SM9 Key Encapsulation Mechanism for Power Monitoring Systems

Chao Hong*[1], Peng Xiao[2], Pandeng Li[3], Zhenhong Zhang[4], Yiwei Yang[5], Biao Bai[6]
Electric Power Research Institute, China Southern Power Grid, Guangzhou 510663, China[1,3,5]
Guangdong Provincial Key Laboratory of Power System Network Security, Guangzhou 510663, China[1,3,5]
Information Center of Yunnan Power Grid Co., Ltd., Kunming 650000, China[2,4,6]

***Abstract*—The boundaries of the new power system network are blurred, and data privacy and security are threatened. Although the SM9 algorithm is widely used in power systems to protect data security, its efficiency and security remain the main issues in application. Therefore, an SM9 key encapsulation mechanism (OSM9-KEM-CRF) was proposed to support outsourced decryption and cryptographic reverse firewall. In order to resist the backdoor attacks, we deployed cryptographic reverse firewalls at the terminals and proved that the proposed OSM9-KEM-CRF is ID-IND-CCA2 secure. The cryptographic reverse firewalls maintain functionality, weakly retain security, and weakly resist penetration, thereby enhancing the security of the scheme. In addition, considering the limited computing resources of terminal devices, decryption operations are outsourced to cloud servers in order to reduce the computational burden on the terminals. Compared with other SM9-KEMs, the proposed mechanism not only reduces computational and communication overhead, but also lowers energy consumption. Therefore, the proposed mechanism is more suitable for power monitoring systems.**

Keywords*—SM9; Outsourced decryption; cryptographic reverse firewall; power monitoring systems

I. Introduction

With the wide application of IoT technology in power systems, the boundaries of new power system networks are becoming increasingly blurred, and a large number of terminal monitoring devices with limited resources have emerged in power monitoring networks. Although data can be stored in the cloud and pre-processed by cloud servers, thus reducing the storage and computational burden on these terminal devices. However, once the data is out of the direct control of the user, it will face the risk of privacy and security. Information security measures will become the main means of protection. Therefore, there is an urgent need to carry out research on power control systems and lightweight security protection technology.

Chen et al. [1] developed a power monitoring system based on the SM2 algorithm in 2022. However, SM2 algorithm requires complex public key certificate management, while identity-based cryptographic algorithm can avoid complex public key certificate management, and is more sui for new power monitoring systems with many members and dynamic changes in members.

The SM9 is an identity based cryptographic algorithm, which was officially released in 2016 and identified as the standard algorithm for the cryptographic industry of China [2]. Cheng et al. [3] formally analyzed the security of the SM9 key agreement and the SM9 encryption scheme. Lai et al. [4] proposed Twin-SM9 key encapsulation mechanism using Twin-Hash-ElGamal technique.

A. Related Work

In power monitoring systems, SM9 cryptographic algorithms are favored for their simplified public key certificate management, but their high computational demand on resource-constrained monitoring devices and sensors highlights the need for efficiency optimization. This is especially true for resource-constrained end devices that are widely deployed in power system networks. These devices, such as smart meters, surveillance cameras, and other sensors, have limited computational power, making the complex bilinear mapping operations in the SM9 algorithm the key to improving the decryption efficiency. Ji et al. [5] pointed out that the operation of the SM9 encryption algorithm consumes a large amount of time and computational resources, which makes it challenging to run it on resource-constrained devices. Wang et al. [6] improved the complex operations in SM9 cryptographic algorithm, which improved the computational efficiency of SM9 algorithm to a certain extent, but it is still a large burden for resource-constrained lightweight devices, and could not solve the problem fundamentally. Lai et al. [7] proposed an efficient online/offline identity-based encryption for this purpose, which provides an idea for the implementation of SM9 cryptographic algorithm on lightweight devices. Sun et al. [8] investigated the SM9-IBE encryption scheme based on online/offline techniques. Peng et al. [9] developed an efficient certificate-free online/offline signature scheme and created a lightweight data authentication protocol specifically for WBAN. Liu et al. [10] introduced outsourcing decryption technique in attribute encryption scheme to reduce the computational overhead of the user. This considers the use of outsourcing technique to solve the problem of computational difficulties in SM9 encryption and decryption. Liu [11] proposed an OSM9 key encapsulation mechanism that supports decryption outsourcing, outsourcing the decryption part of SM9 cryptographic algorithm to the cloud service center for decryption operation, which reduces the computational burden of the terminal equipment, but the mechanism requires the cloud server to generate its own public-private key pairs, which increases the requirements of the system's initialization settings.

However the Snowden incident [12] showed that even

*Corresponding authors.

provably secure cryptographic algorithms can be subject to backdoor attacks that threaten the security and privacy of user data. Mironov et al. [13] proposed the Cryptographic Reverse Firewall (CRF), which is an entity deployed on the user side to re-randomize the information received and sent by users, in order to prevent the leakage of the user's private information. Therefore, CRF can be deployed between the cloud server and the user, and even if the algorithm is tampered with by a backdoor, it will not threaten the security and privacy of the user's data. Therefore, constructing a cryptographic reverse firewall for the SM9 cryptographic algorithm is a very important task. Chen et al. [14] constructed several CRF-based cryptographic protocols by relying on a malleable smooth projective hash function with key malleability and element re-randomization. Zhou et al. [15] proposed an identity-based encryption scheme with CRF. Zhou et al. [16] designed a single-round, certificateless public key encryption scheme incorporating CRF with reduced communication overhead. Furthermore, Zhou [17] suggested a searchable public key encryption approach based on CRF. Zhou et al. [18] designed an identity-based proxy re-encryption scheme with CRF that can resist leakage attacks. Jin et al. [19] designed a blockchain and CRF-based proxy re-encryption scheme. Xiong et al. [20] designed an SM9 encryption scheme with CRFs and supports equation testing, but the scheme only sets CRFs for data owners. Li et al. [21] designed an online/offline attribute-based encryption scheme with CRFs for IoT communication.

As can be seen from the above, The OSM9 key encapsulation mechanism proposed by Liu et al. [11], although considering outsourced decryption, prolongs user waiting time and does not take into account the threat of information leakage. On the other hand, although Xiong et al. [20] proposed the SM9 algorithm with cryptographic reverse firewall, this algorithm does not support outsourced decryption and does not consider the situation where the key generation center and data users are subjected to backdoor attacks. At present, there is no cryptographic reverse firewall built for outsourced decryption. A new mechanism needs to be proposed to consider the potential threat of backdoor attacks during cloud server outsourcing decryption.

B. Research Contributions

This paper focuses on the power monitoring system based on the SM2 cryptographic system proposed by Chen et al. [1], and constructs an SM9-KEM suitable for power monitoring systems, which not only supports outsourcing decryption but also has the function of CRF. The primary contributions include:

1) Improve the efficiency of SM9 algorithm: The bilinear mapping in the decryption operation of SM9 is outsourced to the cloud, and the cloud service center is not required to generate its own public-private key pair. It reduces the computational burden of users and greatly improves the efficiency of the scheme.

2) Enhanced security of the SM9 algorithm: Not only has CRF been set up on the data user side to re-randomize ciphertext, but CRF has also been set up on the KGC and data owner sides to re-randomize public parameters and user keys. This enables the OSM9-KEM-CRF proposed in this paper, which supports outsourced decryption, to maintain its functionality and resist leakage even under backdoor attacks, further improving the security of the scheme.

C. Paper Organization

The remainder of this paper is organized as follows. Section II covers the fundamental concepts related to elliptic curves and reverse firewalls. Section III outlines the system model and the security model of OSM9-KEM-CRF. Section IV details the encapsulation mechanism of OSM9-KEM-CRF along with its security. Section V provides a comparison between our proposed scheme and existing schemes in terms of computational overhead, communication overhead, and energy consumption overhead. The conclusion in Section VI.

II. Relevant Theoretical Foundations

A. Elliptic Curve

For an elliptic curve $\mathbb{E}$: $y^3 = x^3 + ax + b(\mathrm{mod}p)$, where $a, b \in \mathbb{F}_p$, $(4a^3 + 27b^2) \bmod p \neq 0$, $\mathbb{F}_p$ is a finite field of order prime $p > 3$, let $\mathbb{G}$ be the group over $\mathbb{E}$, $p \in \mathbb{G}$, q on an elliptic curve, where $p \in \mathbb{G}$, q is the order of $\mathbb{G}$ and O is the infinity point of $\mathbb{G}$. The operations on the elliptic curve are as follows:

1) Addition of points: let $P(x_1, y_1) \in \mathbb{E}$, $Q(x_1, y_1) \in \mathbb{E}$, where $P \neq O, Q \neq O, P \neq -Q$, let $R(x_3, y_3)$ is equal to $P+Q$, then the calculation of R can be expressed as $x_3 = \lambda^2 - x_1 - x_2, y_3 = \lambda(x_1 - x_3) - y_1$, where $\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, P \neq Q \\ \frac{3x_1^2 + a}{2y_1}, P = Q \end{cases}$.

2) Scalar multiplication: given a point $P(x, y)$ on an elliptic curve and an integer k , scalar multiplication can be defined as $kP = \sum_{i=1}^{k} P_i$.

B. Bilinear Mapping

Let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be cyclic groups, respectively. Then the bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ has the following properties:

(1) Bilinearity: for $a, b \in \mathbb{Z}_p$, $P_1 \in \mathbb{G}_1$, $P_2 \in \mathbb{G}_2$ there is $e\left(aP_1, bP_2\right) = e(P_1, P_2)^{ab}$.
(2) Non-degeneracy: there exist elements $P_1 \in \mathbb{G}_1$, $P_2 \in \mathbb{G}_2$, such that $e\left(P_1, P_2\right) \neq 1$.
(3) Computability: for any elements $P_1 \in \mathbb{G}_1$, $P_2 \in \mathbb{G}_2$, there exists an efficient polynomial time algorithm to evaluate $e(P_1, P_2)$.

C. Cryptographic Reverse Firewall (CRF)

CRF is a stateful algorithm $\mathcal{W}$ with states and messages as inputs and updated states and messages as outputs. Simply, the state information of $\mathcal{W}$ is not explicitly represented. For participant P and cryptographic reverse firewall $\mathcal{W}$ in the system, $\mathcal{W} \circ \mathcal{P}$ is defined as the composed party. If $\mathcal{W}$ is composed of participant P, then we call $\mathcal{W}$ cryptographic reverse firewall P. There are three security requirements for cryptographic reverse firewalls, namely Functionality maintaining, weak security preserving, and weak resistance to exfiltration, as described in [22].

III. Formal Definition and Security Model of OSM9-KEM-CRF

A. OSM9-KEM-CRF System Model

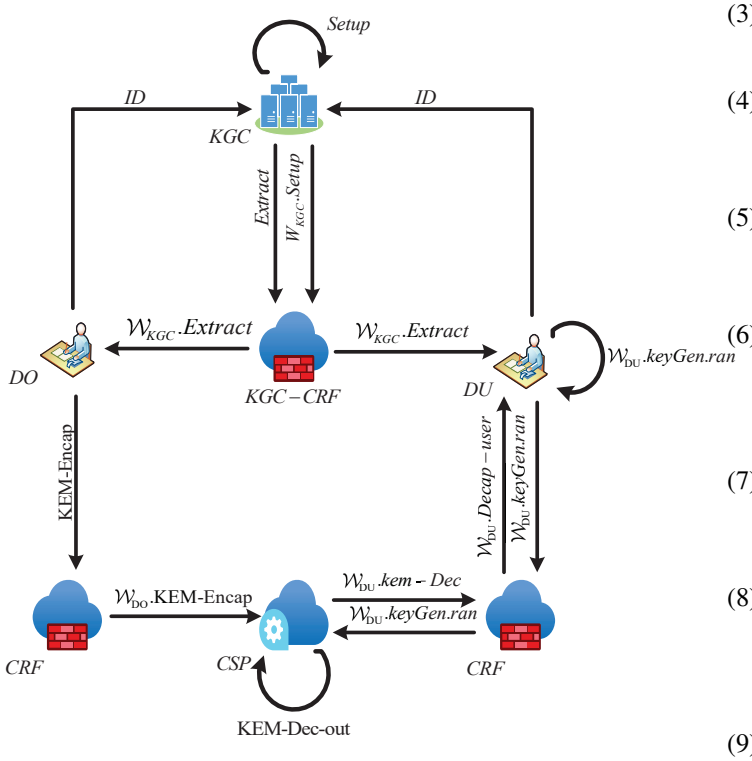

Fig. 1. Illustration of OSM9-KEM-CRF.

The OSM9-KEM-CRF for power monitoring system is shown in Fig. 1, which supports outsourced decryption and CRF and contains four members and three CRFs, that is the cloud service center (CSP), the key generation center (KGC) and its cryptographic reverse firewall $\mathcal{W}_{KGC}$, the data owner (DO) and its cryptographic reverse firewall $\mathcal{W}_{DO}$, the data user (DU) and its cryptographic reverse firewall $\mathcal{W}_{DU}$.

Specifically, KGC generates the master private key and the global public parameter pp. If the process is compromised then $\mathcal{W}_{KGC}$ randomizes pp and broadcasts it globally. The KGC is also responsible for generating the private keys of the users (DO, DU), and if the process is compromised, then $\mathcal{W}_{KGC}$ randomizes the user's private key.The CSP is responsible for storing the user's encrypted data and outsourcing the decryption of the data.The DO encrypts the data and uploads it to the CSP for storage. When the encryption process is compromised then $\mathcal{W}_{DO}$ randomizes the encrypted ciphertext.DU downloads the ciphertext from CSP and decrypts it. If the outsourced decryption key generation process is compromised then $\mathcal{W}_{DU}$ randomizes the outsourced decryption key.

B. OSM9-KEM-CRF System Model

The OSM9-KEM-CRF consists of the following 11 algorithms:

(1) $\text{Setup}(1^\lambda) \rightarrow (msk, pp)$.The algorithm is run by KGC.Input security parameter λ, output global public parameter pp and KGC master private key msk.

(2) $\mathcal{W}_{GA}.\text{Setup}(pp) \rightarrow pp'$.The algorithm is run by KGC's Cryptographic Reverse Firewall $\mathcal{W}_{KGC}$. Input the system public parameters pp and output the updated system public parameters pp'.

(3) $\text{Extract}(pp', msk, ID) \rightarrow sk$.The algorithm is run by KGC. Inputs pp',msk and user identity ID and outputs private key sk for user ID.

(4) $\mathcal{W}_{KGC}.\text{Extract}(sk) \rightarrow sk'$.The algorithm is run by KGC Cryptographic Reverse Firewall $\mathcal{W}_{KGC}$. It inputs the private key sk of the user ID, outputs the updated sk', and returns it to the user ID.

(5) $\text{KEM}-\text{Encap}(pp', ID) \rightarrow (K, C_1)$.The algorithm is run by the data owner DO with input pp' and outputs the encapsulated key K and encapsulated ciphertext C_1.

(6) $\mathcal{W}_{DO}.\text{KEM}-\text{Encap}(K, t, C_1) \rightarrow (K', C_1')$.The algorithm is run offline by the cryptographic reverse firewall $\mathcal{W}_{DO}$ of the data owner DO. Input (K, C_1), output updated encapsulated key K' and encapsulated ciphertext C_1'.

(7) $\text{KenGen.ran}(sk') \rightarrow (TK, RK)$.The algorithm is run by the user DU, which inputs its own private key sk' and outputs the transformation key TK and retrieval RK.

(8) $\mathcal{W}_{DC}.\text{TKUpdate}(TK) \rightarrow (TK', \beta)$.The algorithm is run by the password reversal firewall $\mathcal{W}_{DU}$ of the user user DU. Input TK. Output the updated conversion key TK', keeping the corresponding random number β.

(9) $\text{KEM}-\text{Decap}-\text{out}(pp', TK', C_1') \rightarrow TCT$.The algorithm is run by CSP. Input pp', TK', C_1', Output convert ciphertext.

(10) $\mathcal{W}_{DU}.\text{Decrypt}(TCT, \beta) \rightarrow TCT'$. The algorithm is run by the Cryptographic Reverse Firewall of the data user DU. Input TCT, β and output TCT'.

(11) $\text{KEM}-\text{Decap}-\text{user}(pp', RK, TCT') \rightarrow K'$.The algorithm is run by the data user DU. Input pp', TCT', RK, Output updated encapsulated key K'.

Correctness: For security parameters and encapsulated keys, correctness is required for all
$\text{Setup}(1^\lambda) \rightarrow (msk, pp)$,
$\mathcal{W}_{GA}.\text{Setup}(pp) \rightarrow pp'$,
$\text{Extract}(pp', msk, ID) \rightarrow sk$,
$\mathcal{W}_{KGC}.\text{Extract}(sk) \rightarrow sk'$,
$\text{KEM}-\text{Encap}(pp', ID) \rightarrow (K, C_1)$,
$\mathcal{W}_{DO}.\text{KEM}-\text{Encap}(K, t, C_1) \rightarrow (K', C_1')$,
$\text{KenGen.ran}(sk') \rightarrow (TK, RK)$,
$\mathcal{W}_{DC}.\text{TKUpdate}(TK) \rightarrow (TK', \beta)$,
$\text{KEM}-\text{Decap}-\text{out}(pp', TK', C_1') \rightarrow TCT$,
$\mathcal{W}_{DU}.\text{Decrypt}(TCT, \beta) \rightarrow TCT'$satisfy
$\text{KEM}-\text{Decap}-\text{out}(RK, TCT') \rightarrow K'$.

C. Security Model for OSM9-KEM-CRF

Based on the security models of [3] and [22], this paper defines the security model of OSM9-KEM-CRF. In OSM9-KEM-CRF, it is assumed that KGC, DO and DU are fully trusted and the cloud service provider CSP is semi-trusted. Since the algorithms (Setup, Extract, KEM-Encap, KEM-Decap-out, KEM-Decap-user) of OSM9-KEM-CRF remain functional

after the implantation of a malicious trapdoor, it is necessary to take into account that these algorithms can be attacked without the knowledge of the executor. Also considering that $\mathcal{W}_{DO}$ and $\mathcal{W}_{DU}$ would be curious about the user's data, it is assumed that $\mathcal{W}_{DO}$ and $\mathcal{W}_{DU}$ are semi-trustworthy. Since $\mathcal{W}_{KGCU}$ can access to the user's decryption key, it is assumed to be fully trusted. In addition, all cryptographic reverse firewalls are considered to be trusted areas and will not be tampered with by any outsiders.

The ID-IND-CCA2 security of the OSM9-KEM-CRF is defined by a game between Challenger $\mathcal{C}$ and Adversary $\mathcal{A}$. The game is played by the challenger and the adversary.

Initialization.The adversary sends function maintenance algorithm Setup^*,$\mathrm{Extract}^*$,
$\mathrm{KEM-Encap}^*$,$\mathrm{KeyGen.ran}^*$,$\mathrm{KEM-Decap-out}^*$, and $\mathrm{KEM-Decap-user}^*$ to the challenger $\mathcal{C}$.

Setup. Challenger $\mathcal{C}$ runs $\mathrm{Setup}(1^\lambda) \rightarrow (msk, pp)$, $\mathcal{W}_{KGC}.\mathrm{Setup}(pp) \rightarrow pp'$,then sends pp' to adversary $\mathcal{A}$.

Phase 1.Adversary $\mathcal{A}$ can adaptively query the private key oracle. For the query identity entered by the adversary, the challenger $\mathcal{C}$ runs
$\mathrm{Extract}(pp', msk, ID) \rightarrow sk$,
$\mathcal{W}_{KGC}.\mathrm{Extract}(sk) \rightarrow sk'$,
$\mathrm{KenGen.ran}(sk') \rightarrow (TK, RK)$,
$\mathcal{W}_{DC}.\mathrm{TKUpdate}(TK) \rightarrow (TK', \beta)$,
then returns sk' and TK' to adversary $\mathcal{A}$. Challenge.Adversary $\mathcal{A}$ sends a challenge identity ID^* to challenger $\mathcal{C}$. Challenger $\mathcal{C}$ runs $\mathrm{KEM-Encap}(pp', ID^*) \rightarrow (K_0, C_1^*)$,$\mathcal{W}_{DO}.KEM-Encap(K_0, C_1^*) \rightarrow (K_0', {C'^*}_1)$ and then randomly selects a key K_1' in the key space, bit $b \leftarrow \{0,1\}$, and then sends $(K_b', {C'^*}_1)$ to adversary $\mathcal{A}$.

Phase 2.As in Phase 1, adversary $\mathcal{A}$ can adaptively query the private key of the user, but not the private key of user ID^*. Additionally adversary $\mathcal{A}$ can adaptively query the decapsulation oracle. For the (ID, C) inputted by adversary, the challenger runs $KEM-Decap-out(pp', TK', C_1') \rightarrow TCT$,$\mathcal{W}_{DU}.Decrypt(TCT, \beta) \rightarrow TCT'$,$KEM-Decap-user(RK, TCT') \rightarrow K'$, returns the corresponding decapsulation key K'. but at this point the adversary cannot access the decapsulation key for $(ID^*, {C'^*}_1)$.

Guess. Adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ to send to challenger $\mathcal{C}$.

DEFINITION: OSM9-KEM-CRF is said to be ID-IND-CCA2-secure if for all probabilities polynomial time adversary $\mathcal{A}$ has a negligible advantage of $\varepsilon = \left|\Pr[b = b'] - \frac{1}{2}\right| \leq negl(\lambda)$ in winning the above game.

IV. OSM9-KEM-CRF Encapsulation Mechanisms

A. Description of OSM9-KEM Mechanism

OSM9-KEM consists of the following six algorithms:

(1) $\mathrm{Setup}(1^\lambda)$ .Input the security parameter λ, the algorithm performs the following operations.
① Choose 3 groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ of order prime r, a bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, and randomly choose generators $P_1 \in \mathbb{G}_1$, $P_2 \in \mathbb{G}_2$.
② Randomly select $s \leftarrow \mathbb{Z}_r^*$ and compute $P_{pub} = sP_1$.
③ Make $g = e(P_{pub}, P_2)$.
④ Choose hash function $H_v : \{0,1\}^* \rightarrow \{0,1\}^v$ and an identifier hid.
⑤ Output global public parameters $pp = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2, P_{pub}, g, H_v, hid)$ and master private key $msk = s$.

(2) $\mathrm{Extract}(pp, msk, ID)$.Input user identities $ID \in \{0,1\}^*$, pp and msk, KGC calculates $t_1 = H_v(ID||hid, r) + s$, if $t_1 = 0$, recalculates the master private key, otherwise calculates $sk = t_2 P_2$, where $t_2 = st_1^{-1}$.

(3) $\mathrm{KEM} \rightarrow \mathrm{Encap}(pp, ID)$.With inputs pp and ID, the algorithm performs the following.
① Let $t_1 = H_v(ID||hid, r) + s$,$Q = h_1 P_1 + P_{pub} = (h_1 + s)P_1$.
② Random select $x \leftarrow \mathbb{Z}_r^*$, let $C_1 = xQ$,$t = g^x$.
③ Let
$K = KDF_2(H_v, EC2OSP(C_1)||FE2OSP(t)||ID, l)$,
where l is the key length of DEM.
④ Output (K, C_1).

(4) $\mathrm{KenGen.ran}(sk) \rightarrow (TK, RK)$.Input sk.Randomly select $\alpha \leftarrow \mathbb{Z}_r^*$, compute $TK = \frac{1}{\alpha} sk = \frac{t_2}{\alpha} P_2$, and output conversion key TK and retrieval key $RK = \alpha$.

(5) KEM - Decap - out(pp, TK, C_1).Input pp', the user identity ID and its conversion key TK, the encapsulation portion C_1. The cloud service center computes the conversion ciphertext TCT, where $TCT = e(C_1, TK) = e(xQ, \frac{t_2}{\alpha} P_2) = e(s(h_1 P_1 + sP_1), \frac{s}{t_1 \alpha} P_2) = e(P_{pub}, P_2)^{\frac{x}{\alpha}} = g^{\frac{x}{\alpha}}$

(6) KEM - Decap - user(pp, RK, TCT). Input pp, retrieval key RK for user identity ID, transformed ciphertext TCT, user ID computes $t = (TCT)^\alpha = g^x$, and lets $K = KDF_2(H_v, EC2OSP(C_1)||FE2OSP(t)||ID, l)$, where l is the key length of DEM. Output the encapsulated key K.

Theorem 1 If SM9-KEM is ID-IND-CCA2 secure, then the above OSM9-KEM is ID-IND-CCA2 secure.

Proof In this section, OSM9-KEM is constructed based on SM9-KEM by utilizing the key blinding technique of [23]. From[3], it is known that SM9-KEM is ID-IND-CCA2 secure. Thus it can be proved similarly to [23] that OSM9-KEM is ID-IND-CCA2 secure.

B. Description of OSM9-KEM-CRF Mechanism

Based on the above OSM9-KEM mechanism, this section constructs an OSM9-KEM-CRF mechanism.

After KGC runs $\mathrm{Setup}(1^\lambda)$ to generate msk and pp, KGC first sends pp to $\mathcal{W}_{KGC}$. $\mathcal{W}_{KGC}$ Run Algorithm $\mathcal{W}_{GA}$.Setup.

(1)$\mathcal{W}_{GA}.\mathrm{Setup}(pp) \rightarrow pp'$.For pp ,$\mathcal{W}_{KGC}$ randomly selects $a, b, c \leftarrow \mathbb{Z}_r^*$ and computes $P_1' = aP_1$, $P_2' = aP_2$, $P_{pub}' = aP_{pub} = sP_1'$, $g' = e(P_{pub}, P_2)^{abc} = e(P_{pub}', P_2')^c$. Output $pp' = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1', P_2', P_{pub}', g', H_v, hid)$ but keep c. KGC carries out $\mathrm{Extract}(pp', msk, ID) \rightarrow sk$ after receiving pp' and user identity $\mathrm{Extract}(pp', msk, ID) \rightarrow sk$, sends sk to $\mathcal{W}_{KGC}$.$\mathcal{W}_{KGC}$ runs algorithm $\mathcal{W}_{KGC}$.Extract.

(2)$\mathcal{W}_{KGC}$.Extract(sk) $\rightarrow$ sk'.For sk, $\mathcal{W}_{KGC}$ computes $sk' = c \cdot sk = \frac{cs}{s+h_1} P'_x$ by the previous random selected c, where $h_1 = H_v(ID||hid, r)$.

User ID runs KEM - Encap(pp', ID) $\rightarrow$ (K, C_1) after receiving pp', sends (K, C_1) to $\mathcal{W}_{DO}$, $\mathcal{W}_{DO}$ runs algorithm $\mathcal{W}_{DO}$.KEM - Encap.

(3)$\mathcal{W}_{DO}$.KEM - Encap(K, t, C_1) $\rightarrow$ (K', C'_1). For K, t and C_1. $\mathcal{W}_{DO}$ randomly selects $f \leftarrow \mathbb{Z}_r^*$ and compute $C'_1 = fC_1$, $t' = t^f - g'^{xf}$ and $K' = KDF_2(H_v, EC2OSP(C_1')||FE2OSP(t')||ID, l)$, where l is the key length of DEM. Output (K', C'_1).

The user sends TK to $\mathcal{W}_{DO}$ after running KenGen.ran(sk') $\rightarrow$ (TK, RK), and $\mathcal{W}_{DO}$ runs Algorithm $\mathcal{W}_{DO}$.TKUpdate.

(4)$\mathcal{W}_{DC}$.TKUpdate(TK) $\rightarrow$ (TK', β).For TK, $\mathcal{W}_{DO}$ randomly selects $\beta \leftarrow \mathbb{Z}_r^*$, computes $TK' = \frac{1}{\beta} TK$, and outputs TK' but keeps β.

The cloud service center runs KEM - Decap - out($pp', T K', C'_1$) $\rightarrow$ TCT after receiving TK', sends TCT to $\mathcal{W}_{DU}$.$\mathcal{W}_{DU}$ runs algorithm $\mathcal{W}_{DU}$.Decrypt.

(5)$\mathcal{W}_{DU}$.Decrypt(TCT, β) $\rightarrow$ TCT'.For input TCT and reserved β,$\mathcal{W}_{DU}$ computes $TCT = (TCT)^{\beta}$.

After receiving TCT, DU runs KEM - Decap - user(pp', RK, TCT'), gets $t' = (g'^{\frac{fx}{\alpha}})^{\alpha} = g'^{fx}$ and $K' = KDF_2(H_v, EC2OSP(C'_1)||FE2OSP(t')||ID, l)$.

C. Security Analysis

Theorem 2: If OSM9-KEM is ID-IND-CCA2 secure, then OSM9-KEM-CRF is ID-IND-CCA2 secure and the cryptographic reverse firewalls of KGC, DO, and DU maintain functionality, weakly retain security, and weakly resist penetration.

Proof the security of OSM9-KEM-CRF is proved by the following three sections.

(1) Functionality-maintaining. Because

$$\begin{aligned} TCT &= (TCT)^{\beta} = e(C'_1, TK')^{\beta} \\ &= e\Big(fx\,(h_1 + s)\,P'_1, \tfrac{1}{\beta} TK\Big)^{\beta} \\ &= e\left(fx\,(h_1 + s)\,P'_1, \tfrac{cs}{\alpha t_1} P'_2\right) \\ &= e\left(fxP'_1, \tfrac{cs}{\alpha} P'_2\right) = e\left(P'_{pub}, \tfrac{cs}{\alpha} P'_2\right) \\ &= g'^{\frac{fx}{\alpha}} \end{aligned}$$

Thus the data user, after receiving $K' = KDF_2(H_v, EC2OSP(C'_1)||FE2OSP(t')||ID, l)$, runs $K' = KDF_2(H_v, EC2OSP(C'_1)||FE2OSP(t')||ID, l)$ and can calculate $K' = KDF_2(H_v, EC2OSP(C'_1)||FE2OSP(t')||ID, l)$ which in turn yields the encapsulation key $K' = KDF_2(H_v, EC2OSP(C'_1)||FE2OSP(t')||ID, l)$.The encapsulated key is then obtained. Thus the mechanism satisfies the maintenance functionality.

(2) ID-IND-CCA2 Security. For any tampering algorithms Setup*, Extract*, KEM - Encap*, KeyGen.ran*, KEM - Decap - out* and KEM - Decap - user* on KGCs, DOs and DUs that maintain functionality, we prove that OSM9-KEM-CRF is ID-IND-CCA2 secure by the indistinguishability of the secure game of OSM9-KEM from the secure game of OSM9-KEM–CRF. Consider the following game.

Game0. The security game sames as OSM9-KEM-CRF in Section 3.3.

Game1. Same as Game0 except that pp and msk in the Setup phase are generated by the algorithm of OSM9-KEM instead of Setup* and $\mathcal{W}_{KGC}$.Setup.

Game2. Same as Game1 except that sk and TK in Phase 1 and Phase 2 are generated by the Extract and KeyGen.ran algorithms of OSM9-KEM, not by Extract*, $\mathcal{W}_{KGC}$.Extract, KeyGen.ran* and $\mathcal{W}_{DC}$.TKUpdate.

Game3. It is the same as Game2 except that the challenge key ciphertext pair (K'_b, C'^*_1) in the Challenge phase is generated by KEM - Encap, not by KEM - Encap* and $\mathcal{W}_{DO}$.KEM - Encap.

It can be seen that Game3 is a secure game for OSM9-KEM, so it is only necessary to prove that Game0 is indistinguishable from Game3 to prove the security of OSM9-KEM-CRF.

In fact, since a, b, c is randomly chosen in Algorithm $\mathcal{W}_{KGC}$.Setup, regardless of the distribution of pp generated by Setup*, the pp obtained after the processing of the reverse firewall $\mathcal{W}_{KGC}$.Setup is uniformly random and consistent with the distribution of pp generated by Setup. Thus Game0 is indistinguishable from Game1. Also due to the extensibility of the key, it is similarly known that Game1 is indistinguishable from Game2.

For the challenge key ciphertext pair (K'_b, C'^*_1), the distribution is randomized since K'_b is generated by KDF_2. For C'^*_1, even though C'^*_1 generated by KEM - Encap* is not random, since f is randomly selected in $\mathcal{W}_{DO}.KEM - Encap$, $C_1'^*$ after $\mathcal{W}_{DO}.KEM - Encap$ post-processing is random, which is consistent with the distribution of the ciphertext generated by $KEM - Encap$, thus the indistinguishability of Game2 from Game3 can be obtained. From Game0 and Game1, Game1 and Game2, and Game2 and Game3 are indistinguishable respectively, it can be known that Game0 and Game3 are indistinguishable.

(3) Weak Security Preserving, weak Resistant to Exfiltration. According to the ID-IND-CCA2 security of OSM9-KEM-CRF, it is shown that the cryptographic reverse firewalls $\mathcal{W}_{KGC}$, $\mathcal{W}_{DU}$, and $\mathcal{W}_{DO}$ of KGC, DU, and DO are weakly preserve security. Also the proof of ID-IND-CCA2 security of OSM9-KEM-CRF shows that $\mathcal{W}_{KGC}$,$\mathcal{W}_{DU}$ and $\mathcal{W}_{DO}$ are weakly resistant to exfiltration.

V. Comparative Analysis

In order to ensure the same security strength, the traditional RSA encryption algorithm requires a larger number of key bits than the elliptic curve cipher, resulting in longer encryption time and lower monitoring efficiency in power information systems. Chen et al. [1] proposed a power information system monitoring scheme based on the SM2 algorithm, in which an SM2 encryption component is connected to the server interface, which not only determines the user's access to resources

but also records information about user activities. When the user inserts the SM2 encryption device into the client, the client uses the HTTP protocol and the digital certificate to log in to the server, and then starts to access the server. When accessing the server, the system verifies the certificate by calling the suite "iaccount", and if the verification is unsuccessful, the client's "imidware" will be automatically directed to the security support platform, which supports validation of SM2 digital certificates. The certificate is generated after verifying SM2 digital certificates, and the user's information is sent to the client, the client is redirected to the standby power supply system again through the "imidware", and then returns to the electric power secondary system by submitting a one-time signature certificate and a one-time authorization code verification and destroys the one-time certificate, and decrypts the user information by verifying the authenticity of the user signature information, and finally logs in. The user information is decrypted and finally logged into the power system.

In the above scheme, although the SM2 encryption algorithm has advantages over the RSA algorithm, however, it has some limitations in practical applications.

(1) Before using SM2 for encryption, the public key certificate of the other party must be obtained, otherwise the encryption operation cannot be performed. This requirement increases the complexity of certificate management, which in turn increases the management overhead of the overall power system.
(2) In terms of decryption, the SM2 algorithm has some complexity when decrypting on the Web side.
(3) When communicating securely across domains, it is necessary to establish a chain of trust for certificates.

Unlike SM2, SM9 is an identity-based encryption algorithm with the following advantages:

(1) No certificate management is required, effectively solving the complexity of certificate management in SM2 and significantly reducing the management burden of public key infrastructure (PKI).
(2) When decrypting on the web, there is no need for pre-registration.
(3) It only requires the publication of security parameters without a chain of trust for certificates, and the user's identity is his/her public key.

Therefore replacing the SM9 encryption algorithm with the SM2 encryption algorithm proposed by Chen et al. [1] for the power monitoring scheme not only enables more efficient data encryption, but also reduces the management cost of the power information system.

An outsourcing decryption is introduced on the basis of SM9 algorithm to further improve the decryption efficiency of SM9 algorithm in this paper. In addition, both SM2 and SM9 encryption algorithms have backdoor attacks, so this paper introduces cryptographic reverse firewall into SM9 algorithm to improve the security. Therefore the proposed OSM9-KEM-CRF based on SM9 key encapsulation is more suitable for power monitoring system.

In this section, the proposed OSM9-KEM-CRF is compared with other schemes in terms of computational overhead, communication overhead and energy consumption overhead.

TABLE I. EXECUTION TIME OF DIFFERENT CRYPTOGRAPHIC PRIMITIVES

Symbol	Operation	Times(ms)
T_{pa}	Bilinear-Pairing	13.8196
T_{pm}	ECC Point Addition	0.0110
T_e	ECC Point Exponent	12.2007
T_m	ECC Point Multiplication	2.2001
T_h	Time of hash function	0.4702

TABLE II. COMPUTATION OVERHEAD COMPARISON

Scheme	Computational overhead
[3]	$T_{pa} + T_h \approx 14.2898ms$
[4]	$2T_{pa} + T_h \approx 28.1049ms$
[11]	$T_e + T_h \approx 12.6709ms$
[20]	$T_{pa} + T_h \approx 14.2898ms$
Ours	$T_e + T_h \approx 12.6709ms$

A. Computation Cost Comparison

To evaluate the performance of our OSM9-KEM-CRF mechanism, we consistently used the Python programming language to test decryption operation times, employing 256-bit Barreto-Naehrig (BN) elliptic curves and R-ate bilinear pairings. The specific test setup was a personal desktop computer with the following configurations: 32GB of RAM, Windows 10 operating system (version 10.0.19045), Intel Cor i5-13400 CPU running at 2.5GHz, Visual Studio Code as the development environment, and the Charm cryptographic library. The notation for the operation times of cryptographic algorithms is defined in Table I.

The computational overhead of the decryption phase of each mechanism (scheme) are shown in Table II. In literature [3], one hash operation and bilinear pairing operation need to be run, and the time required is $T_{pa} + T_h \approx 14.2898ms$. In literature [4], one hash operation and two bilinear pairing operations need to be run, and the time required is $2T_{pa}+T_h \approx 28.1049ms$. In literature [11], one exponentiation operation and hash operation need to be performed, and the time required is $T_e + T_h \approx 12.6709ms$. However, in cloud services, the cloud is required to generate its own public-private key pairs, which increases the cloud's computational overhead. In the proposed OSM9 mechanism, the decryption phase needs to perform one exponential operation and one hash operation on the multiplicative group, and the total time required is $T_e + T_h \approx 12.6709ms$. In the literature [20], it needs to perform one bilinear pairing operation and one hash operation, and the time required is $T_{pa} + T_h \approx 14.2898ms$.

The comparison of the time consumed in the decryption phase of each mechanism (scheme) is shown in Fig. 2, which shows that the time consumed in decryption of this paper's mechanism and the scheme of literature [11] is lower than other schemes, and this paper's scheme does not need to generate public-private key pairs in the cloud server, which reduces the time of the cloud computation and the waiting time of the user, compared to the scheme of literature [11].

Fig. 3 and 4 show the time overhead of each algorithm in the OSM9-KEM and OSM9-KEM-CRF mechanisms, respectively, from which it is clear that the addition of the Cryptographic Reverse Firewall to the OSM9-KEM mechanism does

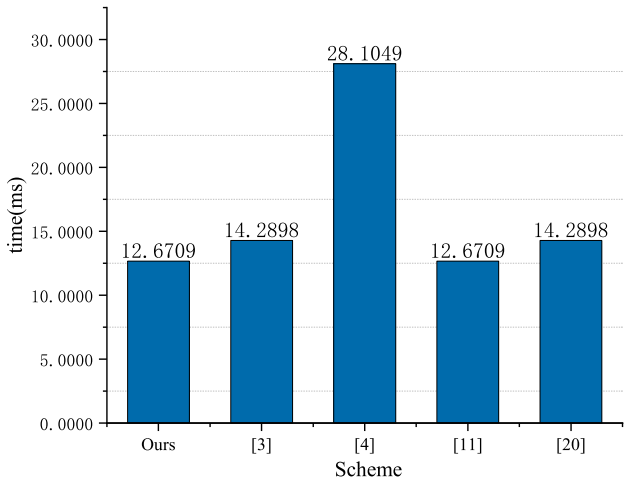

Fig. 2. Comparison of decryption time cost for users in different mechanisms (schemes).

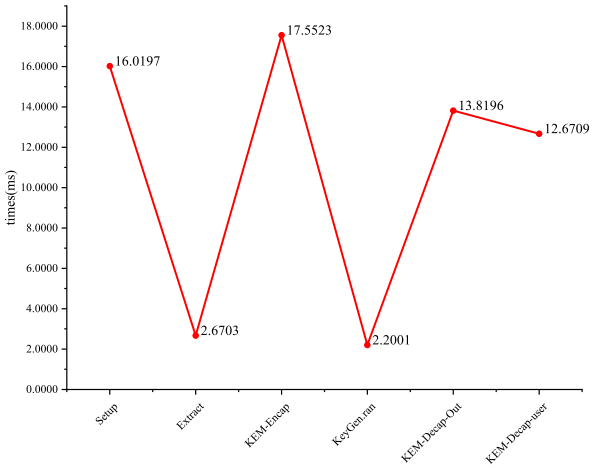

Fig. 3. The algorithm time cost used in OSM9-KEM.

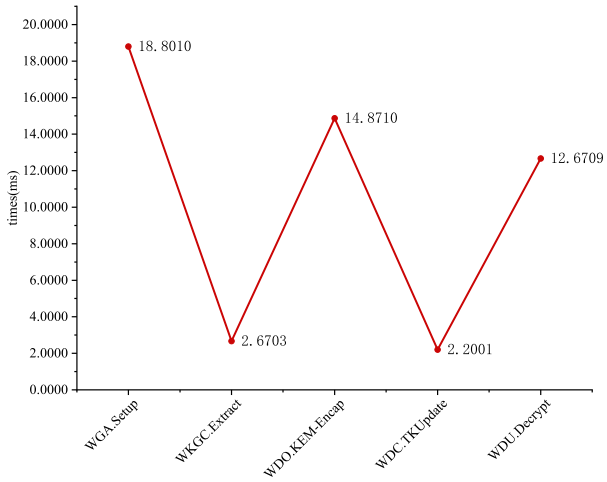

Fig. 4. OSM9-KEM-CRF algorithm time overhead

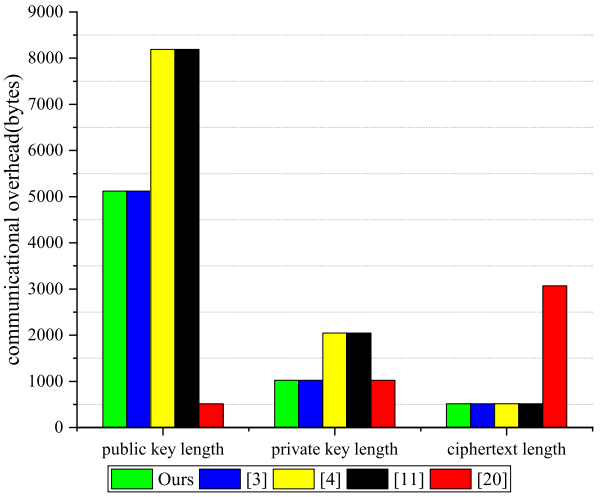

Fig. 5. Comparison of communication costs for different schemes.

not have a significant impact on the time overhead, but greatly increases the security.

B. Communication Cost Comparison

In terms of communication overhead, $|\mathbb{G}_1|$,$|\mathbb{G}_2|$,$|\mathbb{G}_T|$, $|\mathbb{Z}_p|$ denote the size of the elements in the $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ and $\mathbb{Z}_p$, respectively. Specifically, the 256-bit BN curve [24] is used, that is $|\mathbb{G}_1|$=512bit,$|\mathbb{G}_2|$=1024bit,$|\mathbb{G}_T|$=3072bit, $|\mathbb{Z}_p|$=256bit. Table III compares the bit requirements of the key encapsulation mechanism proposed in this paper with those of other schemes across public parameters, private keys, and ciphertexts. Furthermore, as illustrated in Fig. 5, our mechanism demonstrates a significant reduction in communication overhead for public keys, private keys, and ciphertexts.

C. Energy Cost Comparison

In terms of energy overhead, the calculation method in [25] is used, with the formula $vol \times cur \times T$, where vol represents the voltage, cur represents the current, T represents the execution time $(vol = 3V, cur = 1.8\mu A)$, and the energy consumed for sending $1bit$ messages is 0.72μJ, and the energy consumed for receiving messages is 0.81μJ.In literature [3], the energy overhead related to computation is $vol \times cur \times (T_{pa} + T_h) \approx 77.1649\mu$J, and the energy overhead related to communication is $|\mathbb{G}_2| \times 0.81\mu\text{J} + |\mathbb{G}_1| \times 0.72\mu\text{J} \approx 1198.0800\mu$J, so the total energy overhead is 1275.2449μJ; in literature [4], the energy overhead related to computation is $vol \times cur \times (2T_{pa} + T_h) \approx 151.7664\mu$J, and the energy overhead related to communication is $2\,|\mathbb{G}_2| \times 0.81\mu\text{J} + |\mathbb{G}_1| \times 0.72\mu\text{J} \approx 2027.5200\mu$J, so the total energy overhead is 2179.2864μJ; in literature [11], the energy overhead related to computation

TABLE III. COMMUNICATION OVERHEAD OF DIFFERENT SCHEMES

Scheme	Public parameter length	Private key length	Ciphertext length
Ours	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + \|\mathbb{G}_T\| \approx 5120$bits	$\|\mathbb{G}_2\| \approx 1024$bits	$\|\mathbb{G}_1\| \approx 512$bits
[3]	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + \|\mathbb{G}_T\| \approx 5120$bytes	$\|\mathbb{G}_2\| \approx 1024$bytes	$\|\mathbb{G}_1\| \approx 512$bits
[4]	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + 2\|\mathbb{G}_T\| \approx 8192$bits	$2\|\mathbb{G}_2\| \approx 2048$bits	$\|\mathbb{G}_1\| \approx 512$bits
[11]	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + 2\|\mathbb{G}_T\| \approx 8192$bits	$2\|\mathbb{G}_2\| \approx 2048$bits	$\|\mathbb{G}_1\| \approx 512$bits
[20]	$\|\mathbb{G}_1\| \approx 512$bits	$\|\mathbb{G}_2\| \approx 1024$bits	$3\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + 2\|\mathbb{Z}_p\| \approx 3072$bits

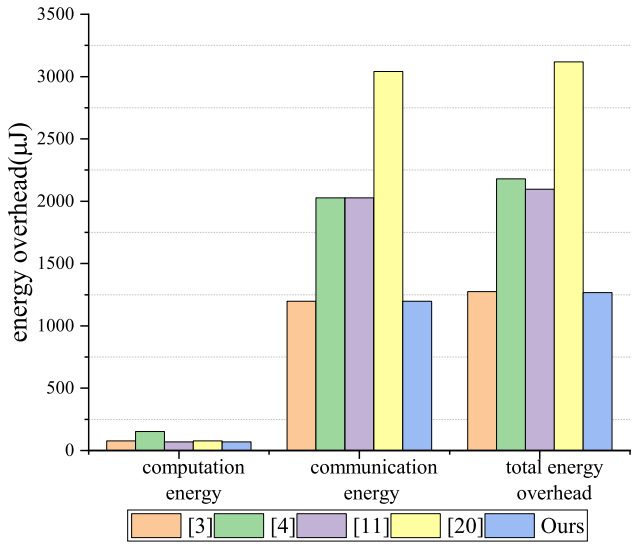

Fig. 6. Comparison of energy consumption of different schemes.

is $vol \times cur \times (T_e + T_h) \approx 68.4228\mu$J, and the energy overhead related to communication is $2|\mathbb{G}_2| \times 0.81\mu\text{J} + |\mathbb{G}_1| \times 0.72\mu\text{J} \approx 2027.5200\mu$J, so the total energy overhead is 2095.9428μJ; in literature [20], the computation-related energy overhead is $vol \times cur \times (T_{pa} + T_h) \approx 77.1649\mu$J and the communication-related energy overhead is $|\mathbb{G}_2| \times 0.81\mu\text{J} + (3|\mathbb{G}_1| + |\mathbb{G}_2| + 2|\mathbb{Z}_p|) \times 0.72\mu\text{J} \approx 3041.28\mu$J, thus the total energy overhead is 3118.4449μJ; in this paper, the computation-related energy overhead is $vol \times cur \times (T_e + T_h) \approx 68.4228\mu$J and the communication-related energy overhead is $|\mathbb{G}_2| \times 0.81 + |\mathbb{G}_1| \times 0.72 \approx 1198.0800\mu$J, thus the total energy overhead is 1266.5028μJ. The comparison of energy overhead of each mechanism (scheme) is shown in Fig. 6. In power monitoring system, less energy overhead is especially important in power system due to limited equipment resources, in the above comparison, the mechanism in this paper has less energy overhead and is more suitable for power system, and it incorporates a reverse firewall to block backdoor attacks and improve the security of the system.

VI. CONCLUSIONS

This paper proposes an SM9 key encapsulation mechanism that supports outsourced decryption and CRF, improving the efficiency and security of the SM9 key encapsulation mechanism. The proposed OSM9-KEM-CRF mechanism outsources the tedious bilinear mapping calculation in the decryption process to cloud servers, and cloud servers do not need to generate its own public-private key pairs, improving the efficiency of the mechanism. In addition, the key encapsulation mechanism adds the cryptographic reverse firewall function for KGC and users respectively, and the deployment of CRF also makes the mechanism resistant to backdoor attacks, resistant to information leakage, protects user privacy, and improves the security of the key encapsulation mechanism. The security proof and comparative analysis comparison show that the mechanism is more suitable for the power monitoring system.

In future work, in order to further reduce the computational burden on users and enrich the functionality of the SM9 algorithm, we will research how to use smart contracts to verify the correctness of outsourced decryption, thereby further reducing users' computational overhead. In addition, the SM9 algorithm will be functionally extended to construct an attribute based encryption scheme based on SM9, achieving fine-grained access control of encrypted data in the cloud.

ACKNOWLEDGMENT

This research was funded by Science and Technology of Yunnan Power Grid (YNKJXM20222419, YNKJXM20222425).

REFERENCES

[1] F. Chen, H. Zou, Y. Wu, X. Liu, D. Liang, Design of power information security monitoring system based on SM2 cryptosystem, Electronic Design Engineering 30 (05) (2022) 100-103+108.

[2] F. Yuan; Z.H. Cheng, Review of SM9 identity cipher algorithm, Information Security Research, 2 (11) (2016) 1008-1027.

[3] Z. Chen, Security analysis of SM9 key agreement and encryption, in:Information Security and Cryptology: 14th International Conference, Inscrypt 2018, Fuzhou, China, December 14-17, 2018, Revised Selected Papers ,Proceedings, Springer, 2019, pp. 3-25.

[4] J. Lai,X. Huang,D. He ; W. Wu, Security analysis of state secret SM9 digital signature and key encapsulation algorithm, Science China: Information Science,51 (11) (2021) 1900-1913.

[5] H. Ji, H. Zhang, L. Shao, D. He, M. Luo, An efficient attribute-based encryption scheme based on SM9 encryption algorithm for dispatching and control cloud, Connection Science,33 (04) (2021) 1094-1115.

[6] M.D. Wang, W.G He, J. Li, R. M, Optimized design of state-secret SM9 algorithm R-ate pair computation, Communication Technology, 53 (11) (2020) 2241-2244.

[7] J. Lai, Y. Mu, F. Guo, Efficient identity-based online/offline encryption and signcryption with short ciphertext, International Journal of Information Security, 16 (2017) 299-311.

[8] Y. Sun, Z. Lu, J. Zhao, M.Q. Fan, Research on SM9-IBE encryption scheme based on online/offline technology, Journal of Qiqihar University (Natural Science Edition), 39 (01) (2023) 26-30.

[9] C. Peng, M. Luo, L. Li, K.K.R. Choo, D. He, Efficient certificateless online/offline signature scheme for wireless body area networks, IEEE Internet of Things Journal, 8 (18) (2021) 14287-14298, .

[10] P. Liu, Q. He, W.Y. Liu, X. Cheng, A CP-ABE scheme supporting revocation of attributes and outsourced decryption, Information Network Security, 20 (03) (2020) 90-97.

[11] Liu, K. An OSM9 identity key encapsulation mechanism supporting decryption outsourcing, Industrial Technology Innovation, 10 (01) (2023) 106-113.

[12] C. Patsakis, A. Charemis, A. Papageorgiou, D. Mermigas, S. Pirounias, The market's response toward privacy and mass surveillance: The Snowden aftermath, Computers Security, 73 (2018) 194-206.

[13] I. Mironov, N. S. Davidowitz, Cryptographic reverse firewalls, in: Advances in Cryptology – EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II, Proceedings, Springer, 2015, pp. 657-686.

[14] R. Chen, Y. Mu, G. Yang, W. Susilo, F. Guo,M. Zhang ,Cryptographic reverse firewall via malleable smooth projective hash functions, in: Advances in Cryptology – ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I, Proceedings, Springer, 2016, pp. 844-876.

[15] Y. Zhou, Y. Guan, Z. Zhang, F. Li, Cryptographic reverse firewalls for identity-based encryption, in:Frontiers in Cyber Security:Second International Conference, FCS 2019, Xi'an, China, November 15–17, 2019, Proceedings, Springer, 2019, pp. 36-52.

[16] Y. Zhou, J. Guo, F. Li, Certificateless public key encryption with cryptographic reverse firewalls, Journal of Systems Architecture, 109 (2020) 101754.

[17] Y. Zhou, Z. Hu, F. Li, Searchable public-key encryption with cryptographic reverse firewalls for cloud storage, IEEE Transactions on Cloud Computing, 11 (01) (2021) 383-396.

[18] Y. Zhou, L. Zhao, Y. Jin, F. Li, Backdoor-resistant identity-based proxy re-encryption for cloud-assisted wireless body area networks, Information Sciences ,604 (2022) 80-96.

[19] C. Jin, Z. Chen, W. Qin, K. Sun, G. Chen, L. Chen, A Blockchain-Based Proxy Re-Encryption Scheme with Cryptographic Reverse Firewall for IoV, International Journal of Network Management, 34 (2024) 80-96.

[20] H. Xiong, Y. Lin, T. Yao, An SM9 logo encryption scheme supporting equation testing and cryptographic reverse firewall, Computer Research and Development ,61 (04) (2024) 1070-1084,

[21] J. Li, Y. Fan, X. Bian, Q. Yuan, Online/Offline MA-CP-ABE with Cryptographic Reverse Firewalls for IoT, Entropy, 25 (4) (2023) 616.

[22] MH. Ma, R. Zhang, G. Yang, Z. Song, S. Sun , Y. Xiao, Concessive online/offline attribute based encryption with cryptographic reverse firewalls—Secure and efficient fine-grained access control on corrupted machines, in: Computer Security:23rd European Symposium on Research in Computer Security, ESORICS 2018, Barcelona, Spain, September 3-7, 2018, Proceedings, Part II, Proceedings, Springer, 2018, pp. 507-526.

[23] M. Green, S. Hohenberger, B. Waters, Outsourcing the decryption of abe ciphertexts, In: Proceedings of the 20th USENIX Conference on Security (USENIX'11), USENIX Association, 2011, pp.1–11.

[24] G.C.C.F Pereira, Jr.M.A Simplício, M. Naehrig, P.S.L.M. Barreto, A family of implementation-friendly BN elliptic curves, Journal of Systems and Software, 84 (08) (2011) 1319-1326.

[25] J. Du, C. Dai, P. Mao, W. Dong, X. Wang, Z. Li, An Efficient Lightweight Authentication Scheme for Smart Meter, Mathematics, 12 (8) (2024) 1264.