Concessive Online/Offline Attribute Based Encryption with Cryptographic Reverse Firewalls—Secure and Efficient Fine-Grained Access Control on Corrupted Machines

Hui Ma[1], Rui Zhang[1,3(✉)], Guomin Yang[2], Zishuai Song[1,3], Shuzhou Sun[1,3], and Yuting Xiao[1,3]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
r-zhang@iie.ac.cn
[2] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. Attribute based encryption (ABE) has potential to be applied in various cloud computing applications. However, the Snowden revelations show that powerful adversaries can corrupt users' machines to compromise the security, and many implementations of provably secure encryption schemes may present undetectable vulnerabilities that can expose secret, e.g., the scheme still works properly even some backdoors have been stealthily engineered on users' machines. Undoubtedly, ABE is also facing the above security threats. Recently, Mironov and Stephens-Davidowitz proposed cryptographic reverse firewall (CRF) to solve the problem. Unfortunately, no CRF-based protection for ABE has been proposed so far due to the complex system model and the extra access structure component. Besides, the encryption scheme in the CRF framework will suffer double computation latency, which is worse for ABE that has already yielded expensive operations. In this paper, we propose a concessive online/offline ciphertext-policy attribute based encryption with cryptographic reverse firewalls (COO-CP-ABE-CRF), which can resist the exfiltration of secret information and achieve selective CPA security. Furthermore, compared with the original scheme without CRF, our scheme reduces the total computation cost by half. Moreover, we develop an extensible library called libabe that is compatible with Android devices, and we implement the prototype on a laptop and a mobile phone. The experimental results indicate that the scheme is efficient and practical.

J. Lopez et al. (Eds.): ESORICS 2018, LNCS 11099, pp. 507–526, 2018.
https://doi.org/10.1007/978-3-319-98989-1_25

1 Introduction

As an innovative cryptographic primitive, attribute based encryption (ABE), that can provide fine-grained access control over encrypted data, has potential to be applied in many cloud-assisted applications, such as Pay TV/Music [1,2], Electronic Medical Record [3,4], audit logs [5,6], and web services [7,8] etc.

However, in the last couple of years, it has become increasingly clear that the practical cryptographic implementation presents many vulnerabilities even the protocol has been proved to be secure in theory. The revelations of Edward Snowden show that powerful actors have remarkable ability to successfully obtain a massive secret information by extraordinary techniques, including embedding backdoors into the public cryptographic standard [9,10] and the pseudorandom generator [11,12], intercepting and tampering with users' hardware deliveries [13]. Meanwhile, many security flaws [14–17] have been reported in widely deployed implementations of cryptographic softwares, which will certainly lead to large-scale security risks. The vulnerabilities of cryptographic implementations are extremely hard to detect in practice, because the implemented protocol still works properly even backdoors have been stealthily engineered without user's knowledge. Unfortunately, ABE is also facing the disturbing and quite real possibility of the above-mentioned compromises, e.g., the adversary tampers the setup algorithm on the private key generator (PKG) to generate some special but functional-maintaining public parameters which can expose the system master secret key or it embeds backdoors into the pseudorandom generator on PKG. This intractable situation motivates us to strengthen the security of ABE when the adversary may arbitrarily tamper with the victim's machine.

Recently, Mironov and Stephens-Davidowitz [18] proposed an innovative concept called cryptographic reverse firewall (CRF) that can strengthen the security to resist inside vulnerabilities such as security backdoors. Informally, a CRF implemented on a trust machine is located between the user's machine and the outside world and is able to intercept and modify the machine's incoming and outgoing messages to provide security protections even if the user's machine has been tampered. Though several CRF-based protections have been proposed for message-transmission protocol [19], key-agreement protocol [19], oblivious transfer protocols [18,20], oblivious signature-based envelope [20] etc., no CRF-based protection for ABE has been proposed so far. To strengthen the security of ABE by applying CRF-based protection, there exist the following serious challenges.

– Dodis et al. [19] proposed that the encryption scheme in the CRF framework should be both key malleable and strongly rerandomizable. While the situation in ABE is somewhat complicated: (1) Unlike the simple system with CRF (e.g., ElGamal), more entities are involved in ABE system and the communication becomes complex, thus the system model should be creatively redesigned to adapt CRF. (2) Since ABE utilizes various access structures to achieve fine-grained access control, it needs careful consideration that whether the property of extra access structure component matches the CRF framework. Therefore, the first challenge is that *how to design ABE with CRF-based protection to resist the exfiltration of secret information?*

– The encryption scheme with CRF always suffers double computation latency due to the rerandomization, which is even worse for ABE that has already yielded the heavy computation cost, largely the pairing and exponentiation operations, which often grow with the complexity of access formula. This is a huge burden for the private key generator (PKG) and users, especially for resource-constrained mobile devices. Therefore, the second challenge is that *how to improve the computation efficiency of ABE in the CRF framework?*

1.1 Our Contribution

Aiming at solving above challenges, we propose a concessive online/offline ciphertext-policy attribute based encryption with cryptographic reverse firewalls (COO-CP-ABE-CRF), which not only resists the exfiltration of secret information from arbitrarily compromised functional-maintaining algorithms, but also improves the computation efficiency of all the algorithms significantly and is suitable for mobile devices. Our contribution is three-fold:

– **Exfiltration Resistance.** We first propose a new system model for ABE in the CRF framework (cf. Fig. 2), where three reverse firewalls are adopted for the PKG, data owner and data consumer respectively. Then, we present a detailed construction, where all the random parameters including the parameters in the LSSS access structure are rerandomized by three reverse firewalls to achieve exfiltration resistance.
– **High Computation Efficiency.** We propose a concessive online/offline attribute based encryption with cryptographic reverse firewalls, where the online computation efficiency of the entities including PKG, data owner, data consumer and reverse firewalls for the PKG and data owner is optimized significantly. Compared with the original scheme without CRF [21], our CRF-based scheme reduces the total computation cost of key generation[1] by half and improves the total computation efficiency of encryption. The data consumer only needs 1 exponentiation to complete the decryption.
– **Compatible Implementation.** We develop an extensible library libabe that is compatible with Android OS. We implement a prototype within libabe on a laptop and a mobile phone. The results indicate the high efficiency and practicability of our methodology. We believe that this library can make ABE a step closer to actual deployment with mobile devices.

1.2 Related Work

Attribute based encryption (ABE) was first introduced by Sahai and Waters under the name fuzzy identity-based encryption [22]. Goyal et al. [6] extended fuzzy IBE to ABE. Up to now, there are two forms of ABE: key-policy ABE (KP-ABE) [6,23–25], where the key is assigned to an access policy and the ciphertext to a set of attributes, and ciphertext-policy ABE (CP-ABE) [26–28], where the

[1] It is the total workload of the PKG and the reverse firewall for the PKG.

ciphertext is assigned to an access policy and the key to a set of attributes. A user can decrypt a ciphertext if the set of attributes satisfies the access policy.

Cryptographic reverse firewall (CRF) was first introduced by Mironov and Stephens-Davidowitz [18], they proposed the CRF-based protection for oblivious protocol and presented a generic construction to protect users from data leakage against eavesdroppers via any protocol. Dodis, Mironov and Stephens-Davidowitz [19] considered message transmission protocols in the CRF framework. They proposed a rich collection of solutions in different settings which vary in efficiency, security, and setup assumptions. Moreover, they proposed a generic framework for constructing two-round protocol from rerandomizable encryption schemes. Chen et al. [20] introduced the notion of malleable smooth projective hash function (SPHF) and showed that how to construct CRFs using malleable SPHAs for some widely used cryptographic protocols. However, all the above CRF-based protections are not suitable for ABE due to the more complex system model and the extra components adopted in ABE construction.

2 Preliminary

In this section, we review some definitions of attribute based encryption and cryptographic reverse firewalls.

2.1 Attribute Based Encryption

Definition 1 (Bilinear Groups). *Let $\mathbb{G}, \mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map with the following properties: (1) Bilinearity: for all $g, h \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, we have $e(g^a, h^b) = e(g, h)^{ab}$. (2) Nondegeneracy: $e(g, h) \neq 1$ whenever $g, h \neq 1_{\mathbb{G}}$.*

Definition 2 (Access Structure [29]). *Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, \ldots, P_n\}}$ is monotone for $\forall B$ and C, if $B \in \mathbb{A}, B \subseteq C$, then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) of nonempty subsets of $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, \ldots, P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.*

Definition 3 (Linear Secret Sharing Schemes (LSSS) [29]). *A secret sharing scheme Π over a set of parties is called linear over $\mathbb{Z}_p$ if (1) The shares of the parties form a vector over $\mathbb{Z}_p$; (2) There exists a matrix M with l rows and n columns called the share-generating matrix for Π. There exists a function ρ which maps each row of the matrix to an associated party, i.e., for $i = 1, \ldots, l$, the value $\rho(i)$ is the party associated with row i. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then Mv is the vector of l shares of the secret s according to Π. The share $(Mv)_i$ belongs to party $\rho(i)$.*

2.2 Cryptographic Reverse Firewalls

We review the definitions of reverse firewall introduced in [18,20]. We assume that a cryptographic scheme $\mathcal{E}$ satisfies functionality requirement $\mathcal{F}$ and security requirement $\mathcal{S}$. There are two kinds of reverse firewalls, one can access all the public parameters and messages but not the private input or output of entities, another one can access all the public and private input and output of entities.

Definition 4 (Cryptographic Reverse firewall (CRF)). *A cryptographic reverse firewall is a stateful algorithm $\mathcal{W}$ that takes as input its state and a message and outputs an updated state and message. For simplicity, we do not write the state of $\mathcal{W}$ explicitly. For a party P and reverse firewall $\mathcal{W}$, we define $\mathcal{W} \circ P$ as the composed party where $\mathcal{W}$ is applied to the incoming and outgoing messages of P. When the composed party engages in a protocol, the state of $\mathcal{W}$ is initialized to the public parameters. If $\mathcal{W}$ is meant to be composed with a party P, we call it a reverse firewall for P.*

Definition 5 (Functionality-maintaining CRFs). *For any reverse firewall $\mathcal{W}$ and any party P, let $\mathcal{W}^1 \circ P = \mathcal{W} \circ P$, for $k \geq 2$, let $\mathcal{W}^k \circ P = \mathcal{W} \circ (\mathcal{W}^{k-1} \circ P)$. For a scheme $\mathcal{E}$ that satisfies functionality requirement $\mathcal{F}$, we say a reverse firewall $\mathcal{W}$ maintains $\mathcal{F}$ for P in $\mathcal{E}$ if $\mathcal{W}^k \circ P$ maintains $\mathcal{F}$ for P in $\mathcal{E}$ for any polynomial bounded $k \geq 1$. When $\mathcal{F}, P, \mathcal{E}$ are clear, we say $\mathcal{W}$ maintains functionality.*

We use $\hat{P}$ to represent the functionality-maintaining adversarial implementations. For a scheme $\mathcal{E}$ with party P, we write $\mathcal{E}_{P \to \hat{P}}$ to represent the scheme where the role of party P is replaced by party $\hat{P}$.

Definition 6 (Weakly security-preserving CRFs). *For a scheme $\mathcal{E}$ that satisfies security requirement $\mathcal{S}$ and functionality $\mathcal{F}$ and a reverse firewall $\mathcal{W}$, $\mathcal{W}$ weakly preserves $\mathcal{S}$ for P in $\mathcal{E}$ if the scheme $\mathcal{E}_{P \to \mathcal{W} \circ \hat{P}}$ satisfies $\mathcal{S}$. When $\mathcal{E}, \mathcal{F}, \mathcal{S}, P$ are clear, we say that $\mathcal{W}$ weakly preserves security.*

A reverse firewall should also achieve weakly exfiltration resistance which means that no corrupted functionality-maintaining implementation of P can leak information through the firewall. We define a game LEAK that is presented in Fig. 1. The game asks the adversary to distinguish between a tampered implementation and an honest implementation. An exfiltration-resistant reverse firewall therefore prevents an adversary from even learning whether a party has been compromised, let alone leaking information.

Definition 7 (Weakly exfiltration-resistant CRFs). *For a scheme $\mathcal{E}$ that satisfies functionality $\mathcal{F}$ and a reverse firewall $\mathcal{W}$, we say $\mathcal{W}$ is weakly exfiltration-resistant for party P_1 against party P_2 in scheme $\mathcal{E}$, if for any PPT adversary $\mathcal{A}$, $Adv^{\mathsf{LEAK}}_{\mathcal{A},\mathcal{W}}(l) = Pr[\mathsf{LEAK}(\mathcal{E}, P_1, P_2, \mathcal{W}, l) = 1] - \frac{1}{2}$ is negligible[2] in the security parameter l provided that $\overline{P}_1$ maintains $\mathcal{F}$ for P_1.*

[2] A function f is negligible if for every $c > 0$ there exists $\lambda_0 > 0$ such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_0$.

$\mathbf{Proc.}\mathsf{LEAK}(\mathcal{E}, P_1, P_2, \mathcal{W}, l)$
$(\overline{P_1}, \overline{P_2}, I) \rightarrow \mathcal{A}(1^l)$
$b \xleftarrow{\$} \{0,1\}$
If $b = 1$, $P^* \leftarrow \mathcal{W} \circ \overline{P_1}$
Else, $P^* \leftarrow \mathcal{W} \circ P_1$
$\mathcal{T}^* \leftarrow \mathcal{E}_{P_1 \rightarrow P^*, P_2 \rightarrow \overline{P}_2}(I)$
$b^* \leftarrow \mathcal{A}(\mathcal{T}^*, st_{\overline{P}_2})$
Output $(b = b^*)$

Fig. 1. The exfiltration resistance security game for a reverse firewall $\mathcal{W}$ for party P_1 in scheme $\mathcal{E}$ against party P_2. $\mathcal{A}$ is the adversary, l the security parameter, $st_{\overline{P}_2}$ the state of $\overline{P}_2$ after the run of the scheme, I valid input for $\mathcal{E}$, and $\mathcal{T}^*$ is the transcript of running scheme $\mathcal{E}_{P_1 \rightarrow P^*, P_2 \rightarrow \overline{P}_2}(I)$.

3 System Model and Security Model

3.1 System Model

As illustrated in Fig. 2, four different entities are involved in our system: the private key generator (PKG), the public cloud, the data owner (DO) and the data consumer (DC). Moreover, three reverse firewalls are adopted. $\mathcal{W}_{\mathbf{PKG}}$ is the reverse firewall for PKG, $\mathcal{W}_{\mathbf{DO}}$ is the reverse firewall for the data owner and $\mathcal{W}_{\mathbf{DC}}$ is the reverse firewall for the data consumer.

PKG is responsible to generate public parameters and the master secret key.
Data Owner defines access policies and encrypts data under these policies before uploading them to the public cloud.
Public Cloud is deployed to provide cloud data storage service and outsourced decryption service. Users can upload and download the cloud file.
Data Consumer can download any encrypted data of his/her interest from public cloud and try to decrypt the ciphertext.
$\mathcal{W}_{\mathbf{PKG}}$ is responsible to rerandomize public parameters and users' secret keys in case that the setup and key generation algorithms of PKG are compromised.
$\mathcal{W}_{\mathbf{DO}}$ is responsible to rerandomize the ciphertexts generated by the data owner in case that the encryption algorithm of the data owner is compromised.
$\mathcal{W}_{\mathbf{DC}}$ is responsible to rerandomize the conversion key generated by the data consumer in case that the conversion key[3] generation algorithm of the data consumer is compromised.

Let S represent a set of attributes, and (M, ρ) be an access structure. The concessive online/offline ciphertext-policy attribute based encryption with cryptographic reverse firewalls (COO-CP-ABE-CRF) for access structure space $\mathcal{G}$ consists of 15 algorithms:

[3] The public cloud can use it to do outsourced decryption.

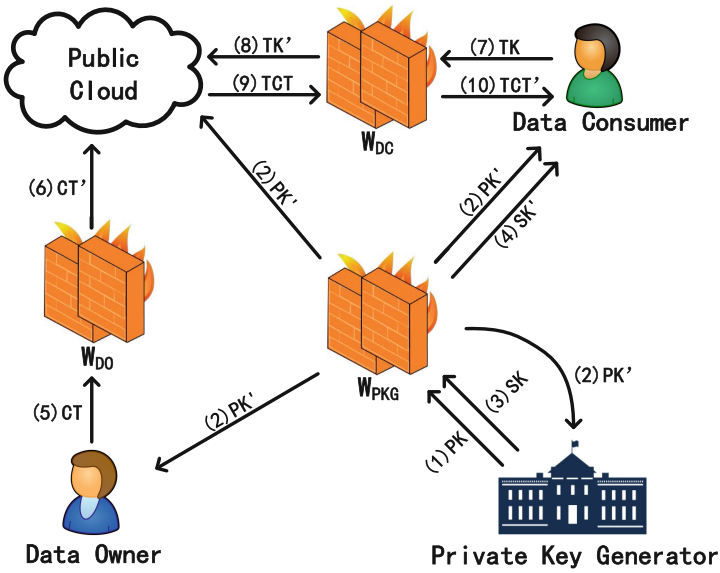

Fig. 2. System model of COO-CP-ABE-CRF

Setup$(\lambda, U) \rightarrow (PK, MSK)$. It is performed by the PKG. On input a security parameter λ and a universe description U, it outputs public parameters PK and a master secret key MSK.
$\mathcal{W}_{\mathbf{PKG}}$**.Setup**$(PK) \rightarrow (PK', f)$. It is performed by the reverse firewall for PKG $\mathcal{W}_{\mathbf{PKG}}$. On input public parameters PK, it outputs updated public parameters PK' and a corresponding random f.
KeyGen.offline$(PK', MSK, N) \rightarrow ISK$. It is performed by PKG. On input updated public parameters PK', a master secret key MSK and a number N that assumes the attribute number, it outputs an intermediate secret key ISK.
KeyGen.online$(PK', S, ISK) \rightarrow SK$. It is performed by the PKG. On input updated public parameters PK', an attribute set S and an intermediate secret key ISK, it outputs a secret key SK.
$\mathcal{W}_{\mathbf{PKG}}$**.KG.offline**$(PK', f, N) \rightarrow ISK'$. It is performed by the reverse firewall for PKG $\mathcal{W}_{\mathbf{PKG}}$. On input the updated public parameters PK', a random f and a number N that assumes the attribute number, it outputs an updated intermediate secret key ISK'.
$\mathcal{W}_{\mathbf{PKG}}$**.KG.online**$(PK', ISK', SK) \rightarrow SK'$. It is performed by the reverse firewall for PKG $\mathcal{W}_{\mathbf{PKG}}$. On input the updated public parameters PK', an updated intermediate secret key ISK' and a secret key SK, it outputs an updated SK'.
Encrypt.offline$(PK', N') \rightarrow IT$. It is performed by the data owner. On input the updated public parameters PK' and a number N' that assumes a maximum bound of N' rows in LSSS structure, it outputs an intermediate ciphertext IT.

Encrypt.online$(PK', IT, m, (M, \rho)) \rightarrow CT$. It is performed by the data owner. On input updated public parameters PK', an intermediate ciphertext IT, a plaintext m and an LSSS access structure (M, ρ), it outputs a ciphertext CT.
$\mathcal{W}_{\mathbf{DO}}$**.Enc.offline**$(PK', N') \rightarrow IT'$. It is performed by the reverse firewall for data owner $\mathcal{W}_{\mathbf{DO}}$. On input updated public parameters PK' and a number N' that assumes a maximum bound of N' rows in any LSSS access structure, it outputs an updated intermediate ciphertext IT'.
$\mathcal{W}_{\mathbf{DO}}$**.Enc.online**$(PK', IT', CT) \rightarrow CT'$. It is performed by the reverse firewall for data owner $\mathcal{W}_{\mathbf{DO}}$. On input updated public parameters PK', an updated intermediate ciphertext IT' and a ciphertext CT, it outputs an updated CT'.
KeyGen.ran$(SK') \rightarrow (TK, RK)$. It is performed by the data consumer. On input an updated secret key SK', it outputs a conversion key TK and the corresponding retrieval key RK.
$\mathcal{W}_{\mathbf{DC}}$**.TKUpdate**$(TK) \rightarrow (TK', \beta)$. It is performed by the reverse firewall for data consumer $\mathcal{W}_{\mathbf{DC}}$. On input a conversion key TK, it outputs an updated conversion key TK' and a corresponding random β.
Decrypt.out$(TK', CT') \rightarrow TCT$ or $\perp$. It is performed by the public cloud. On input an updated conversion key TK' and an updated ciphertext CT', it outputs the transformed ciphertext TCT or $\perp$.
$\mathcal{W}_{\mathbf{DC}}$**.Decrypt**$(TCT, \beta) \rightarrow TCT'$. It is performed by the reverse firewall for data consumer $\mathcal{W}_{\mathbf{DC}}$. On input a transformed ciphertext TCT and a random β, it outputs an updated transformed ciphertext TCT'.
Decrypt.user$(RK, TCT') \rightarrow m$. It is performed by the data consumer. On input a retrieval key RK and an updated transformed ciphertext TCT', it outputs the plaintext m.

Correctness. For the fixed universe description U, the security parameter $\lambda \in \mathbb{N}$, the access structure space $\mathcal{G}$ and the message m, the correctness property requires that for all $(PK, MSK) \in \mathit{Setup}(\lambda, U)$, all $(PK', f) \in \mathcal{W}_{\mathbf{PKG}}.\mathit{Setup}(PK)$, all $S \subseteq U$, all $(M, \rho) \in \mathcal{G}$, all $ISK \in \mathit{KeyGen.offline}(PK', MSK, N)$, all $SK \in \mathit{KeyGen.online}(PK', S, ISK)$, all $ISK' \in \mathcal{W}_{\mathbf{PKG}}.\mathit{KG.offline}(PK', f, N)$, all $SK' \in \mathcal{W}_{\mathbf{PKG}}.\mathit{KG.online}(PK', ISK', SK)$, all $IT \in \mathit{Encrypt.offline}(PK', N')$, all $CT \in \mathit{Encrypt.online}(PK', IT, m, (M, \rho))$, all $IT' \in \mathcal{W}_{\mathbf{DO}}.\mathit{Enc.offline}(PK', N')$, all $CT' \in \mathcal{W}_{\mathbf{DO}}.\mathit{Enc.online}(PK', IT', CT)$, all $(TK, RK) \in \mathit{KeyGen.ran}(SK')$, all $(TK', \beta) \in \mathcal{W}_{\mathbf{DC}}.\mathit{TKUpdate}(TK)$, all $TCT \in \mathit{Decrypt.out}(TK', CT')$, all $TCT' \in \mathcal{W}_{\mathbf{DC}}.\mathit{Decrypt}(TCT, \beta)$, if S satisfies (M, ρ), $\mathit{Decrypt.user}(RK, TCT') \rightarrow m$.

3.2 Security Model

Adversarial Model. In the system, the PKG, the data owner and the data consumer are totally trusted, but the *Setup, KeyGen.offline, KeyGen.online* algorithms run by the PKG, the *Encrypt.offline, Encrypt.online* algorithms run by the data owner and the *KeyGen.ran* algorithm run by the data consumer may be stealthily compromised without the executors' knowledge, because the algorithms will maintain the functionality even malicious backdoors have already

been implanted. The public cloud and the reverse firewalls $\mathcal{W}_{\mathbf{DO}}, \mathcal{W}_{\mathbf{DC}}$ are "honest but curious" [30,31]. More precisely, they will follow the protocol but try to find out as much private information as possible. While the reverse firewall $\mathcal{W}_{\mathbf{PKG}}$ should be totally trusted because it has to access every user's secret key. Moreover, all the reverse firewalls are considered as the trust zones that will not be tampered by any outsiders. Next, we will introduce the selective CPA security Game for COO-CP-ABE-CRF.

Init. The adversary $\mathcal{A}$ sends the challenge access policy $\mathbb{A}^*$ and the functionality-maintaining algorithms $Setup^*$, $KeyGen.offline^*$, $KeyGen.online^*$, $KeyGen.ran^*$, $Encrypt.offline^*$, $Encrypt.online^*$ to the challenger $\mathcal{C}$.

Setup. $\mathcal{C}$ runs $Setup^*$ to get PK, MSK, then runs $\mathcal{W}_{\mathbf{PKG}}.Setup(PK)$ to get the updated PK' and the corresponding random f. $\mathcal{C}$ keeps MSK and f to itself and sends PK' to $\mathcal{A}$.

Phase 1. In this phase, $\mathcal{A}$ can adaptively ask for secret keys for attribute sets $S_1, S_2, \ldots, S_q$. For each query S_i, $\mathcal{C}$ calls $KeyGen.offline^*(PK', MSK, N) \rightarrow ISK$, $KeyGen.online^*(PK', S_i, ISK) \rightarrow SK_i$, then runs $\mathcal{W}_{\mathbf{PKG}}.KG.offline(PK', f, N) \rightarrow ISK'_i, \mathcal{W}_{\mathbf{PKG}}.KG.online(PK', ISK'_i, SK_i) \rightarrow SK'_i$. Next, $\mathcal{C}$ calls $KeyGen.ran^*(SK'_i) \rightarrow TK_i$ and runs $\mathcal{W}_{\mathbf{DC}}.TKUpdate$ to get TK'_i. At last, $\mathcal{C}$ sends (SK'_i, TK'_i) to $\mathcal{A}$. The restriction that has to be satisfied for each query is that none of the queried attribute sets satisfies the challenge policy.

Challenge. $\mathcal{A}$ sends two equal-length plaintexts m_0, m_1 to $\mathcal{C}$. $\mathcal{C}$ selects a random bit $b \in \{0,1\}$ and runs $Encrypt.offline^*(PK', N') \rightarrow IT, Encrypt.online^*(PK', IT, m_b, \mathbb{A}^*)$ to obtain CT_b. Then $\mathcal{C}$ calls $\mathcal{W}_{\mathbf{DO}}.Enc.offline(PK', N') \rightarrow IT'$, $\mathcal{W}_{\mathbf{DO}}.Enc.online(PK', IT', CT_b) \rightarrow CT'_b$ and sends CT'_b to $\mathcal{A}$.

Phase 2. Same as Phase 1.

Guess. $\mathcal{A}$ outputs the guess $b' \in \{0,1\}$ for b.

Definition 8. *A COO-CP-ABE-CRF scheme is selective CPA-secure if all probabilistic polynomial time (PPT) adversaries have at most a negligible advantage in the above security game, denote:*

$$\epsilon = |\Pr[b = b'] - \frac{1}{2}| \leq negl(\lambda).$$

4 Concessive Online/Offline CP-ABE with Cryptographic Reverse Firewalls

In this section, we first present a basic construction of concessive online/offline ciphertext-policy attribute based encryption (COO-CP-ABE) which is based on the Rouselakis and Waters's CP-ABE scheme in [21]. Then, we propose the construction of COO-CP-ABE with cryptographic reverse firewalls (COO-CP-ABE-CRF) and give the security proof in the standard model.

4.1 Basic Construction of Concessive Online/Offline CP-ABE

Technique Overview. To improve the computation efficiency, we propose the basic construction of concessive online/offline CP-ABE. For key generation and encryption, if we directly utilize the "connect and correct" technique in online/offline ABE [32], the randomness of ciphertexts and secret keys cannot be all rerandomized in the CRF framework. Thus, we propose the concessive version, which is suitable for the CRF framework but sacrifices a small amount of efficiency. Now, we will introduce the construction.

Setup(λ, U). The PKG chooses a bilinear map $D = (\mathbb{G}, \mathbb{G}_T, e, p)$, where $p \in \Theta(2^\lambda)$ is the prime order of the groups $\mathbb{G}$ and $\mathbb{G}_T$. The attribute universe is consisting of elements in $\mathbb{Z}_p$. It chooses random generators $g, u, h, w, v \in \mathbb{G}$ and picks a random $\alpha \in \mathbb{Z}_p$. It sets the keys as $PK = (D, g, u, h, w, v, e(g,g)^\alpha)$, $MSK = \alpha$.
KeyGen.offline(PK, MSK, N). On input public parameters PK, a master secret key MSK and a number N which assumes the number of attributes, the PKG picks $N+1$ random $r, r_1, r_2, \ldots, r_N \in \mathbb{Z}_p$ and computes $\hat{K_0} = g^\alpha w^r, \hat{K_1} = g^r$. Then for $i = 1$ to N, it computes $\hat{K_{i,2}} = g^{r_i}$, $\hat{K_{i,3}} = h^{r_i} v^{-r}$. It sets the intermediate secret key $ISK = (\hat{K_0}, \hat{K_1}, \{r_i, \hat{K_{i,2}}, \hat{K_{i,3}}\}_{i\in[1,N]})$.
KeyGen.online(PK, S, ISK). On input public parameters PK, an attribute set $S = \{A_1, A_2, \ldots, A_k\} \subseteq \mathbb{Z}_p$ where $k \leq N$ and an intermediate secret key ISK, the PKG sets $K_0 = \hat{K_0} = g^\alpha w^r$, $K_1 = \hat{K_1} = g^r$. Then for $i = 1$ to k, it sets and computes $K_{i,2} = \hat{K_{i,2}} = g^{r_i}$, $K_{i,3} = \hat{K_{i,3}} \cdot u^{A_i r_i} = (u^{A_i} h)^{r_i} v^{-r}$. It sets the secret key $SK = (S, K_0, K_1, \{K_{i,2}, K_{i,3}\}_{i\in[1,k]})$.
KeyGen.ran(SK). On input a secret key SK, the data consumer chooses a random $\tau \in \mathbb{Z}_p$ and computes $K_0' = K_0^{1/\tau} = g^{\alpha/\tau} w^{r/\tau}$, $K_1' = K_1^{1/\tau} = g^{r/\tau}$. For $i = 1$ to k, compute $K_{i,2}' = K_{i,2}^{1/\tau} = g^{r_i/\tau}$, $K_{i,3}' = K_{i,3}^{1/\tau} = (u^{A_i} h)^{r_i/\tau} v^{-r/\tau}$. The conversion key is $TK = (S, K_0', K_1', \{K_{i,2}', K_{i,3}'\}_{i\in[1,k]})$, the retrieval key is $RK = \tau$.
Encrypt.offline(PK, N'). On input public parameters PK and the number N' which assumes a maximum bound of N' rows in any LSSS structure, the data owner first picks a random $s \in \mathbb{Z}_p$ and computes $\hat{C} = e(g,g)^{\alpha s}, \hat{C_0} = g^s$. For $j = 1$ to N', choose random $t_j \in \mathbb{Z}_p$ and compute $\hat{C_{j,1}} = v^{t_j}$, $\hat{C_{j,2}} = h^{-t_j}$, $\hat{C_{j,3}} = g^{t_j}$. It sets the intermediate ciphertext $IT = (s, \hat{C}, \hat{C_0}, \{t_j, \hat{C_{j,1}}, \hat{C_{j,2}}, \hat{C_{j,3}}\}_{j\in[1,N']})$.
Encrypt.online$(PK, IT, m, (M, \rho))$. On input public parameters PK, an intermediate ciphertext IT, a plaintext m and an LSSS access structure (M, ρ), where M is an $l \times n$ ($l \leq N'$) matrix, the data owner first picks $\overrightarrow{y} = (s, y_2, \ldots, y_n)^T \in \mathbb{Z}_p^{n\times 1}$ where the random secret s from IT will be shared among the shares. The vector of the shares is $\overrightarrow{\lambda} = (\lambda_1, \ldots, \lambda_l)^T = M\overrightarrow{y}$. It computes $C = \hat{C} \cdot m = e(g,g)^{\alpha s} \cdot m$ and sets $C_0 = \hat{C_0} = g^s$. For $j = 1$ to l, compute and set

$$C_{j,1} = \hat{C_{j,1}} \cdot w^{\lambda_j} = w^{\lambda_j} v^{t_j},\ C_{j,2} = \hat{C_{j,2}} \cdot u^{-\rho(j)t_j} = (u^{\rho(j)} h)^{-t_j},\ C_{j,3} = \hat{C_{j,3}} = g^{t_j}.$$

The ciphertext is $CT = ((M, \rho), C, C_0, \{C_{j,1}, C_{j,2}, C_{j,3}\}_{j\in[1,l]})$.
Decrypt.out(TK, CT). On input a conversion key TK for the attribute set S and a ciphertext CT for access structure (M, ρ), if S does not satisfy this access

structure, the public cloud outputs $\perp$. Otherwise, it calculates $I = \{i : \rho(i) \in S\}$ and computes the constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \omega_i \cdot M_i = (1, 0, \ldots, 0)$, where M_i is the i-th row of the matrix M. Then it computes

$$A = \frac{e(C_0, K_0')}{\prod_{i \in I}(e(C_{i,1}, K_1') \cdot e(C_{i,2}, K_{j,2}') \cdot e(C_{i,3}, K_{j,3}'))^{\omega_i}} = e(g,g)^{\alpha s/\tau}.$$

where j is the index of the attribute $\rho(i)$ in S (it depends on i). It outputs the partially decrypted transformed ciphertext $TCT = (C = e(g,g)^{\alpha s} m,\ A)$.

Decrypt.user(RK, TCT). On input a retrieval key RK and a transformed ciphertext TCT, the data consumer computes $\frac{C}{A^{\tau}} = \frac{e(g,g)^{\alpha s} m}{(e(g,g)^{\alpha s/\tau})^{\tau}} = m$.

Theorem 1. *The basic COO-CP-ABE scheme is selective CPA-secure if the CP-ABE scheme in [21] is selective CPA-secure.*

Proof. The online and offline algorithms in our scheme are executed by the same entity, and the forms of user secret keys SK and ciphertexts CT are identical to those in [21]. Therefore, the modification does not affect the security proof. Moreover, we utilize the key blinding technique in [33]. The proof is simple and similar to [33], thus we omit it.

4.2 Construction of COO-CP-ABE-CRF

Technique Overview. To resist the exfiltration of secret information from arbitrarily compromised functional-maintaining algorithms executed by the PKG, the data owner and the data consumer, we propose the construction of COO-CP-ABE with cryptographic reverse firewalls (COO-CP-ABE-CRF) that is based on the basic construction in Sect. 4.1. In the construction, we introduce three reverse firewalls: $\mathcal{W}_{\mathbf{PKG}}$ (between the PKG and other entities), $\mathcal{W}_{\mathbf{DO}}$ (between the data owner and the public cloud), $\mathcal{W}_{\mathbf{DC}}$ (between the data consumer and the public cloud) to rerandomize the cryptographic keys PK, SK, TK and the ciphertexts CT. To rerandomize the cryptographic keys, we utilize the malleability of keys PK, SK, TK and rewind the updated PK' to the PKG. To rerandomize the ciphertexts, we utilize the homomorphism of the ciphertext [21] and the linear secret sharing schemes (LSSS) [29]. Moreover, we utilize the concessive online/offline technique to optimize the online computation efficiency of the reverse firewalls $\mathcal{W}_{\mathbf{PKG}}, \mathcal{W}_{\mathbf{DO}}$. Now, we present the construction in detail.

System Initialization. The PKG runs $Setup(\lambda, U) \rightarrow (PK, MSK)$ and keeps MSK by itself. Before broadcasting PK to other entities, PKG will first send PK to the reverse firewall $\mathcal{W}_{\mathbf{PKG}}$, which runs the following algorithm.

$\mathcal{W}_{\mathbf{PKG}}$**.Setup**$(PK)$. Receiving public parameters PK from the PKG, the reverse firewall $\mathcal{W}_{\mathbf{PKG}}$ picks random $a, b, c, d, e, f \in \mathbb{Z}_p$ and computes $g' = g^a, u' = u^b, h' = h^c, w' = w^d, v' = v^e, \alpha' = \alpha + f, e(g', g')^{\alpha'} = e(g,g)^{\alpha a^2} e(g,g)^{a^2 f} = e(g,g)^{a^2(\alpha+f)}$. It stores f and broadcasts the updated public parameters $PK' = (D, g', u', h', w', v', e(g', g')^{\alpha'})$ to all the entities including the PKG.

During the key generation phase, the PKG runs *KeyGen.offline*$(PK', MSK, N) \rightarrow ISK$ (which can be done in the spare time) and *KeyGen.online* $(PK', S, ISK) \rightarrow SK$, then sends $SK = (S, K_0, K_1, \{K_{i,2}, K_{i,3}\}_{i\in[1,k]})$ to the reverse firewall $\mathcal{W}_{\mathbf{PKG}}$, which does as follows.

Before receiving the user secret key SK, the reverse firewall $\mathcal{W}_{\mathbf{PKG}}$ does some preparation work such as the below algorithm in the spare time.

$\mathcal{W}_{\mathbf{PKG}}$**.KG.offline**$(PK', f, N)$. On input updated public parameters PK', the stored random f and a number N which assumes the number of attributes, the reverse firewall $\mathcal{W}_{\mathbf{PKG}}$ first picks $N+1$ random $r', r'_1, r'_2, \ldots, r'_N \in \mathbb{Z}_p$ and computes $\hat{K'_0} = g'^f w'^{r'}$, $\hat{K'_1} = g'^{r'}$. Then for $i = 1$ to N, it computes $\hat{K'_{i,2}} = g'^{r'_i}, \hat{K'_{i,3}} = h'^{r'_i} v'^{-r'}$. It sets the updated intermediate secret key $ISK' = (\hat{K'_0}, \hat{K'_1}, \{r'_i, \hat{K'_{i,2}}, \hat{K'_{i,3}}\}_{i\in[1,N]})$.

When the secret key SK is arriving, $\mathcal{W}_{\mathbf{PKG}}$ runs the following algorithm.

$\mathcal{W}_{\mathbf{PKG}}$**.KG.online**$(PK', ISK', SK)$. On input updated public parameters PK', an updated intermediate secret key ISK' and a secret key SK, the reverse firewall $\mathcal{W}_{\mathbf{PKG}}$ computes $K'_0 = K_0 \cdot \hat{K'_0} = g'^{\alpha+f} w'^{r+r'} = g'^{\alpha'} w'^{r+r'}$, $K'_1 = K_1 \cdot \hat{K'_1} = g'^{r+r'}$. Then for $i = 1$ to k where $k \leq N$, compute

$$K'_{i,2} = K_{i,2} \cdot \hat{K'_{i,2}} = g'^{r_i+r'_i},\ K'_{i,3} = K_{i,3} \cdot \hat{K'_{i,3}} \cdot u'^{A_i r'_i} = (u'^{A_i} h')^{r_i+r'_i} v'^{-r-r'}.$$

It sends the updated secret key $SK' = (S, K'_0, K'_1, \{K'_{i,2}, K'_{i,3}\}_{i\in[1,k]})$ to the user.

Data Upload. The data owner calls *Encrypt.offline*$(PK', N') \rightarrow IT$ (which can be done in the spare time) and *Encrypt.online*$(PK', IT, m, (M, \rho)) \rightarrow CT$, then sends $CT = ((M, \rho), C, C_0, \{C_{j,1}, C_{j,2}, C_{j,3}\}_{j\in[1,l]})$ to the reverse firewall $\mathcal{W}_{\mathbf{DO}}$, which does as follows.

Before receiving the ciphertext CT, the reverse firewall $\mathcal{W}_{\mathbf{DO}}$ does some preparation work such as the below algorithm in the spare time.

$\mathcal{W}_{\mathbf{DO}}$**.Enc.offline**$(PK', N')$. On input updated public parameters PK' and a number N' which assumes a maximum bound of N' rows in any LSSS structure, the reverse firewall $\mathcal{W}_{\mathbf{DO}}$ first picks another random secret $s' \in \mathbb{Z}_p$ to be shared among the shares. Then pick N' random exponents $t'_1, t'_2, \ldots, t'_{N'} \in \mathbb{Z}_p$ and compute $\hat{C'} = e(g', g')^{\alpha' s'}$, $\hat{C'_0} = g'^{s'}$. For $j = 1$ to N', compute $\hat{C'_{j,1}} = v'^{t'_j}$, $\hat{C'_{j,2}} = h'^{-t'_j}$, $\hat{C'_{j,3}} = g'^{t'_j}$. It sends the updated intermediate ciphertext $IT' = (s', \hat{C'}, \hat{C'_0}, \{t'_j, \hat{C'_{j,1}}, \hat{C'_{j,2}}, \hat{C'_{j,3}}\}_{j\in[1,N']})$.

When the ciphertext CT is arriving, $\mathcal{W}_{\mathbf{DO}}$ runs the following algorithm.

$\mathcal{W}_{\mathbf{DO}}$**.Enc.online**$(PK', IT', CT)$. On input updated public parameters PK', an updated intermediate ciphertext IT' and a ciphertext CT, the reverse firewall $\mathcal{W}_{\mathbf{DO}}$ first sets and picks $\overrightarrow{y'} = (s', y'_2, \ldots, y'_n)^T \in \mathbb{Z}_p^{n\times 1}$ where s' is the same random secret in IT'. The vector of the shares is $\overrightarrow{\lambda'} = (\lambda'_1, \ldots, \lambda'_l)^T = M\overrightarrow{y'}$. Then it computes $C' = C \cdot \hat{C'} = m \cdot e(g', g')^{\alpha'(s+s')}$, $C'_0 = C_0 \cdot \hat{C'_0} = g'^{s+s'}$. For $j = 1$ to l where $l \leq N'$, compute

$$C'_{j,1} = C_{j,1} \cdot \hat{C'_{j,1}} \cdot w'^{\lambda'_j} = w'^{\lambda_j+\lambda'_j} v'^{t_j+t'_j},\ C'_{j,3} = C_{j,3} \cdot \hat{C'_{j,3}} = g'^{t_j+t'_j},$$
$$C'_{j,2} = C_{j,2} \cdot \hat{C'_{j,2}} \cdot u'^{-\rho(j)t'_j} = (u'^{\rho(j)} h')^{-(t_j+t'_j)}.$$

It sends the updated ciphertext $CT' = ((M, \rho), C', C'_0, \{C'_{j,1}, C'_{j,2}, C'_{j,3}\}_{j\in[1,l]})$ to the public cloud.

Data Download. The data consumer first runs $KeyGen.ran(SK') \rightarrow (TK, RK)$ and sends $TK = (S, K''_0, K''_1, \{K''_{i,2}, K''_{i,3}\}_{i\in[1,k]})$ to the reverse firewall $\mathcal{W}_{\mathbf{DC}}$, which runs the following algorithm.

$\mathcal{W}_{\mathbf{DC}}$.**TKUpdate**$(TK)$. On input a conversion key TK, the reverse firewall $\mathcal{W}_{\mathbf{DC}}$ chooses a random $\beta \in \mathbb{Z}_p$ and computes $K'''_0 = {K''_0}^{1/\beta} = {g'}^{\alpha'/\tau\beta} {w'}^{(r+r')/\tau\beta}, K'''_1 = {K''_1}^{1/\beta} = {g'}^{(r+r')/\tau\beta}$. Then for $i = 1$ to k, it computes

$$K'''_{i,2} = {K''_{i,2}}^{1/\beta} = {g'}^{(r_i+r'_i)/\tau\beta},\ K'''_{i,3} = {K''_{i,3}}^{1/\beta} = ({u'}^{A_i} h')^{(r_i+r'_i)/\tau\beta} {v'}^{-(r+r')/\tau\beta}.$$

It stores β and sends the updated conversion key $TK' = (S, K'''_0, K'''_1, \{K'''_{i,2}, K'''_{i,3}\}_{i\in[1,k]})$ to the public cloud.

Receiving the decryption request from the data consumer, the public cloud runs $Decrypt.out(TK', CT') \rightarrow TCT$ and sends $TCT = (C' = e(g', g')^{\alpha'(s+s')} m, A = e(g', g')^{\alpha'(s+s')/\tau\beta})$ to the reverse firewall $\mathcal{W}_{\mathbf{DC}}$, which runs the algorithm.

$\mathcal{W}_{\mathbf{DC}}$.**Decrypt**$(TCT, \beta)$. On input a transformed ciphertext TCT and the stored β, the reverse firewall $\mathcal{W}_{\mathbf{DC}}$ computes $A' = A^\beta = e(g', g')^{\alpha'(s+s')/\tau}$ and sends the updated $TCT' = (C', A')$ to the data consumer.

Receiving the updated transformed ciphertext TCT', the data consumer runs $Decrypt.user(RK, TCT')$ to recover the plaintext m.

4.3 Security Analysis

Theorem 2. *The proposed COO-CP-ABE-CRF is selective CPA-secure and the reverse firewalls for the PKG, the data owner and the data consumer maintain functionality, weakly preserve security, and weakly resist exfiltration if the basic construction of COO-CP-ABE in Sect. 4.1 is selective CPA-secure.*

Proof. We verify that our construction satisfies the following properties.

Functionality Maintaining. The correctness can be easily verified. If the attribute set S of the secret key is authorized, we have that $\sum_{i\in I} \omega_i \cdot (\lambda_i + \lambda'_i) = s + s'$. Therefore,

$$\begin{aligned} A' &= \frac{e(C'_0, K'''_0)}{\prod_{i\in I}(e(C'_{i,1}, K'''_1) \cdot e(C'_{i,2}, K'''_{j,2}) \cdot e(C'_{i,3}, K'''_{j,3}))^{\omega_i}} \\ &= \frac{e(g', g')^{\alpha'(s+s')/\tau\beta} e(g', w')^{(r+r')(s+s')/\tau\beta}}{\prod_{i\in I} e(g', w')^{(r+r')(\lambda_i+\lambda'_i)\omega_i/\tau\beta} e(g', v')^{(r+r')(t_i+t'_i)\omega_i/\tau\beta}} \\ &\cdot \frac{1}{\prod_{i\in I} e(g', u')^{-\rho(i)(t_i+t'_i)(r_i+r'_i)\omega_i/\tau\beta} e(g', h')^{-(t_i+t'_i)(r_i+r'_i)\omega_i/\tau\beta}} \\ &\cdot \frac{1}{\prod_{i\in I} e(g', u')^{(t_i+t'_i)(r_i+r'_i)A(i)\omega_i/\tau\beta} e(g', h')^{(t_i+t'_i)(r_i+r'_i)\omega_i/\tau\beta}} \end{aligned}$$

$$\cdot \frac{1}{\prod_{i\in I} e(g', v')^{-(r+r')(t_i+t_i')\omega_i/\tau\beta}}$$

$$= \frac{e(g', g')^{\alpha'(s+s')/\tau\beta} e(g', w')^{(r+r')(s+s')/\tau\beta}}{e(g', w')^{(r+r')\sum_{i\in I}(\lambda_i+\lambda_i')\omega_i/\tau\beta}} = e(g', g')^{\alpha'(s+s')/\tau\beta}$$

$$\frac{C'}{A'^{\tau}} = \frac{C'}{A^{\beta\tau}} = \frac{m \cdot e(g', g')^{\alpha'(s+s')}}{e(g', g')^{\alpha'(s+s')}} = m$$

Weak Security Preservation and Weak Exfiltration Resistance. For any tampered implementation on the PKG, the data owner and data consumer that maintains functionality, we will prove the selective CPA security of our proposed COO-CP-ABE-CRF with tampered algorithms $Setup^*, KeyGen.offline^*$, $KeyGen.online^*$, $KeyGen.ran^*$, $Encrypt.offline^*, Encrypt.online^*$ by proving the indistinguishability between the security game of COO-CP-ABE-CRF and the security game of the basic construction COO-CP-ABE in Sect. 4.1. Additionally, the weak security preservation and weak exfiltration resistance for CRF can be easily proved. Next, we consider the following games:

Game 0. It is identical to the security game of COO-CP-ABE-CRF in Sect. 3.2.

Game 1. Same as *Game 0* except that during the setup phase, PK, MSK are generated by *Setup* in the basic construction, not $Setup^*$ and $\mathcal{W}_{\mathbf{PKG}}.Setup$.

Game 2. Same as *Game 1* except that during *Phase 1* and *Phase 2*, the secret key SK is generated by $KeyGen.offline, KeyGen.online$ in the basic construction, not $KeyGen.offline^*, KeyGen.online^*$, $\mathcal{W}_{\mathbf{PKG}}.KG.offline$ and $\mathcal{W}_{\mathbf{PKG}}.KG.online$, and the conversion key TK is generated by $KeyGen.ran$ in the basic construction, not $KeyGen.ran^*$ and $\mathcal{W}_{\mathbf{DC}}.TKUpdate$.

Game 3. Same as *Game 2* except that during the challenge phase, the challenge ciphertext CT_b are generated by $Encrypt.offline, Encrypt.online$ in the basic construction, not $Encrypt.offline^*, Encrypt.online^*$, $\mathcal{W}_{\mathbf{DO}}.Enc.offline$ and $\mathcal{W}_{\mathbf{DO}}.Enc.online$. Actually, *Game 3* is the security game of the basic construction.

Then we prove the indistinguishability between the pairs *Game 0* and *Game 1*, *Game 1* and *Game 2*, *Game 2* and *Game 3* respectively. For the pair *Game 0* and *Game 1*, for any tampered algorithm $Setup^*$, after the post-processing by the reverse firewall $\mathcal{W}_{\mathbf{PKG}}.Setup$, the public parameters PK are uniformly random due to the key malleability, which is identical to the original algorithm *Setup* in the basic construction, regardless of the behavior of $Setup^*$. Thus *Game 0* and *Game 1* are indistinguishable. Since the user secret key SK and the conversion key TK also have key malleability, *Game 1* and *Game 2* are indistinguishable. For the pair *Game 2* and *Game 3*, for any tampered algorithm $Encrypt.offline^*, Encrypt.online^*$, after the post-processing by the reverse firewall $\mathcal{W}_{\mathbf{DO}}.Enc.offline, \mathcal{W}_{\mathbf{DO}}.Enc.online$, the updated ciphertext CT' are uniformly regenerated because the ABE scheme and the linear secret sharing scheme are rerandomizable, which is identical to the encryption algorithm in the basic

Table 1. Efficiency comparison

Operation	[21]	[32]	Ours
Setup	1Exp + 1P	1Exp + 1P	1Exp + 1P
KeyGen.online	(4y + 3)Exp	3yExp	yExp
Encrypt.online	(5l + 2)Exp	0	2lExp
KeyGen.ran	×	×	(2l + 2)Exp
Decrypt.user	(|I|)Exp + (3|I| + 1)P	(|I + 1|)Exp + (3|I| + 2)P	1Exp
$\mathcal{W}_{\mathbf{PGK}}.Setup$	×	×	7Exp + 1P
$\mathcal{W}_{\mathbf{PKG}}.KG.online$	×	×	yExp
$\mathcal{W}_{\mathbf{DO}}.Enc.online$	×	×	2lExp
$\mathcal{W}_{\mathbf{DC}}.TKUpdate$	×	×	(2l + 2)Exp
$\mathcal{W}_{\mathbf{DC}}.Decrypt$	×	×	1Exp

‡Exp and P denote a modular exponentiation and a pairing computation, respectively. y, l, and I indicate the number of attributes, the access policy size, and the set that satisfies decryption requirement, respectively.

construction, regardless of the behavior of $Encrypt.offline^*, Encrypt.online^*$. Thus *Game 2* and *Game 3* are indistinguishable. Therefore, we conclude that *Game 0* and *Game 3* are indistinguishable. Since the basic construction is selective CPA-secure, the proposed OO-CP-ABE-CRF is selective CPA-secure.

The selective CPA security of the proposed scheme indicates that the reverse firewalls for PKG, the data owner and data consumer maintain weakly preserve security. The indistinguishability between *Game 0* and *Game 3* indicates that the reverse firewalls for PKG, the data owner and data owner maintain weakly resist exfiltration. Combining all the discussions, we complete the proof.

5 Performance Evaluations

5.1 Theoretical Analysis

The online computation cost of the PKG, the data owner, the data consumer and the reverse firewalls refers to the execution time of *Setup, KeyGen.online, Encrypt.online, KeyGen.ran, Decrypt.user,* $\mathcal{W}_{\mathbf{PGK}}.Setup$, $\mathcal{W}_{\mathbf{PKG}}.KG.online$, $\mathcal{W}_{\mathbf{DO}}.Enc.online$, $\mathcal{W}_{\mathbf{DC}}.TKUpdate$, $\mathcal{W}_{\mathbf{DC}}.Decrypt$. Table 1 compares the number of modular exponentiations and pairing operations in our construction with those in the original scheme [21] and online/offline ABE [32].

For *Setup*, the computation cost of three schemes are the same. For *KeyGen.online*, the efficiency rank is $[21] < [32] < Ours$. The efficiency of our construction is four times that of [21] and three times that of [32]. For *Encrypt.online*, the efficiency rank is $[21] < Ours \ll [32]$. The reason that we propose concessive online/offline technique instead of directly utilizing the technique in [32] is that the randomness of ciphertexts and user secret keys in [32]

cannot be all rerandomized, thus we choose to achieve stronger security by sacrificing a small amount of efficiency. For *Decrypt.user*, the efficiency rank is $[21] < [32] \ll Ours$, the data consumer only needs to do one exponentiation to complete the decryption.

Next, we analyze the efficiency of reverse firewalls. For *Setup*, the overhead of the reverse firewall $\mathcal{W}_{\mathbf{PKG}}$ is more than that of the PKG, because $\mathcal{W}_{\mathbf{PKG}}$ needs to do the rerandomization. For *KG.online* and *Enc.online*, the reverse firewalls $\mathcal{W}_{\mathbf{PKG}}, \mathcal{W}_{\mathbf{DO}}$ have the same workload with *KeyGen.online, Encrypt.online* in our construction. For *Decrypt*, the computation cost of $\mathcal{W}_{\mathbf{DC}}.TKUpdate, \mathcal{W}_{\mathbf{DC}}.Decrypt$ are the same as *KeyGen.ran, Decrypt.user* in our construction. In general, the encryption scheme in the CRF framework will suffer double computation latency during key generation and encryption phase, while the total latency of key generation in our construction is two times less than the original scheme [21], and the total latency of encryption is less than [21]. We remark that our proposed COO-CP-ABE-CRF not only strengthens the security to resist the exfiltration of secret information, but also achieves high computation efficiency.

5.2 Experimental Analysis

To evaluate the practical performance, we develop an extensible library called libabe, which offers essential APIs for implementing ABE schemes. To be compatible with Android OS, libabe is developed by C language and only dependent on Pairing-Based Cryptography (PBC) library [34], thus we can develop the evaluation program on Android OS with Java Native Interface (JNI). The curve that we choose is the 224-bit MNT elliptic curve from PBC. We use a laptop produced by HASEE to act as the PKG, the public cloud and three reverse firewalls, a mobile device produced by XIAOMI plays the part of the data owner and the data consumer. The device configuration is presented in Table 2.

Experiment Setting. We set access policies for CTs in the form of (S_1 AND ... AND S_l) to simulate the worst situation. We set 20 distinct access policies with l increasing from 10 to 100, repeat each instance 20 times and take the average value. The time is given in milliseconds. Since the routines of MNT elliptic curve adopt asymmetric groups while the groups in the scheme are symmetric, only a small change needs to be made. Specifically, there are three groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ and an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Because the time taken to execute operations in $\mathbb{G}_1$ is much less than $\mathbb{G}_2$ in MNT224 elliptic curve group, more operations in the scheme are executed in $\mathbb{G}_1$ rather than $\mathbb{G}_2$.

Computation Time. In Fig. 3, we show *KeyGen.offline Time* and *KeyGen.online Time* of the PKG, *Encrypt.offline Time* and *Encrypt.online Time* of the data owner, *KeyGen.ran Time* and *Decrypt.user Time* of the data consumer, *Decrypt.out Time* of the public cloud, $\mathcal{W}_{\mathbf{PKG}}.KG.offline$ *Time* and $\mathcal{W}_{\mathbf{PKG}}.KG.online$ *Time* of $\mathcal{W}_{\mathbf{PKG}}$, $\mathcal{W}_{\mathbf{DO}}.Enc.offline$ *Time* and $\mathcal{W}_{\mathbf{DO}}.Enc.online$ *Time* of $\mathcal{W}_{\mathbf{DO}}$, $\mathcal{W}_{\mathbf{DC}}.TKUpdate$ *Time* and $\mathcal{W}_{\mathbf{DC}}.Decrypt$ *Time* of $\mathcal{W}_{\mathbf{DC}}$.

Table 2. Device configuration

Type	Configuration	Role	Algorithm
Laptop (HASEE)	Intel Core i7-4710MQ @2.5 GHz, 8 GB RAM, Ubuntu 16.04LTS 64-bit	PKG, $\mathcal{W}_{\mathbf{PKG}}$, $\mathcal{W}_{\mathbf{DO}}$, $\mathcal{W}_{\mathbf{DC}}$, Public Cloud	*Setup*, *KeyGen.offline*, *KeyGen.online*, *Decrypt.out*, $\mathcal{W}_{\mathbf{PKG}}.Setup$, $\mathcal{W}_{\mathbf{PKG}}.KG.offline$, $\mathcal{W}_{\mathbf{PKG}}.KG.online$, $\mathcal{W}_{\mathbf{DO}}.Enc.offline$, $\mathcal{W}_{\mathbf{DO}}.Enc.online$, $\mathcal{W}_{\mathbf{DC}}.TKUpdate$, $\mathcal{W}_{\mathbf{DC}}.Decrypt$
Mobile Device (MIX 2)	Qualcomm Snapdragon 835@2.45 GHz, 6 GB RAM, Android 8.0	Data Owner, Data Consumer	*Encrypt.offline*, *Encrypt.online*, *KeyGen.ran*, *Decrypt.user*

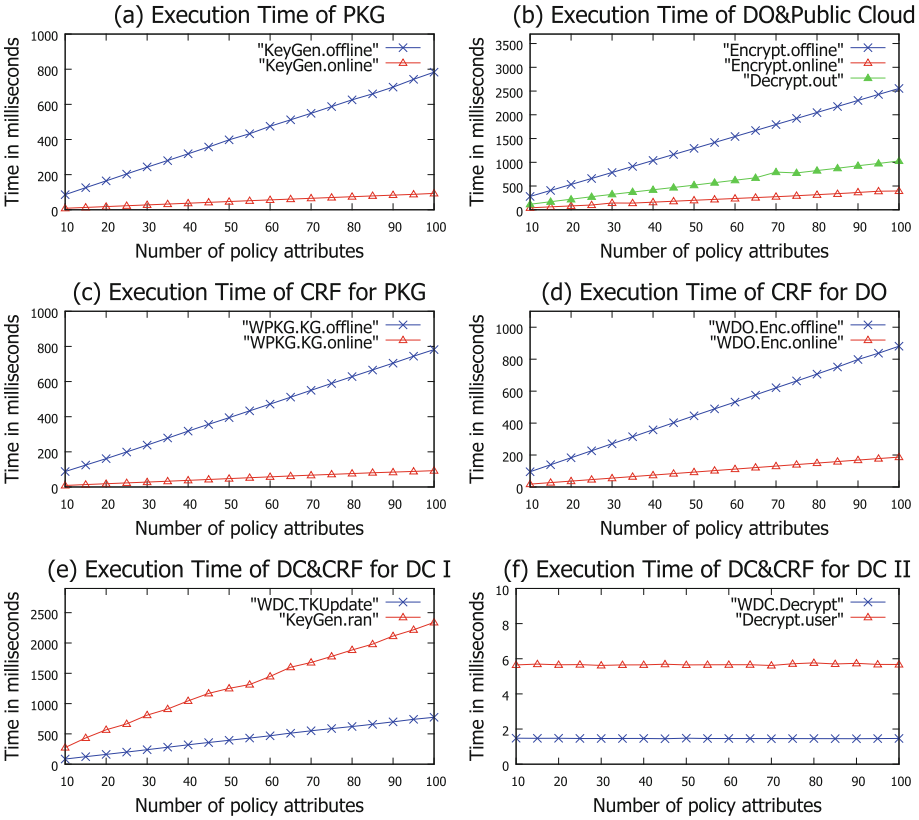

Fig. 3. Experimental results

In Fig. 3(a), *KeyGen.offline Time* is about 47 ms–783 ms while *KeyGen.online Time* is about 3 ms–93 ms. In Fig. 3(b), *Encrypt.offline Time* on the mobile phone is about 0.15 s–2.55 s while *Encrypt.online Time* on the mobile phone is about 22 ms–386 ms. *Decrypt.out Time* is about 64 ms–1.02 s. In

Fig. 3(e) and (f), *KeyGen.ran Time* on the mobile phone is about 0.14 s–2.33 s and *Decrypt.user Time* on the mobile phone is always about 5.6 ms, which is quite efficient for mobile devices. In Fig. 3(c), $\mathcal{W}_{\mathbf{PKG}}.KG.offline\ Time$ is about 48 ms–781 ms while $\mathcal{W}_{\mathbf{PKG}}.KG.online\ Time$ is about 3 ms–93 ms. In Fig. 3(d), $\mathcal{W}_{\mathbf{DO}}.Enc.offline\ Time$ is about 52 ms–881 ms while $\mathcal{W}_{\mathbf{DO}}.Enc.online\ Time$ is about 8 ms–186 ms. In Fig. 3(e), $\mathcal{W}_{\mathbf{DC}}.TKUpdate\ Time$ is about 46 ms–774 ms. In Fig. 3(f), $\mathcal{W}_{\mathbf{DC}}.Decrypt\ Time$ is always about 1.4 ms.

6 Conclusion

In this paper, we propose a concessive online/offline ciphertext-policy attribute based encryption with cryptographic reverse firewalls, which can resist the exfiltration of secret information. Furthermore, compared with the original scheme without CRF, our scheme reduces the total computation cost by half. Moreover, we develop an extensible library called libabe that is compatible with Android devices, and we implement the prototype on a laptop and a mobile phone. In the future, we will focus on designing more compact system model of ABE in the CRF framework without rewinding PK to PKG.

Acknowledgments. The authors thank anonymous reviewers for their valuable comments, and Wenhan Xu for many helps on the experiments. This work was supported in part by National Natural Science Foundation of China (Nos. 61632020, 61472416, 61772520), Key Research Project of Zhejiang Province (No. 2017C01062), Fundamental theory and cutting edge technology Research Program of Institute of Information Engineering, CAS (No. Y7Z0321102), Australian Research Council Discovery Early Career Researcher Award (No. DE150101116), Scientific Research Plan Project of Tianjin Municipal Education Commission (Grant No. 2017KJ237).

References

1. Wan, Z., Liu, J., Zhang, R., Deng, R.H.: A collusion-resistant conditional access system for flexible-pay-per-channel pay-tv broadcasting. IEEE Trans. Multimed. **15**(6), 1353–1364 (2013)
2. Ning, J., Cao, Z., Dong, X., Liang, K., Ma, H., Wei, L.: Auditable σ-time outsourced attribute-based encryption for access control in cloud computing. IEEE Trans. Inf. Forensics Secur. **13**(1), 94–105 (2018)
3. Zhou, J., Cao, Z., Dong, X., Lin, X.: TR-MABE: white-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems. In: IEEE Conference on Computer Communications, INFOCOM 2015, Kowloon, Hong Kong, 26 April–1 May, pp. 2398–2406 (2015)
4. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans. Parallel Distrib. Syst. **24**(1), 131–143 (2013)
5. Ma, H., Zhang, R., Wan, Z., Lu, Y., Lin, S.: Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing. IEEE Trans. Dependable Secur. Comput. **14**(6), 679–692 (2017)

6. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, 30 October–3 November, pp. 89–98 (2006)
7. Liu, J.K., Au, M.H., Huang, X., Lu, R., Li, J.: Fine-grained two-factor access control for web-based cloud computing services. IEEE Trans. Inf. Forensics Secur. **11**(3), 484–497 (2016)
8. Zhang, R., Ma, H., Lu, Y.: Fine-grained access control system based on fully outsourced attribute-based encryption. J. Syst. Softw. **125**, 344–353 (2017)
9. James Ball, J.B., Greenwald, G.: Revealed: how US and UK spy agencies defeat internet privacy and security. Guardian Weekly (2013)
10. Perlroth, N., Larson, J., Shane, S.: NSA able to foil basic safeguards of privacy on web. The New York Times, 5 September 2013
11. Dodis, Y., Ganesh, C., Golovnev, A., Juels, A., Ristenpart, T.: A formal treatment of backdoored pseudorandom generators. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 101–126. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_5
12. Checkoway, S., et al.: On the practical exploitability of dual EC in TLS implementations. In: Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, 20–22 August 2014, pp. 319–335 (2014)
13. Greenwald, G.: No place to hide: Edward Snowden, the NSA, and the US surveillance state. Macmillan, New York (2014)
14. Vulnerability summary for CVE-2014-1260 (heartbleed), April 2014. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1260
15. Vulnerability summary for CVE-2014-1266 (goto fail), February 2014. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1266
16. Vulnerability summary for CVE-2014-6271 (shellshock), September 2014. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
17. Lenstra, A.K., Hughes, J.P., Augier, M., Bos, J.W., Kleinjung, T., Wachter, C.: Public keys. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 626–642. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_37
18. 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
19. Dodis, Y., Mironov, I., Stephens-Davidowitz, N.: Message transmission with reverse firewalls—secure communication on corrupted machines. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 341–372. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_13
20. 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
21. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, Berlin, Germany, 4–8 November, pp. 463–474 (2013)
22. 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

23. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: PACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, 28–31 October, pp. 195–203 (2007)
24. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_4
25. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_11
26. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4
27. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy (S&P 2007), Oakland, CA, USA, 20–23 May 2007, pp. 321–334. IEEE (2007)
28. Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, VA, USA, 28–31 October, pp. 456–465 (2007)
29. Beimel, A.: Secure schemes for secret sharing and key distribution. Ph.D. thesis, Technion-Israel Institute of Technology, Faculty of Computer Science (1996)
30. Li, J., Jia, C., Li, J., Chen, X.: Outsourcing encryption of attribute-based encryption with mapreduce. In: Chim, T.W., Yuen, T.H. (eds.) ICICS 2012. LNCS, vol. 7618, pp. 191–201. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34129-8_17
31. Li, J., Huang, X., Li, J., Chen, X., Xiang, Y.: Securely outsourcing attribute-based encryption with checkability. IEEE Trans. Parallel Distrib. Syst. **25**(8), 2201–2210 (2014)
32. Hohenberger, S., Waters, B.: Online/offline attribute-based encryption. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 293–310. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_17
33. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of ABE ciphertexts. In: 20th USENIX Security Symposium, San Francisco, CA, USA, 8–12 August (2011)
34. Lynn, B.: The stanford pairing based crypto library. http://crypto.stanford.edu/pbc