta, b \xleftarrow{R} \mathbb{Z}_N$, $g \xleftarrow{R} \mathbb{G}p_1$, $p \xleftarrow{R} \mathbb{G}p_3$, and $q, r \xleftarrow{R} \mathbb{G}p_4$. Lastly, it returns public parameters PK $= \big(N, \mathbb{G}, \mathbb{G}_T, e, g, g^b, e(g,g)^\beta, p, q, r\big)$ and master key MSK $= \beta$.
- **KeyGen**: TA selects random values $k \overset{R}{\leftarrow} \mathbb{Z}_N$ and $d, \eta, \mu_i \overset{R}{\leftarrow} \mathbb{G}p_2$, where $\forall i \in \mathcal{I}_S$. It takes PK, MSK, id, and $\mathcal{S}$ as input to return the private key

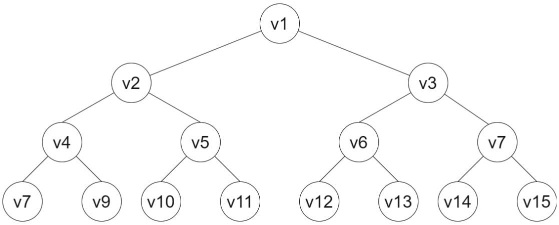


Fig. 2. $Tree_x$ for attribute x

$sk_{id,\mathcal{S}} = \left(\mathcal{S}, K, K'', \{K_i\}_{i \in \mathcal{I}_S}\right)$, in which $K = g^k d$, $K'' = g^\beta g^{bk} \eta$, and $K_i = \left(g^{s_i}\right)^k \mu_i$, $\forall i \in \mathcal{I}_S$, and $\mathcal{I}_S$ is attribute name index.

- **AttrGroupKeyGen**: $AttrGroupKeyGen(x) \rightarrow (KEK_{TREE_x}, AGK)$: It takes an attribute x as input and generates the necessary keys for the attribute user group AG_x. TA creates a binary state tree $TREE_x$ [13] to represent the attribute and maintains the group AG_x which includes all users possessing the attribute. Every node v_j in the tree corresponds to a random key $KEK_j \in \mathbb{Z}_p$, and the leaf nodes corresponds to the elements of AG_x. For a user u in the group, $PATH_u$ represents the path from leaf node assigned to u to root node, and u stores $PATH_u$ as its path keys. The CA also generates a unique attribute group key $AGK_x \in \mathbb{Z}_p$, which is shared with CSP after encrypting it using KEK as a symmetric key.
- **Encryption**: It inputs message M, public key PK, access policy $\mathbb{A}$ which corresponds to the LSSS (with dimensions $\ell \times n$), a vector $\mathcal{T}$ representing attribute values, and a mapping ρ from $\{1, 2, \ldots, \ell\}$ to the attribute name universe. It then picks random vectors v and v' in attribute name space, along with $2(\ell + 1)$ random subgroup elements r_1, r_2, $r_{1,x}$, and $r_{2,x}$, where $x \in \{1, 2, \ldots, \ell\}$. Lastly, a message $R \in \mathbb{G}_T$ is randomly selected to compute the ciphertext CT:

$$\mathrm{CT} = \Big((\mathbb{A}, \rho), \hat{W}, W_1, W_2, \hat{W}_1, \hat{W}_2, \{W_{1,x}, W_{2,x}\}_{x \in [\ell]}\Big) \tag{1}$$

where we have $\hat{W} = p^{H(M)} q^{H(R)} ('H'$ is a hash function$), W_1 = M \cdot e(g,g)^{\beta \cdot s}, \hat{W}_1 = g^s r_1, W_{1,x} = g^{bA_x \cdot v} g^{-s \cdot t_{\rho(x)}} r_{1,x}, W_2 = R \cdot e(g,g)^{\beta \cdot s'}, \hat{W}_2 = g^{s'} r_2, W_{2,x} = g^{bA_x \cdot v'} g^{-s' \cdot t_{\rho(x)}} \cdot r_{2,x}$, then sends to CSP.
- **ReEncryption**:$ReEncrypt\left(CT, \{AGK_i\}_{i \in \mathcal{I}_S}\right) \rightarrow CT'$: CSP obtains the CT from data owner generated by $Encryption$, and $\{AGK_i\}_{i \in \mathcal{I}_S}$ from CA generated by $KeyGeneration$. This algorithm applies re-encryption on CT using $AGK_{\rho(i)}$ corresponding to each attribute $\rho(i)$ in $\mathbb{A}$ and obtains CT$'$.

$$\mathrm{CT}' = \Big((\mathbb{A}, \rho), \hat{W}, W_1, \hat{W}_1, W_2, \hat{W}_2, \left\{W'_{1,x}, W'_{2,x}\right\}_{x \in [\ell]}\Big) \tag{2}$$

where $W'_{1,x} = W_{1,x}^{AGK_{\rho(x)},}, W'_{2,x} = W_{2,x}^{AGK_{\rho(x)}}$.
- **Decryption**:$Decryption\left(CT', \mathcal{S}, PK, \{AGK_i\}_{i \in \mathcal{I}_S}, sk_{id,\mathcal{S}}\right) \rightarrow (m/\perp)$: For attribute $\rho(x)$ in S, the user recovers $AGK_{\rho(x)}$ using the specific path keys $\{KEK_i\}_{i \in \mathcal{I}_S}$, and then calculates $K^* = (K)^{1/AGK_{\rho(x)}}$. Finally it computes:

$$\begin{aligned}
&W_2 \cdot \frac{\prod_{x\in\mathcal{X}}\left(e\left(W'_{2,x},K^*\right)\cdot e\left(\hat{W}_2,K_{\rho(x)}\right)\right)^{w_x}}{\hat{e}\left(\hat{W}_2,K''\right)}\\
&=W_2 \cdot \frac{\prod_{x\in\mathcal{X}}\left(e\left(W_{2,x},K\right)\cdot e\left(\hat{W}_2,K_{\rho(x)}\right)\right)^{w_x}}{\hat{e}\left(\hat{W}_2,K''\right)}\\
&=R\cdot e(g,g)^{\beta\cdot s'}\cdot\frac{\prod_{x\in\mathcal{X}}\left(e(g,g)^{bA_x v' k}\right)^{w_x}}{\hat{e}\left(g^{s'},g^{\beta}\cdot g^{bk}\right)}\\
&=R\cdot e(g,g)^{\beta\cdot s'}\cdot\frac{e(g,g)^{\sum_{x\in\mathcal{X}}\left(A_x\omega_x v'\right)bk}}{\hat{e}\left(g^{s'},g^{\beta}\cdot g^{bk}\right)}\\
&=R
\end{aligned}\tag{3}$$

$$\begin{aligned}
&W_1 \cdot \frac{\prod_{x\in\mathcal{X}}\left(e\left(W'_{1,x},K^*\right)\cdot e\left(\hat{W}_1,K_{\rho(x)}\right)\right)^{w_x}}{\hat{e}\left(\hat{W}_1,K''\right)}\\
&=W_1 \cdot \frac{\prod_{x\in\mathcal{X}}\left(e\left(W_{1,x},K\right)\cdot e\left(\hat{W}_1,K_{\rho(x)}\right)\right)^{w_x}}{\hat{e}\left(\hat{W}_1,K''\right)}\\
&=M\cdot e(g,g)^{\beta\cdot s}\cdot\frac{\prod_{x\in\mathcal{X}}\left(e(g,g)^{bA_x v k}\right)^{w_x}}{\hat{e}\left(g^{s},g^{\beta}\cdot g^{bk}\right)}\\
&=M\cdot e(g,g)^{\beta\cdot s}\cdot\frac{e(g,g)^{\sum_{x\in\mathcal{X}}\left(A_x\omega_x v\right)bk}}{\hat{e}\left(g^{s},g^{\beta}\cdot g^{bk}\right)}\\
&=M
\end{aligned}\tag{4}$$

It returns M if $\hat{W}=p^{H(M)}q^{H(R)}$.

- **Revocation**: CA update the membership of AG_u and select a new attribute group key $AGK'_u \in \mathbb{Z}_p$ for the affected attribute. Then, CA computes a new minimum cover set G_u, which consists of descendant nodes covering unrevoked users. To update the keys for attribute u, the CA encrypts AGK'_u using KEK_v for the affected attribute and sends to the unaffected users. Attribute-level revocation can be achieved using the following two algorithms. $KeyReGen\left(\mathcal{S}, sk_{id,\mathcal{S}}, u, AGK'_u\right) \rightarrow sk'_{id,\mathcal{S}}$: If u is the attribute be revoked, $\rho\left(j'\right) = u$. Unaffected users obtain AGK'_u from $\{AGK'_u\}_{KEK_{G_u}}$ by using KEK, where $KEK \in \left(KEK_{G_u} \cap PATH_{gid}\right)$. It updates $sk'_{id,\mathcal{S}} = (\mathcal{S}, K, K'', \{K^*_j\}_{j\in\mathcal{I}_S})$, where

$$\forall j \in [l]\backslash\{j'\} : K^*_j = \left(\left(g^{s_i}\right)^k \mu_i\right)^{\frac{1}{AGK_{\rho(j)}}}, j = j' : K^*_j = \left(\left(g^{s_j}\right)^k \mu_j\right)^{\frac{1}{AGK'_u}} \tag{5}$$

$CTReGen\left(CT', u, AGK'_u\right) \rightarrow CT^*$: It randomly picks vectors $v'' = \left(s'', v''_2, \ldots, v''_n\right)^T, v''' = \left(s''', v'''_2, \ldots, v'''_n\right)^T$. Updates CT^*:

$$\mathrm{CT}^* = \Big((A, \rho), \hat{W}, W''_1, \hat{W}''_1, W''_2, \hat{W}''_2, \left\{W''_{1,x}, W''_{2,x}\right\}_{x\in[\ell]}\Big) \tag{6}$$

where $W''_1 = M \cdot \hat{e}(g,g)^{\beta\cdot s''}, \hat{W}''_1 = g^{s''} r_1, W''_{1,x} = \Big(g^{bA_x\cdot v''} g^{-t_{\rho(x)}s''} r_{1,x}\Big)^{AGK_{\rho(x)}}$, $W''_2 = R \cdot \hat{e}(g,g)^{\beta\cdot s'''}, \hat{W}''_2 = g^{s'''} r_2, W''_{2,x} = \Big(g^{bA_x\cdot\hat{v}} g^{-t_{\rho(x)}s'''} r_{2,x}\Big)^{AGK_{\rho(x)}}$.

5 Our Construction with CRFs

To enhance the confidentiality and integrity of TA, CSP, and DA, we introduce a revocable CP-ABE scheme with CRFs that builds upon the basic revocable CP-ABE. Three reverse firewalls are introduced: W_{TA}, W_{CSP} and W_{DO}. These firewalls are used to rerandomize PK, MK, $sk_{id,\mathcal{S}}$, CT and CT'.

- *Setup*: TA runs setup algorithm to generate PK $= (N, \mathbb{G}, \mathbb{G}_T, e, g, g^b, e(g,g)^\beta, p, q, r)$ sent to other entities and master key MSK $= \beta$ kept secret.
- ***W_{TA}.Setup***: W_{TA} receives PK and MSK, then randomly choose z_1, z_2, β' from $\mathbb{Z}_N$, let $\tilde{p} = p^{z_1}, \tilde{q} = q^{z_2}$, get updated

$$\tilde{PK} = \Big(N, \mathbb{G}, \mathbb{G}_T, e, g, g^b, e(g,g)^{\beta'}, \tilde{p}, \tilde{q}, r\Big) \tag{7}$$

  and $\tilde{MSK} = \beta'$.
- Key Generation: TA takes $\tilde{PK}, \tilde{MSK}$, user identity id and $\mathcal{S}$ as input to run $KeyGen$ to obtain $sk_{id,\mathcal{S}}$ and sends to user. CA takes attribute set $\{x\}$ as input to run $AttrGroupKeyGen(x)$.
- ***W_{TA}.KenGen***: W_{TA} receives $sk_{id,\mathcal{S}}$ from TA, W_{TA} randomly chooses m, n from $\mathbb{Z}_N$, let $\tilde{K} = g^m K, \tilde{K}_i = g^n \left(g^{s_i}\right)^k \mu_i, i \in \mathcal{I}_S, \tilde{K}'' = g^{m+n} K'', \tilde{sk}_{id,\mathcal{S}} = \left(\mathcal{S}, \tilde{K}, \tilde{K}'', \left\{\tilde{K}_i\right\}_{i\in\mathcal{I}_S}\right)$ and sends to user.
- Encryption: Data owner takes message M, public key PK, access policy $\mathbb{A}$ as input to generate CT $= \Big((\mathbb{A}, \rho), \hat{W}, W_1, W_2, \hat{W}_1, \hat{W}_2, \{W_{1,x}, W_{2,x}\}_{x\in[\ell]}\Big)$, then sends to CSP.
- ***W_{DO}.Encrypt***: W_{DO} receives CT, then randomly choose $\{h_{1,x}, h_{2,x}\}_{x\in[\ell]}$ and h_1, h_2 from $\mathbb{Z}_p$, where we have $\sum_{x\in[\ell]} h_{1,x} = h_1, \sum_{x\in[\ell]} h_{2,x} = h_2$. then we compute $\tilde{W}_1 = g^{h_1} W_1, \tilde{W}_2 = g^{h_2} W_2, \tilde{\hat{W}}_1 = g^{h_1}\hat{W}_1, \tilde{\hat{W}}_2 = g^{h_2}\hat{W}_2, \tilde{W}_{1,x} = \left(g^b\right)^{h_{1,x}} \frac{W_{1,x}}{i}, \tilde{W}_{2,x} = \left(g^b\right)^{h_{2,x}} \frac{W_{2,x}}{i}$. get updated

$$\tilde{CT} = \Big((\mathbb{A}, \rho), \hat{W}, \tilde{W}_1, \tilde{W}_2, \tilde{\hat{W}}_1, \tilde{\hat{W}}_2, \left\{\tilde{W}_{1,x}, \tilde{W}_{2,x}\right\}_{x\in[\ell]}\Big) \tag{8}$$

- ***W_{CSP}.PreReEncrypt***: W_{CSP} obtains the $\{AGK_x\}_{x\in[\ell]}$ from CSP, then randomly picks $\{t_x\}_{x\in[\ell]} \in \mathbb{Z}_p$, get updated $\{A\tilde{G}K_x\}_{x\in[\ell]}$ where $A\tilde{G}K_x = t_x AGK_x$, and save t_x.
- ReEncryption: CSP takes $\tilde{CT}$ generated by *Encryption*, and $\{AGK_i\}_{i\in\mathcal{I}_S}$ generated by *KeyGeneration* as input to run *ReEncrypt* and obtains CT′.

$$\mathrm{CT}' = \left((\mathbb{A},\rho), \hat{W}, W_1, \hat{W}_1, W_2, \hat{W}_2, \left\{W'_{1,x}, W'_{2,x}\right\}_{x\in[\ell]}\right) \tag{9}$$

  where $W'_{1,x} = (W_{1,x})^{AG\tilde{K}_{\rho(x)}}, W'_{2,x} = (W_{2,x})^{AG\tilde{K}_{\rho(x)}}$.
- ***W_{CSP}.AfterReEncrypt***: W_{CSP} obtains the CT' from CSP, then use the stored $\{t_x\}_{x\in[\ell]} \in \mathbb{Z}_p$ to compute

$$\tilde{\mathrm{CT}}' = \left((A,\rho), \hat{W}, W_1, \hat{W}_1, W_2, \hat{W}_2, \left\{\tilde{W'_{1,x}}, \tilde{W'_{2,x}}\right\}_{x\in[\ell]}\right) \tag{10}$$

  where $\tilde{W'_{1,x}} = {W'_{1,x}}^{\frac{1}{t_x}}, \tilde{W'_{2,x}} = {W'_{2,x}}^{\frac{1}{t_x}}$.
- Decryption: Data user takes $\tilde{\mathrm{PK}}$, $\tilde{CT}'$, and $\tilde{sk_{id,\mathcal{S}}}$ as input to run *Decryption* to get R and M, if $\hat{W} = p^{H(M)}q^{H(R)}$, it returns M.
- Revocation: It does the same as algorithms *Revocation* in basic revocable CP-ABE.

6 Secure Analysis

6.1 Proof of RH-CPABE

Theorem 1. Suppose there exists the attacker $\mathcal{A}$ can break our method with non-negligible advantage e, then we can create an attacker $\mathcal{B}$ to break method [24].

Proof. We define attacker $\mathcal{A}$, challenger $\mathcal{B}$ of our scheme, also an attacker of scheme [24], $\mathcal{C}$ as an challenger of scheme [24].

1. Setup, $\mathcal{A}$ sends a access control policy to $\mathcal{B}$. Then $\mathcal{B}$ get $\mathrm{PK}_c = \left(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, e, g, g^b, e(g,g)^\beta\right)$ from $\mathcal{C}$ in scheme [24]. $\mathcal{B}$ selects p from G_{p_3} and q, r from G_{p_4}, it returns the public parameters $\mathrm{PK} = \left(N, \mathbb{G}, \mathbb{G}_T, e, g, g^b, e(g,g)^\beta, p, q, r\right)$ to the adversary $\mathcal{A}$ and key master key $MSK = \beta$ secret.
2. Query phase 1: $\mathcal{A}$ send some queries.
   (a) Secret keys query: $\mathcal{A}$ sends attributes sets $\mathcal{S}$ to $\mathcal{B}$ and $\mathcal{B}$ delivery it to $\mathcal{C}$, $\mathcal{C}$ launch key generation algorithm in [24] to produce $sk_{id,\mathcal{S}}$ and returns to $\mathcal{B}$, $\mathcal{B}$ then sends it to $\mathcal{A}$.
   (b) Decryption query: With ciphertext CT as input, $\mathcal{B}$ runs *Decrypt* to get message M, it then sends M to $\mathcal{A}$.

3. Challenge: $\mathcal{A}$ sends two messages M_0, M_1 and two access structure $\mathbb{A}_i : (A, \rho, \mathcal{T}_i)\,(i = 0, 1)$ to $\mathcal{B}$ and delivery to $\mathcal{C}$. $\mathcal{C}$ picks bit $\omega \overset{R}{\leftarrow} \{0, 1\}$, launch Encrypt $(PK, M_\omega, \mathbb{A}_\omega)$ and ReEncrypt (CT_ω, AGK) and give ciphertext CT'_ω back to $\mathcal{B}$. $\mathcal{B}$ sends to $\mathcal{A}$ where $\mathbb{A}_0$ or $\mathbb{A}_1$ cannot be satisfied by any set sub $(sub \subseteq \mathcal{S})$.
4. Query phase 2: $\mathcal{A}$ repeats Query phase 1, with restrictions that attribute set $\mathcal{S}'$ cannot satisfy $\mathbb{A}_0$ or $\mathbb{A}_1$ and $CT = CT'$.
5. Guess: $\mathcal{A}$ guesses bit $\omega' \in \{0, 1\}$. If $\omega' = \omega$, $\mathcal{A}$ wins the game.

Our scheme achieves selective security since it shares the same properties and security advantage as the method presented in [24].

6.2 Proof of RH-CPABE-CRF

For some damage caused by central authority, data owner, Cloud Server Provider, we utilize tampering algorithms $Setup^*$, $KeyGeneration^*$, $DataEncryption^*$, $DataReEncryption^*$ to verify selective CPA-secure through the indistinguishability of Rh-CPABE-CRF security game and Rh-CPABE scheme security game. In addition, the weak security resistance and weak resistance leakage of the reverse firewall can also be proved in this part:

Game 0: Its aligns with the security game in 3.1.

Game 1: Aligns with Game 0, but with the generation of PK and MSK occurs exclusively within the basic construction, as opposed to involving $Setup^*$ and $W_{TA}.Setup$.

Game 2: Aligns with Game 1, but with the difference that in the $KeyGeneration$ phase, the generation of SK is carried out solely by the $KeyGeneration$ process in the basic construction, rather than involving $KeyGeneration^*$ and $W_{TA}.KeyGen$.

Game 3. Aligns with Game 2, but with CT generated by the basic construction's $DataEncryption$, not $DataEncryption^*$ and $W_{DO}.Encrypt$.

Game 4. Aligns with Game 3, but with CT' are generated by $DataReEncryption$ in the basic construction, not $W_{CSP}.PreReEncrypt$ and $W_{CSP}.AfterReEncrypt$.

We prove indistinguishability between Game 0 and Game 1, Game 1 and Game 2, Game 2 and Game 3, and Game 3 and Game 4. After using the reverse firewall $W_{TA}.Setup^*$ on any tampered algorithm, the public parameters PK remain uniformly random, preserving the original algorithm's behavior and ensuring security.

Game 0 and Game 1 are indistinguishable. Game 1 and Game 2 are indistinguishable because user secret key also have key malleability. For the pair Game 2 and Game 3, for any tampered algorithm $Dataencryption^*$, after the post-processing of $W_{DO}.Encrypt$. The updated ciphertext CT are uniformly regenerated, which is smae as encryption algorithm in the basic construction. And also, after pre-processing by $W_{CSP}.PreReEncrypt$ and $W_{CSP}.AfterReEncrypt$. Game 3 and Game 4 are indistinguishable since the updated ciphertext CT' are uniformly reproduced. Thus, we can deduce that Game 0 and Game 4 are indistinguishable. As the basic construction ensures selective CPA-security, the proposed Rh-CPABE-CRF scheme also achieves selective CPA-security.

The selective CPA security implies that the reverse firewalls maintain weakly preserved security. Additionally, the indistinguishability between Game 0 and Game 4 demonstrates their effectiveness in weakly resisting exfiltration attempts. This completes the proof of the proposed scheme's security.

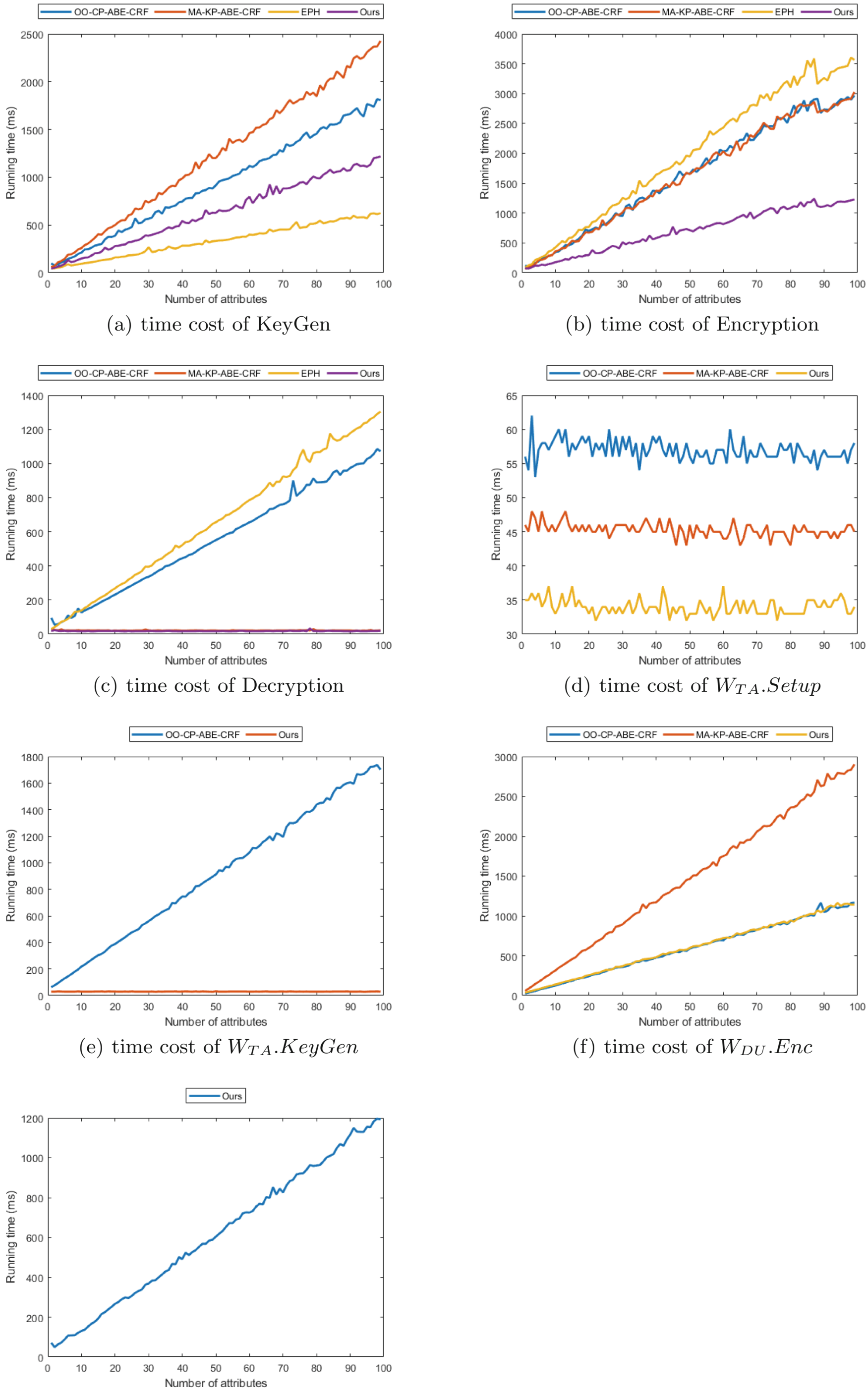
7 Performance

In Table 1, the primary procedures of our scheme are juxtaposed with those of other methods for comparison. Our scheme proposes an encryption-based access control scheme that addresses the critical issue of data sharing and protection. In comparison with [11,12,14], our scheme exhibits several notable advantages. Firstly, it supports reverse firewall, which provides more reliable protection against internal attack. Secondly, it supports attribute revocation, enabling attributes to be revoked when they are no longer needed, ensuring data controllability and security. Thirdly, it supports partial policy hiding, which enables data owners to protect data privacy and confidentiality by hiding part of the access policy. Fourthly, it supports large universe attributes, which enables a wider range of attributes to be used, thereby increasing system flexibility and scalability. Finally, our scheme uses a secret sharing scheme based on LSSS for access control, which is more efficient and flexible compared to the Tree structure used in [11].

In contrast, [11,14] only support reverse firewall with no support for attribute revocation, partial policy hiding, or large universe attributes. Similarly, paper [12] does not support reverse firewall or attribute revocation, but it does support attribute hiding and uses LSSS for access control. Thus, the proposed access control scheme in this paper offers superior support and security compared to the three comparison papers, making it a significant contribution to the field of data sharing and protection.

Regarding time efficiency, Table 2 indicates that our scheme outperforms other solutions at encryption and decryption. Specifically, in terms of key generation, our scheme ranks in the middle. Regarding CRFs' performances, our scheme demonstrates superior efficiency compared to all the solutions listed in Table 2. In the $W_{TA}Setup$ setup, three schemes from Table 2 remain unchanged with an increase in the number of attributes, whereas our scheme outperforms others during the $W_{TA}KeyGen$ and $W_{DO}Enc$ procedures.

We evaluated the efficiency of our proposed schema using the JPBC library and conducted experiments on Windows 10 operating system with an i7-11700 2.50GHz CPU. The performance of our scheme is shown in Fig. 3, while Figs. 3(a), 3(b), and 3(c) display the performance of KeyGen, Encryption and Decryption. Additionally, Figs. 3(d), 3(e), 3(f), and 3(g) demonstrate time cost of CRFs. The experimental outcomes consistently corroborate the efficiency comparison table, as depicted in Table 2.



(a) time cost of KeyGen

(b) time cost of Encryption

(c) time cost of Decryption

(d) time cost of $W_{TA}.Setup$

(e) time cost of $W_{TA}.KeyGen$

(f) time cost of $W_{DU}.Enc$

(g) time cost of $W_{CSP}.AfterPreEnc$

Fig. 3. Time cost of Rh-CPABE-CRF

Table 1. Function comparison

	OOCAC [14]	AKAC [11]	EPH [12]	Ours
With CRFs	✓	✓	×	✓
Revocable	×	×	×	✓
Policy hiding	×	×	✓	✓
Large Universe	×	×	×	✓
Access structure	LSSS	LSSS	Tree	LSSS

Table 2. Efficiency comparison

	OOCAC [14]	AKAC [11]	EPH [12]	Ours
$KeyGen$	$(3l+4)E$	$(4l+2)E+P$	$(l+4)E$	$(2l+3)E$
$Encryption$	$(5l+2)E$	$(5l+2)E$	$(6l+4)E$	$(2l+6)E$
$Decryption$	$E+(3l+1)P$	$E+P$	$lE+2lP$	E
$W_{TA}.Setup$	$7E+P$	$5E+P$	×	$3E+P$
$W_{TA}.KeyGen$	$(3k+5)E$	×	×	$3E$
$W_{DO}.Enc$	$2lE$	$(5l+2)E+P$	×	$(2l+2)E$
$W_{CSP}.AfterReEnc$	×	×	×	$2lE$

E stands for modular exponentiation. P represents a bilinear pairing. l indicates the number of attributes associated with the user. k denotes the count of attributes in the access structure that fulfill the decryption requirement.

8 Conclusion

This paper proposes an ABE scheme with revocable attributes and hidden policy, enhanced with a reverse firewall. Our scheme addresses the confidentiality and integrity challenges of existing ABE schemes, including efficient attribute revocation and protection of the access control policy.

We evaluated the properties of our scheme and demonstrated its advantages over existing ABE with CRFs schemes. The experiments shows that our scheme achieves strong security while maintaining good performance.

References

1. Beimel, A., et al.: Secure schemes for secret sharing and key distribution (1996)
2. Bellare, M., Paterson, K.G., Rogaway, P.: Security of symmetric encryption against mass surveillance. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 1–19. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_1
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP 2007), pp. 321–334. IEEE (2007)

4. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
5. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_16
6. Chen, R., Mu, Y., Yang, G., Susilo, W., Guo, F., Zhang, M.: Cryptographic reverse firewall via malleable smooth projective hash functions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 844–876. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_31
7. Cheung, L., Cooley, J.A., Khazan, R., Newport, C.: Collusion-resistant group key management using attribute-based encryption. Cryptology ePrint Archive (2007)
8. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 456–465 (2007)
9. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
10. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of {ABE} ciphertexts. In: 20th USENIX Security Symposium (USENIX Security 11) (2011)
11. Hong, B., Chen, J., Zhang, K., Qian, F.H.: Multi-authority non-monotonic KP-ABE with cryptographic reverse firewall. IEEE Access **7**, 159002–159012 (2019)
12. Lai, J., Deng, R.H., Li, Y.: Expressive CP-ABE with partially hidden access structures. In: ACM Asia Conference on Computer and Communications Security (2012)
13. Liu, Z., Jiang, Z.L., Wang, X., Yiu, S.M.: Practical attribute-based encryption: outsourcing decryption, attribute revocation and policy updating. J. Netw. Comput. Appl. **108**, 112–123 (2018)
14. Ma, H., Zhang, R., Yang, G., Song, Z., Sun, S., Xiao, Y.: Concessive online/offline attribute based encryption with cryptographic reverse firewalls—secure and efficient fine-grained access control on corrupted machines. In: Lopez, J., Zhou, J., Soriano, M. (eds.) ESORICS 2018. LNCS, vol. 11099, pp. 507–526. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98989-1_25
15. Mironov, I., Stephens-Davidowitz, N.: Cryptographic reverse firewalls. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 657–686. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_22
16. Ouyang, M., Wang, Z., Li, F.: Digital signature with cryptographic reverse firewalls. J. Syst. Architect. **116**, 102029 (2021)
17. Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 99–112 (2006)
18. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
19. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
20. Wang, W., Zhang, G., Shen, Y.: A CP-ABE scheme supporting attribute revocation and policy hiding in outsourced environment. In: 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), pp. 96–99. IEEE (2018)

21. Xiong, H., Zhou, Z., Wang, L., Zhao, Z., Huang, X., Zhang, H.: An anonymous authentication protocol with delegation and revocation for content delivery networks. IEEE Syst. J. **16**(3), 4118–4129 (2021)
22. Yu, S., Ren, K., Lou, W.: Attribute-based content distribution with hidden policy. In: 2008 4th Workshop on Secure Network Protocols, pp. 39–44. IEEE (2008)
23. Yu, S., Ren, K., Lou, W.: Attribute-based on-demand multicast group setup with membership anonymity. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, pp. 1–6 (2008)
24. Zeng, P., Zhang, Z., Lu, R., Choo, K.K.R.: Efficient policy-hiding and large universe attribute-based encryption with public traceability for internet of medical things. IEEE Internet Things J. **8**(13), 10963–10972 (2021)
25. Zhang, Y., Zheng, D., Deng, R.H.: Security and privacy in smart health: efficient policy-hiding attribute-based access control. IEEE Internet Things J. **5**(3), 2130–2145 (2018)
26. Zhou, Y., Guo, J., Li, F.: Certificateless public key encryption with cryptographic reverse firewalls. J. Syst. Architect. **109**, 101754 (2020)