Privacy-Preserving Electricity Theft Detection based on Blockchain

Zhiqiang Zhao, Yining Liu, *Member IEEE,* Zhixin Zeng, Zhixiong Chen, Huiyu Zhou

***Abstract*—In most electricity theft detection schemes, consumers' power consumption data is directly input into the detection center. Although it is valid in detecting the theft of consumers, the privacy of all consumers is at risk unless the detection center is assumed to be trusted. In fact, it is impractical. Moreover, existing schemes may result in some security problems, such as the collusion attack due to the presence of a trusted third party, and malicious data tampering caused by the system operator (SO) being attacked. Aiming at the problems above, we propose a blockchain-based privacy-preserving electricity theft detection scheme without a third party. Specifically, the proposed scheme uses an improved functional encryption scheme to enable electricity theft detection and load monitoring while preserving consumers' privacy; distributed storage of consumers' data with blockchain to resolve security problems such as data tampering, etc. Meanwhile, we build a long short-term memory network (LSTM) model to perform higher accuracy for electricity theft detection. The proposed scheme is evaluated in a real environment, and the results show that it is more accurate in electricity theft detection within acceptable communication and computational overhead. Our system analysis demonstrates that the proposed scheme can resist various security attacks and preserve the consumer's privacy.**

***Index Terms*—Privacy protection, electricity theft detection, smart grid, blockchain, long short-term memory network (LSTM).**

I. Introduction

SMART grid (SG) is an advanced grid integrating smart technology, which uses smart meters (SMs) to collect, analyze and process fine-grained power consumption data from consumers to manage energy effectively [1]. While the smart grid brings convenience, it also brings serious challenges [2]. For one thing, the communication of the smart grid is exposed to potential malicious attacks, such as data tampering attack and false data injection. If these malicious attacks cannot be resisted, the smart grid will be unable to operate normally [3]. For another thing, electricity theft has become a widespread phenomenon in the grid. Annual economic losses due to electricity theft are estimated to be about 170 million dollars in the United Kingdom [4] and 6 billion dollars in the United States [5]. Meanwhile, electricity theft can also seriously affect energy management and endanger the normal operation of the smart grid [6].

Since the smart grid has access to consumer's fine-grained power consumption data, the traditional machine learning model [7] and deep learning model [8] based on big data have achieved good performance. However, directly giving fine-grained power consumption data of consumers to the SO raises serious privacy issues [9]. Meanwhile, as the security and privacy of data are becoming more and more concerned, related laws and regulations have been proposed, such as the General Data Protection Regulations (GDPR) in Europe, and the utilities' disregard for privacy aspects could lead to strong consumer objection and significant curtailment of service deployment [10]. Therefore, there is an urgent demand for a privacy-preserving electricity theft detection scheme.

Although existing schemes are beginning to consider the privacy of consumers' power consumption data during the electricity theft detection process, most schemes have serious challenges. On the one hand, a serious threat is the potential leakage of consumers' privacy due to the presence of a trusted third party. In [11], the model requires a fully trusted third party to perform the detection using the original consumers' data. However, it is difficult to guarantee that the third party is fully trustworthy in reality, so consumers' privacy is still at risk of being compromised. In [12], the scheme requires a fully trusted key distribution center. However, once the SO colludes with the key distribution center, then the SO can get the consumers' raw power consumption data, which leads to consumers' privacy leakage. On the other hand, the security of data and smart grid is not considered. In [11], if the trustworthy detection center is maliciously attacked, then it will possibly lead to malicious data tampering. In [12], this scheme does not verify the legitimacy of the transmitted data, so it is unable to resist data falsification and forgery attacks. The existing schemes do not consider the security of the smart grid in operation and data tampering due to centralized data storage when performing electricity theft detection, thus making it impossible to achieve electricity theft detection. Therefore, how to accomplish the security of smart grid operations and consumers' privacy while utilizing consumers' power consumption data is a major challenge of current research.

In this paper, we aim to achieve more secure electricity theft detection and load monitoring without the involvement of a third party. The main contributions of this work are threefold:

1) We propose a blockchain-based electricity theft detection scheme, which uses the distributed storage of blockchain to solve security problems such as data tampering of centralized storage, etc.
2) We improve the functional encryption scheme [12] to enable privacy-preserving electricity theft detection and load monitoring without a trusted key distribution center, which eliminates potential security and privacy problems caused by a trusted third party.
3) We build an electricity theft detection model based on

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.
Manuscript received *; revised *.

TABLE I
THE COMPARISON OF RELATED WORK.

	Joker et al. [7]	Wen et al. [13]	Yao et al. [11]	I.Ibrahem et al. [12]	Richardson et al. [14]	Nabil et al. [15]
Technique adopted	SVM	Federal Learning	CNN	FNN	DBSCAN	1-D CNN
Attack Defense	No	No	No	No	No	No
Grid monitoring	Yes	No	Yes	Yes	No	Yes
Third-party	No	Yes	Yes	Yes	No	No

long short-term memory networks that is more suitable for processing time-series data, and the model parameter settings are analyzed to obtain higher performance.

The remainder of this paper is organized as follows. In Section II, we review the related work. Section III illustrates the related knowledge. In Section IV, we define the system model and design goals. Section V presents the proposed scheme. Experimental results and system characterization are presented in Sections VI and VII, respectively. Finally, the paper is summarized in Section VIII.

II. Related Work

In this section, we briefly review recent research work on electricity theft detection schemes in the smart grid and distributed blockchain-based smart grid framework.

Currently, due to the seriousness of the electricity theft problem and the importance of privacy-preserving, we focus on electricity theft detection schemes with privacy-preserving, which can be broadly classified into two categories with or without the participation of a third party. The comparison of the related work is given in Table I.

In the case of schemes where a third party is involved, the third party is used to perform tasks such as distributing keys or performing electricity theft detection, etc. Wen et al. [13] proposed a privacy-preserving federal learning framework consisting of a data center, a control center, and multiple detection stations, which requires a high cost to complete the system. Moreover, the author's scheme does not consider other functional requirements such as load monitoring. Yao et al. [11] proposed to send the encrypted data of SMs to a fully trusted detection center to decrypt and then detect using the convolutional Neural Network (CNN) model, meanwhile, SMs send the encrypted data to an untrusted center that aggregates power consumption data for load monitoring. In [12], Ibrahem et al. proposed to use functional encryption and the feed-forward neural network (FNN) to perform electricity theft detection and privacy protection under the condition that the key distribution center is fully trusted. All of the above schemes assume that the third party is trustworthy, but in practice, consumers' privacy can still be compromised such as once the third party colludes with other entities. Untrustworthy third parties have caused the above-mentioned problems in other areas as well [16], so it is important to eliminate the risks associated with the presence of untrustworthy third parties.

In schemes where no third party is required, the scheme is performed by only two entities, SM and SO. Joker et al. [7] proposed to use the clustering method support vector machine (SVM) to monitor consumption pattern anomalies and identify suspicious consumers in the case of low sampling of consumers' power consumption data. However, this scheme is difficult to resist malicious attacks, such as replay attacks, fake data injection, etc. In [14], the Euclidean distance between the normalized photovoltaic power output of any two installations in the region in a day is calculated by homomorphic encryption. Then the Euclidean distances are clustered to analyze the anomalous users. However, this scheme detects energy theft from the perspective of energy output, and when a smart meter is tampered with due to external attacks, it can no longer be detected properly. Meanwhile, the author's scheme cannot obtain the sum of power consumption in the region for load monitoring. Nabil et al. [15] proposed a CNN machine learning model based on secure two-party computation protocols using arithmetic and binary circuits. This scheme requires high computation and communication overhead to complete the detection of a consumer, which takes at least 35 minutes for detection and a minimum of 1375 MB for communication overhead. None of the above schemes consider the problem of ensuring the operational security of the smart grid when performing electricity theft detection, such as data tampering problem when SO is maliciously attacked. Therefore a more secure detection model with acceptable computation and communication overhead is needed.

To ensure the security of the smart grid, in [17], Liang et al. proposed a new distributed blockchain-based protection framework to enhance the self-defense of modern power system. In [18], the authors design a blockchain-based platform to prevent user data from being tampered with and propose a multifaceted mechanism to protect user privacy. In [19], Hamouda et al. proposed a blockchain-based comprehensive transactive energy market framework that enables a safer and fairer electricity market. Fan et al. [20] proposed a decentralized privacy-preserving data aggregation scheme for smart grid based on blockchain, which uses the Paillier cryptographic algorithm to aggregate consumers' power consumption data. In [21], the authors propose a new blockchain-based strategy for interconnected microgrids energy trading that enhances the security and transparency of the platform. In [22], the authors propose an efficient and robust blockchain-based multidimensional data aggregation scheme in smart grid to resist more internal and external attacks. Chen et al. [23] proposed a blockchain-based framework to prevent energy market failures caused by dishonest participants.

There are many recent studies that consider distributed blockchain-based smart grid framework can secure the grid. Meanwhile, it is also a good solution for electricity theft detection, and to advance the state of the art, we propose a

blockchain-based privacy-preserving electricity theft detection scheme, which will be further explained and evaluated in the following sections.

III. Preliminaries

A. Secure Aggregation

Bonawitz et al. [24] proposed a secure aggregation scheme where the server can only see the gradient after the aggregation is completed and cannot know the private true gradient value of each user. Unlike the original text, the proposed scheme uses elliptic curve Diffie-Hellman key agreement. The steps of secure aggregation are as follows:

1) Key agreement between arbitrary SMs: Each SM_i negotiates key masks with each other.

- KA.Setup$(\lambda) \rightarrow (E, G, g, p, q, H)$: The setup algorithm takes as input the security parameters λ. Then it outputs cyclic additive group G of prime order q, a basis point g, a hash function H, and an elliptic curve E on $GF(p)$ as well as a large prime number p.
- KA.Gen$(E, G, g, p, q, H) \rightarrow (x, xg)$: Each user chooses a random $x \in Z_q$ as own secret key s_u^{sk} and calculates xg as the public key s_u^{pk}.
- KA.Agree$(x_u, x_v g) \rightarrow s_{u,v}$: After receiving the public key $x_v g$ from user v, user u uses its own secret key x_u to generate $s_{u,v} = H(x_u(x_v g))$.

2) Generating masks for aggregation: A mask is generated by key agreement between arbitrary users. Assume that all users form a user set $\mathcal{U}$ in order and each user $u \in \mathcal{U}$ computes:

$$y_u = x_u + \sum_{v \in \mathcal{U}, u<v} s_{u,v} - \sum_{v \in \mathcal{U}, u>v} s_{v,u} \tag{1}$$

where $u < v$ represents the users whose serial number is less than v, by the same token, we can get $u > v$.

Each user u sends y_u to the server, the server computes Eq. (2) to securely aggregate the secret keys.

$$\begin{aligned} z &= \sum_{u \in \mathcal{U}} y_u \\ &= \sum_{u \in \mathcal{U}} (x_u + \sum_{v \in \mathcal{U}, u<v} s_{u,v} - \sum_{v \in \mathcal{U}, u>v} s_{v,u}) \\ &= \sum_{u \in \mathcal{U}} x_u \end{aligned} \tag{2}$$

B. Boneh-Lynn-Shacham Short Signature

Boneh-Lynn-Shacham (BLS) short signature [25] is a signature algorithm that enables signature aggregation and speeds up block verification, which is divided into three phases: key generation, signature, and verification.

1) Key generation: Sampling random number $x \in Z_q^*$ as private key and calculate the public key $PK = x \cdot g$.
2) Signature: The message m is mapped to a point $H(m)$ in the cyclic group G_1. Generating signature $\delta = x \cdot H(m)$.
3) Verification: If $e(\delta, g) = e(H(m), PK)$, where $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map, then the signature is verified. Otherwise fails.

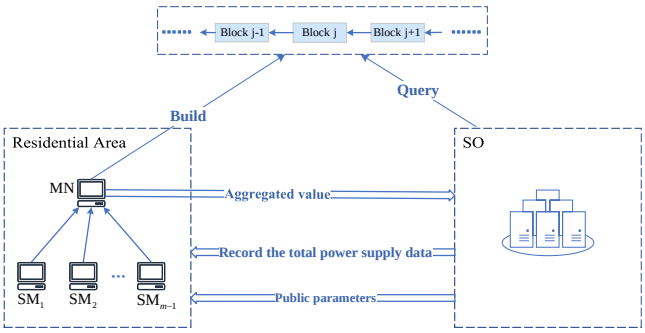

Fig. 1. System Model.

IV. System Model and Design Goals

This section focuses on the construction of the system model and threat model as well as describes our design goals.

A. System Model

As shown in Fig. 1, the model of our system scheme includes smart meters (SMs) in the residential area (RA), system operator (SO) and distribution transformer meters (DTMs). The function of each entity is described below.

1) Smart meter (SM): SM is an electricity meter that sends the consumer's power consumption data to MN periodically (e.g., every 30 minutes) after implementing a predefined privacy-preserving scheme.
2) Mining node (MN): The MN is a smart meter selected by the votes of all SMs in each residential area, it is responsible for verifying the legitimacy of the data, aggregating the encrypted data reported by SMs, and creating blocks to record power consumption data. If the MN goes down, all SMs will continue to vote for a new MN. If a malicious SM wants to become an MN, it needs to control at least 50% of the SMs in the entire network to be elected as MN, but this is unrealistic.
3) System operator (SO): The SO can generate system parameters and read the consumer's encrypted power consumption data through blockchain as well as get the real-time total power consumption $\sum_i E_{SM_i}(t)$ of the area sent by MN, which are used for power consumption analysis and energy management. The SO uses a distribution transformer meter (DTM) to record the total power supply data $E_{DTM}(T)$ for the residential area during the electricity theft detection period in order to judge the existence of electricity theft and perform electricity theft detection.

B. Threat Model

For the system model proposed in the previous sub-section, we consider the threat from three aspects: consumers, the SO and external attackers.

1) Consumers: Malicious consumers may falsify their electricity consumption data to reduce their bills. Also, they may collude with other consumers or SOs to infer sensitive information about the victimized consumers. In

addition, malicious consumers may deny their transmitted data when they are detected. With respect to MN, it may also maliciously falsify the data reported by SMs.

2) SO: The SO is assumed to be honest but curious, i.e., it performs operations according to the protocol, but it may attempt to obtain fine-grained power consumption data from consumers to analyze valuable information.
3) External attackers: External attackers may attempt to eavesdrop on consumer communications to obtain consumer data, and may also forge malicious data to harm the SO, as well as initiate attacks on the SO to tamper with stored data.

Therefore, the scheme aims to achieve smart grid can resist malicious attacks and preserve consumers' privacy while still enabling energy management and electricity theft detection.

C. Design Goals

In order to protect the security and privacy of consumers' data without relying on third-party organizations, the proposed scheme should achieve the following design goals:

1) Privacy-preservation: For any one consumer, their original power consumption data is not obtainable by DTM, SO and other consumers, and no entity can infer any private information from the encrypted data.
2) Confidentiality: Consumers' data is encrypted for transmission, storage, aggregation and theft detection, so that the original consumers' data cannot be recovered even if entities collude with each other.
3) Data unforgeability and non-repudiation: The consumer's encrypted data is signed and then transmitted to ensure that the data cannot be forged, while the transmission information is recorded in the blockchain to achieve data non-repudiation and data unforgeability.
4) Resist collusive attacks: The proposed scheme can resist the attack that smart grid entities collude with each other to obtain consumers' power consumption data.

V. The Proposed Scheme

Our scheme consists of five phases: (1) system initialization phase; (2) reporting phase; (3) aggregation phase; (4) judgement phase; (5) electricity theft detection phase. The notations are listed in Table II.

A. Overview

The main process of our scheme is summarized as follows:

- In the initialization phase, SO divides the residential area RA and generates the system parameters as well as parameters of the first layer of the neural network. The SMs in each detection region select the MN by Byzantine fault-tolerant consensus mechanism [26], while the SM generates encryption and decryption keys.
- In the reporting phase, each SM encrypts the power consumption data $r(t)$ during the detection period $T = \{t_1, t_2, \cdots t_d\}$, then signs and sends encrypted data to the MN.

TABLE II
Notations

Notation	Description
$E_{DTM}(t)$	Power supply data for a residential area
$\sum_i E_{SM_i}(t)$	Total of uploaded data for all SMs
SM_i	i-th smart meter
$\mathbb{C}_i[t]$	Encrypted reading of SM_i at time t
$\mathcal{U}$	The set of SMs in the detection region
W	The first layer's weights of the model
T	Electricity theft detection period
TS_t^i	Timestamp of SM_i
δ_i	Signature of SM_i
DA	Decryption keys for aggregating readings
DW_i	Decryption keys for electricity theft detection
$\lvert RA\rvert$	Number of residential areas
m	Number of smart meters in the detection area
d	Number of readings for electricity theft detection period

- In the aggregation phase, MN verifies the legitimacy of the data, then constructs blocks and aggregates the power consumption data through the Merkle tree.
- In the judgement phase, SO judges whether there is electricity theft in a region based on the difference between the DTM statistics and the aggregated data of MN within the tolerance range.
- In the electricity theft detection phase, SO reads the encrypted data from the blockchain that is reported by each SM in the suspected electricity theft area during the theft detection period. The encrypted data are decrypted (still in ciphertext state after decryption) and then fed to the detection model to identify the electricity theft consumers.

B. System Initialization

System initialization includes three parts. First, SO generates the parameters of the system and the first layer's weights of the model, and delineates $|RA|$ residential areas with m SMs in each detection area. Second, all SMs in the region reach consensus to choose the MN. Third, Each SM generates its own keys.

1) System parameters generation:

- **Step 1:** The SO generate (q, g, G, G_1, e) where G and G_1 are two cyclic additive group of prime order q, g is a generator of G.
- **Step 2:** The SO generates (G_2, q, g_2) where G_2 is a cyclic additive group of prime order q and generator g_2 based on elliptic curves.
- **Step 3:** The SO chooses a full-domain hash function $H_1 : \{0,1\}^* \rightarrow G^2$ and a hash function H_2.
- **Step 4:** The SO publishes public parameters $(q, g, g_2, G, G_1, G_2, e, H_1, H_2)$.

2) The first layer's weights of the model: The SO trains the electricity theft detection model based on historical honest and malicious consumers' power consumption data, and then saves the weight of the first layer of the network, the weight $W = [w_1^{\mathrm{T}}, w_2^{\mathrm{T}}, \cdots w_n^{\mathrm{T}}]$ can be represented as:

$$W = \begin{bmatrix} w_1[1] & w_2[1] & \cdots & w_n[1] \\ w_1[2] & w_2[2] & \cdots & w_n[2] \\ \vdots & \vdots & \cdots & \vdots \\ w_1[d] & w_2[d] & \cdots & w_n[d] \end{bmatrix}_{d \times n}$$

where d is the number of power reporting in the electricity theft detection period $T = \{t_1, t_2, \cdots t_d\}$ and n is the number of neurons in the first layer of the neural network, n should be fewer than the number of inputs d, because if $n \geq d$, the SO will calculate the consumers' fine-grained power consumption data, since d unknowns in d equations may be solved to obtain the data.

3) Key Generation: SM generates the encryption keys and decryption keys. All SMs $\mathcal{U} = \{SM_1, SM_2, \cdots, SM_m\}$ cooperate using the secure aggregation algorithm to generate a decryption public key DA for aggregating the power consumption readings of all SMs. Meanwhile, each SM generates electricity theft detection public keys DW.

- Secret key generation: SM_i selects a random number $x_i \in Z_q$ as the secret key for signing and key negotiation and selects $s_i \in Z_q^2$ as the secret key for encryption.
- Generation of DA: Arbitrary SMs negotiate key masks among themselves and the MN performs secure aggregation to generate decryption public keys DA.
  Step 1: Each SM_i calculates and publishes the public key $s_i^{pk} = x_i g_2$.
  Step 2: Each SM_i receives the public keys s_o^{pk} of other SMs and then calculates $sk_{i,o} = x_i s_o^{pk} (i \in \mathcal{U}; o \in \mathcal{U}, o \neq i)$. Fig. 2 shows an example of four SMs perform key masks agreement and generate DA.
  Step 3: Each SM_i calculates y_i and sends the results to MN for aggregation by Eq. (3).

$$y_i = s_i + \sum_{o \in \mathcal{U}, i<o} sk_{i,o} - \sum_{o \in \mathcal{U}, i>o} sk_{o,i} \tag{3}$$

  Step 4: MN aggregates its own y and the results sent by other SMs, as shown in Eq. (4).

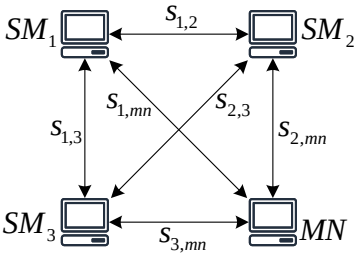

Fig. 2. Example of four SMs key masks agreement and generate DA.

$$\begin{aligned} DA &= \sum_{i \in \mathcal{U}} y_i \\ &= \sum_{i \in \mathcal{U}} (s_i + \sum_{o \in \mathcal{U}, i<o} sk_{i,o} - \sum_{o \in \mathcal{U}, i>o} sk_{o,i}) \\ &= \sum_{i \in \mathcal{U}} s_i \in Z_q^2 \end{aligned} \tag{4}$$

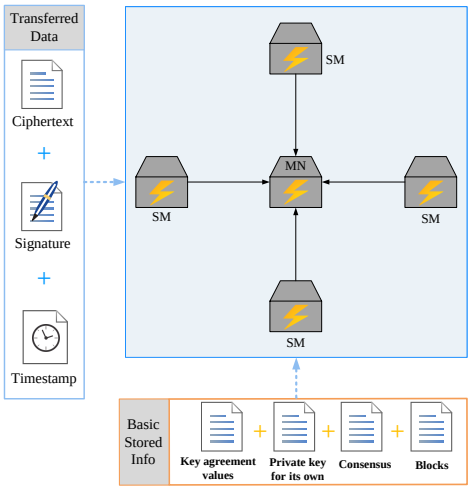

Fig. 3. The data within each smart meter node consists of basic stored information and primary transmitted data.

- Generation of DW: SO publishes the weights of the first layer network of the electricity theft detection model to each SM_i, and each SM_i generates decryption public keys to enable theft detection without obtaining the original power consumption data.
  Step 1: Each SM_i generates a timestamp TS_t^i of the current detection time $T = \{T_1, T_2, \cdots T_d\}$ by Eq. (5).

$$TS_t^i = H_1(T_t) \in G^2, t = \{1, 2, \cdots, d\} \tag{5}$$

  Step 2: Each SM_i generates $c_{th}, c = \{1, 2, \cdots, n\}$ decryption public key by Eq. (6):

$$DW_{ci} = \sum_{t=1}^{d} w_c[t](s_i^\top \cdot TS_t^i) \in G \tag{6}$$

  Step 3: Each SM_i generates decryption public keys by Eq. (7).

$$DW_i = \{DW_{1i}, DW_{2i}, \cdots, DW_{ni}\} \tag{7}$$

C. Reporting Phase

In the reporting phase, each SM_i encrypts its power consumption readings and then performs signature operations.

- **Step 1:** For each electricity theft detection period T, each SM_i encrypts its power consumption readings by Eq. (8).

$$\mathbb{C}_i[t] = (s_i^\top \cdot TS_t^i) + r_i[t]g \in G \tag{8}$$

- **Step 2:** Each SM_i computes the public key $PK_i = x_i \cdot g_2$ and then generates the BLS short signature by Eq. (9), TS_t^i is the current timestamp to prevent replay attack.

$$\delta_i = x_i \cdot H_2(\mathbb{C}_i[t] || TS_t^i || DW_i || PK_i) \tag{9}$$

- **Step 3:** Each SM_i sends $\mathbb{C}_i[t] || TS_t^i || DW_i || \delta_i || PK_i$ to MN. The data within each SM_i node consists of basic storage information and primary transmission data, as shown in the Fig. 3.

D. Aggregating Phase

Efficient message propagation methods are important building blocks for various networks [27], and in the proposed scheme, the SM sends messages directly to the MN, which is responsible for broadcasting and aggregating the total area power consumption. After receiving the data from the SMs, first, the $MN_j, j \in |RA|$ in the residential area RA_j verifies the signature and timestamp, after the verification is passed. MN_j generates the Merkle tree and then creates the block though Byzantine fault-tolerant consensus mechanism, the block head stores the timestamp, the hash of the previous block, and the Merkle tree root hash, and the block body stores the encrypted data and decryption Keys. After that, MN_j aggregates the ciphertext and decrypts it to get the total power consumption of all SMs at the current time. Fig. 4 shows the blockchain structure of the proposed scheme. The detailed steps are as follows:

- **Step 1:** MN_j verifies signature and timestamp. If the Eq. (10) and $TS_t^i = TS_t^{MN}$ is valid, the verification passes, and fails otherwise. To make verification more efficient, MN_j can perform batch verification.

$$e(\delta_i, g_2) = e(H_2(\mathbb{C}_i[t]||TS_t^i||DW_i||PK_i), PK_i) \quad (10)$$

- **Step 2:** MN_j performs hash operation to generate Merekle tree root hash value. Then MN_j generates a new block $Block = (Data||H(Data||Timestamp)||Timestamp)$, and broadcasts the block to other SMs in the residential area RA_j.
- **Step 3:** After receiving the block, SMs verify the block's hash value, timestamp and its data, then send the result of the verification to other SMs to achieve mutual supervision among SMs.
- **Step 4:** SMs send their own check results to MN_j. MN_j collects feedback from all SMs and checks them. If all SMs agree on the legitimacy and integrity of the block, MN_j adds the block to the blockchain in chronological order and sends the block to other SMs. If there exists SM disagrees with the check result, MN_j checks the feedback information and sends the block to this SM again for a second check.
- **Step 5:** MN_j aggregates the encrypted data of all SMs and decrypts it to get the total power consumption of the area at the current time by Eq. (11).

$$\begin{aligned} &\sum_{i\in\mathcal{U}} \mathbb{C}_i[t] - DA^\top TS_t \\ &= \sum_{i\in\mathcal{U}} ((s_i^\top \cdot TS_t) + r_i[t]g) - (\sum_{i\in\mathcal{U}} s_i)^\top TS_t \\ &= \sum_{i\in\mathcal{U}} r_i[t]g \in G \end{aligned} \quad (11)$$

Since $\sum_{i\in\mathcal{U}} r_i[t]g$ is not a very large value, there are many ways to calculate the aggregated value, such as Shank's baby-step giant-step algorithm [28].

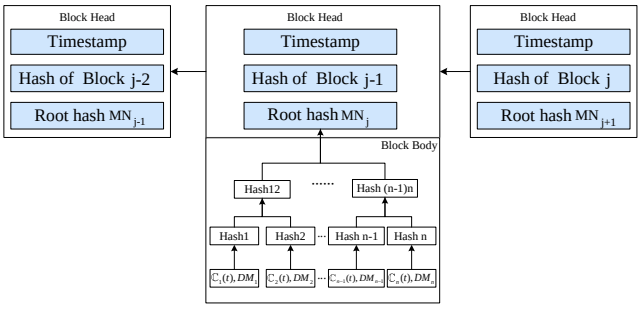

Fig. 4. The blockchain structure of the proposed scheme.

E. Judgement Phase

To achieve efficient detection, our solution will perform electricity theft detection after discriminating whether there is electricity theft in residential areas.

For each residential area, the transformer meter measures the total amount of electricity supplied to that residential area during the electricity theft detection period, $E_{DTM}(t)$. Meanwhile, the MN aggregates the readings uploaded by all SMs in the residential area, $\sum_i E_{SM_i}(t)$, the SO determines whether there is electricity theft by Eq. (12):

$$E_{DTM}(t) > \sum\nolimits_i E_{SM_i}(t) + E_{TL}(t) + \varepsilon \quad (12)$$

where $E_{TL}(t)$ is the technical loss (TL) in transmission lines within the residential area and ε is the calculation error for TL. The SO can use historical data to analyze the technical loss, while many methods exist [7] to calculate the technical loss. If the Eq. (12) is valid, SO considers that there is electricity theft in the current area. Afterwards, SO reads the power consumption data uploaded by each SM_i in the blockchain for electricity theft detection.

F. Electricity Theft Detection Phase

In this sub-section, a privacy preserving electricity theft detection model is presented in the proposed scheme, and then we explain the experimental settings, including computing platforms, dataset and data pre-processing.

*1) **Privacy-preserving Electricity Theft Detection Model**:* As shown in Fig. 5, our model is composed of the fully connected layer and the long short-term memory network. The core operation of the fully connected layer is the multiplication of a matrix and a vector, which can be expressed as $\boldsymbol{xW}$. More detailed representations are:

$$\begin{aligned} &[x_1, x_2, \cdots x_d] \times \begin{bmatrix} w_1[1] & w_2[1] & \cdots & w_c[1] \\ w_1[2] & w_2[2] & \cdots & w_c[2] \\ \vdots & \vdots & \cdots & \vdots \\ w_1[d] & w_2[d] & \cdots & w_c[d] \end{bmatrix} \\ &= [\boldsymbol{x}\cdot\boldsymbol{w_1}^\top, \boldsymbol{x}\cdot\boldsymbol{w_2}^\top, \cdots, \boldsymbol{x}\cdot\boldsymbol{w_c}^\top] \end{aligned} \quad (13)$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{W}$ is the weight matrix, and then $\boldsymbol{b}$ is the bias vector is added. This operation can be seen as an inner product of the input vector $\boldsymbol{x}$ and each column of the weight matrix $\boldsymbol{W}$. It can also be viewed as a group of n

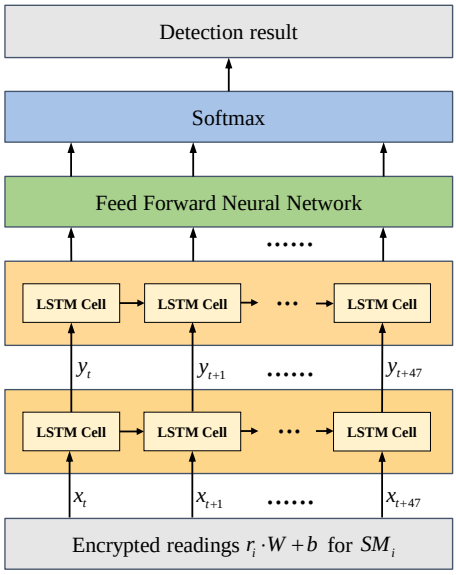

Fig. 5. The LSTM based privacy-preserving electricity theft detection framework.

d-equations, where the input vector $\boldsymbol{x}$ are the unknowns and the weight matrix $\boldsymbol{W}$ are the coefficients, and since n is less than d, the input vector $\boldsymbol{x}$ cannot be solved.

Therefore, in order to perform electricity theft detection in the ciphertext state of the consumers' power consumption data, the result of the inner product of consumers' power consumption data $r_i = [r_i[1], r_i[2], \cdots, r_i[d]]$ and each column of the weight matrix is obtained by Eq. (14).

$$\begin{aligned}&\sum_{t=1}^{d} w_c[t] \times \mathbb{C}_i[t] - DW_{ci} \\ &= \sum_{t=1}^{d} w_c[t]((s_i^\top \cdot TS_t) + r_i[t]g) - \sum_{t=1}^{d} w_c[t](s_i^\top \cdot TS_t) \\ &= (\sum_{t=1}^{d} r_i[t]w_c[t])g\end{aligned} \quad (14)$$

The output of the fully connected layer is obtained by calculating the inner product of each column of the weight matrix with the consumer power consumption data, and then adding the bias vector $\boldsymbol{b}$ as follows:

$$[(\boldsymbol{r_i} \cdot \boldsymbol{w_1^T}) + \boldsymbol{b_1}, (\boldsymbol{r_i} \cdot \boldsymbol{w_2^T}) + \boldsymbol{b_2}, \cdots, (\boldsymbol{r_i} \cdot \boldsymbol{w_c^T}) + \boldsymbol{b_c}]$$

After SO gets the output result of the fully connected layer, it still cannot solve the original consumer power consumption data, and the consumers' power consumption data is input to the next layer of the network in the encrypted state, finally, detection result is inferred after layer-by-layer computation.

The detection model uses the categorical cross-entropy as the loss function. In the model training phase, we use the RMSprop optimizer to train the model for 30 epochs, 512 batch sizes and 0.001 learning rate. To prevent overfitting, we use the kernel $\ell 2-$regularizer in the LSTM layer, and at the same time the callback function ReduceLROnPlateau in the keras framework [29] is used to dynamically reduce the learning rate, and the callback function EarlyStopping is used to obtain the optimal model. Parameters of our model structure are summarized in Table III, where AF stands for activation function.

TABLE III
PARAMETERS OF MODEL STRUCTURE.

Layer(type)	No. of neurons	No. of parameters	AF
dense(Dense)	10	20	tanh
lstm(LSTM)	300	373200	tanh,sigmoid
lstm-1(LSTM)	300	721200	tanh,sigmoid
dense-1(Dense)	2	602	softmax

*2) **Computing Platforms**:* In our experiments, we build a Tensorflow virtual environment on a server with unbutu 18.04.6 LTS system and NVIDIA Tesla T4 GPU as well as use the Keras framework to train and evaluate the model.

*3) **Dataset**:* We use the dataset from the Irish Smart Energy Trials [30], which contains the power consumption data of more than 1000 consumers in 535 days from 2009 to 2010, and fine-grained power consumption data is reported by each SM every 30 minutes.

*4) **Data Pre-processing**:* We select the smart meter data of 200 consumers from the dataset and create one record of the consumer's power consumption data (48 readings) for one day, with a total of 107,200 records. Since all the data in the dataset are from honest consumers' data, so we use the electricity theft attack proposed by [7] to generate malicious consumers' data. We based on the dataset of benign samples, for each sample $X = \{x_t | 1 \leq t \leq 48\}$, perform the following operations to generate six malicious types of data:

1) $f_1(x_t) = \alpha x_t$, $\alpha = random(0.1, 0.8)$;
2) $f_2(x_t) = \beta_t x_t$, $\beta_t = random(0.1, 0.8)$;
3) $f_3(x_t) = mean(X)$;
4) $f_4(x_t) = \beta_t mean(X)$, $\beta_t = random(0.1, 0.8)$;
5) $f_5(x_t) = x_{48-t}$;
6) $f_6(x_t) = \gamma_t x_t$.

$$\gamma_t = \begin{cases} 0 & ts < t < te \quad ts = random(0, 42) \\ 1 & else \quad te - ts = random(6, 48) \end{cases}$$

Through electricity theft attacks generate 643,200 records of malicious data. Since the data of honest data records are only 107200, which leads to the problem of unbalanced sample categories of data. Therefore, we apply for each record the adaptive synthetic sampling method (ADASYN) [31] to balance the size of honest and malicious classes. We randomly divide the balanced dataset into a training dataset (80%) and a testing dataset(20%) to perform the training of the model.

VI. PERFORMANCE EVALUATION

In this section, at first, our method is compared with other methods that deal with time series to demonstrate the better performance of our method. Then, we study the parameters of our model. Finally, we evaluate the performance of our electricity theft detection model in our test set. Meanwhile, we compare the computation and communication of the model with other schemes.

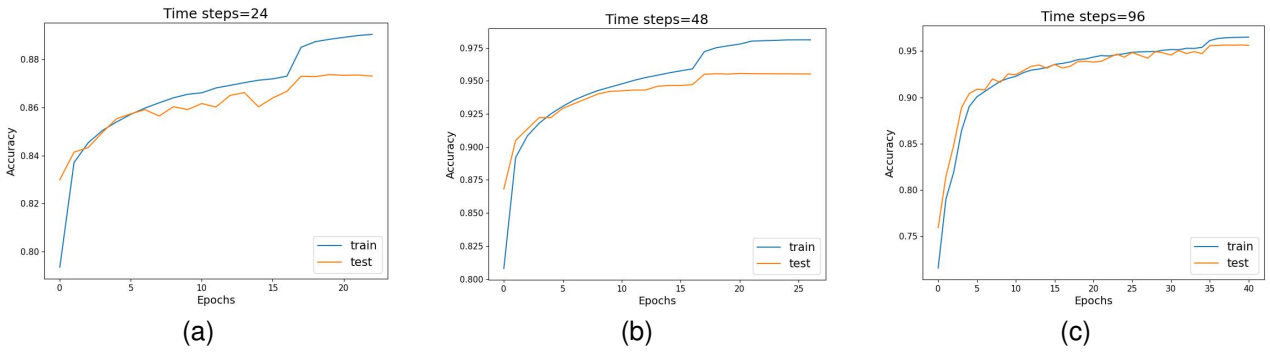

Fig. 6. Parameter study of time steps t.

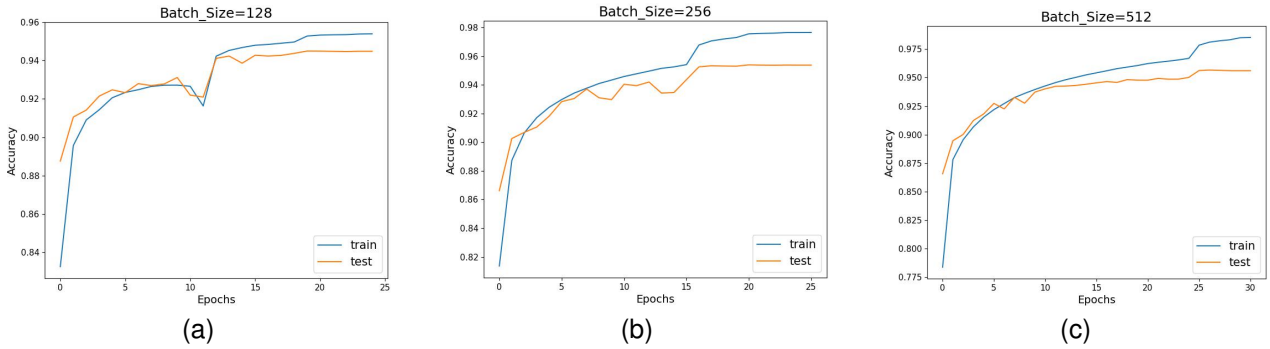

Fig. 7. Parameter study of batch size β.

A. Method Comparison

To demonstrate the better performance of our model, the experimental comparison with other methods was performed on a test dataset. Concretely, Deng et al. proposed a tree-ensemble method, referred to as time series forest (TSF), for time series classification [32]. Middlehurst et al. proposed an improved hierarchical vote collective of transformation-based ensembles (HIVE-COTE) for time series classification [33]. Dempster et al. proposed a simple linear classifier based on the random convolution kernels (ROCKET) [34]. Meanwhile, in [15], the authors proposed to use one-dimensional convolutional neural network (CNN) for electricity theft detection. Table IV gives the experimental results for each method using the same training data set and testing data set, and we see that the LSTM model gets the highest accuracy score of 95.56%.

TABLE IV
ALGORITHM ACCURACY SCORES.

Algorithm	Accuracy Score(%)
The LSTM model	**95.56**
1-D CNN model [15]	93.20
Time Series Forest [32]	86.36
The improved HIVE-COTE [33]	90.91
ROCKET [34]	78.76

B. Parameter Study

Various hyper-parameters of the model have an impact on the performance of the model. For our model, what is more important is the time step, which is the number of power readings input in the model. In our model, the time step is the same as the theft detection period. For the theft detection model, increasing the detection time period means that the communication overhead of the model will increase, so a reasonable theft detection period must be determined. Therefore, we deeply analyze the impact of these parameters on the performance of our model.

*1) **Effect of time steps** t:* Fig. 6 shows the accuracy of the validation set with varying epochs when the time steps are different. We can find out that different time steps affect the accuracy of the model as well as the training time, while the longer the time steps, the longer the theft detection period will be, which will lead to a rise in the overall model in terms of communication overhead. Although the difference in accuracy between time steps 96 and 48 is not significant, the training time is shorter and communication is less when the time steps are 48.

*2) **Effect of learning rate** ℓ:* In the model training progress, we use the RMSprop optimizer with a default learning rate $\ell = 0.001$. To find the optimal model, we use the callback function ReduceLROnPlateau in the Keras framework, which serves to reduce the learning rate when learning stagnates. As shown in Fig. 6, there is some improvement in accuracy after reducing the learning rate.

*3) **Effect of batch size** β:* Fig. 7 shows the performance of our model with setting the batch size as 512 which gets the highest accuracy with 95.60 $\%$ while needs more epochs to optimize. The experimental results show that a smaller batch

TABLE V
THE PERFORMANCE OF OUR MODEL AND OTHER SCHEMES.

Model	DR(%)	FA(%)	HD(%)	Accuracy(%)	Model detection overhead	Communication overhead
Our model	93.72	**2.62**	**91.10**	**95.56**	**56.03 ms**	600 Bytes
ETDFE [12]	92.56	5.84	86.72	93.36	1.94 seconds	40 Bytes
PPETD MD1 [15]	91.50	7.40	84.10	91.80	48 minutes	1900 MB
PPETD MD2 [15]	90.00	8.79	81.2	90.20	39 minutes	1675 MB
PPETD MD3 [15]	88.60	3.90	84.60	90.30	35 minutes	1375 MB
Jokar et al [7]	**94.00**	11.0	83.0	-	-	-

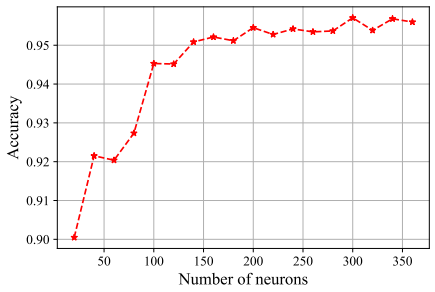

Fig. 8. Parameter study of neurons η.

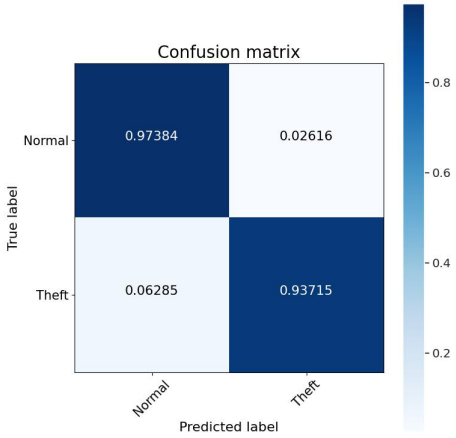

Fig. 9. Confusion matrix of our model.

size can speed up the optimization within same epochs, which suggests that setting the bath size between 256 and 512 is more acceptable.

*4) **Effect of neurons** η:* Fig. 8 shows that the highest accuracy is achieved when the number of neurons in the LSTM layer is 300-360. A more number of model neurons represents a slower model inference, so the neurons of our model are set to 300.

C. Performance of electricity theft detection model

*1) **Performance Metrics**:* In order to evaluate our electricity theft detection model, we conduct the experiments by considering four performance metrics: accuracy, the detection rate (DR), and the false acceptance rate (FA) as well as the highest difference (HD). Accuracy measures the percentage of correct classifications in the testing dataset. The detection rate measures the percentage of detected malicious consumers in the total malicious consumers. The false acceptance rate measures the percentage of honest consumers who are mistakenly detected as malicious consumers. When DR, accuracy, and HD are high and FA is low, the model performance is better.

$$DR = \frac{TP}{TP+FN}, FA = \frac{FP}{TN+FP}, HD = DR - FA,$$

where TP, FP and TN stands for true positive, false positive and true negative, respectively.

$$accuracy = \frac{1}{s}\sum_{i=1}^{s}\phi(f(x_i), y_i),$$

$$\phi(f(x_i), y_i) = \begin{cases} 1 & f(x_i) = y_i \\ 0 & f(x_i) \neq y_i \end{cases},$$

where s is the total number of samples in the testing dataset, y_i is the label for the $i-$th consumer, $f(x_i)$ is is the inference result of the model.

*2) **Performance Comparison**:* We obtain the confusion matrix of our model by using the Scikit-learn python library. As shown in Fig. 9, in the confusion matrix of our model, the proportion of consumers who are predicted to be electricity theft consumers among those who are really electricity theft consumers is the DR, the proportion of consumers predicted to be electricity theft as a percentage of those who are truly normal consumers is the FA.

Table V shows the evaluation results of our model and the existing models with privacy preservation. The proposed scheme is better in terms of FA, accuracy, and HD among the schemes considering privacy protection. Our privacy detection model has higher accuracy and HD, 95.56%, and 91.10%, respectively. At the same time, the FA in our model is 2.62%, which is lower than other schemes. From the evaluation results, we can demonstrate that the proposed scheme has a better performance. Moreover, the performance of our model is not decreased by the use of encryption compared to [15] due to the fact that we use the inner product operation of the parameters of the first layer of the model with the consumers' power consumption data, which has the same output as the plaintext direct input to the first layer of the model.

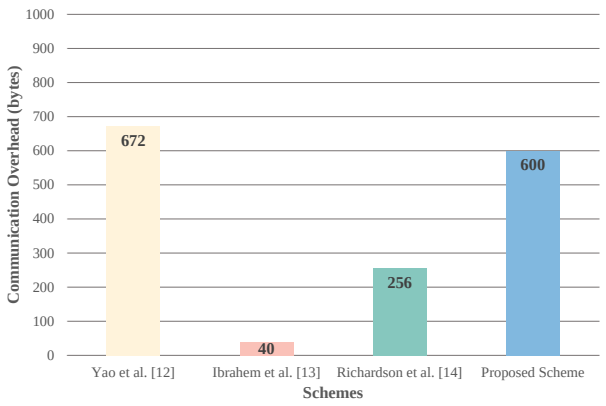

Fig. 10. Comparison of the communication overhead with other schemes.

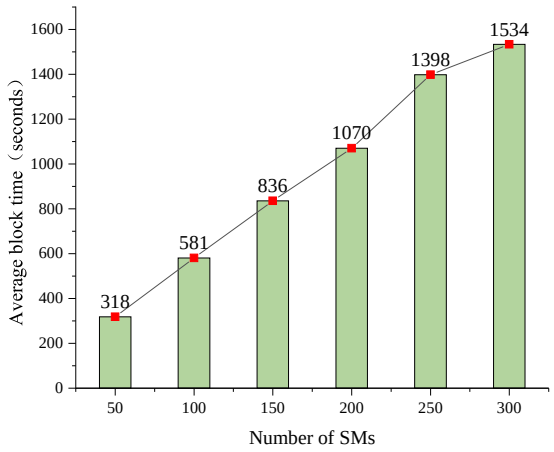

Fig. 11. Average block time for the number of SMs from 50 to 300.

D. Computation and Communication Overhead

To evaluate this in a more realistic environment, we used the Python "Charm" crypto-graphic library [35] on a Raspberry Pi Zero W device with a 1.0 GHz single-core CPU and 512 MB of RAM. The elliptic curve of size 160 bits (MNT159 curve) was also used.

*1) **Communication overhead**:* In our model, the main communication overhead comes from the SMs transferring $\mathbb{C}_i||TS_t^i||DW_i||\delta_i||PK_i$ to the MN. We use an elliptic curve with 160-bit security level. From Eq. (5) to Eq. (9), it can see that the ciphertext, signature, and public key PK size are all 40 bytes, the $DW_i = \{DW_{1i}, DW_{2i}, \cdots, DW_{10i}\}$ size is 400 bytes, and the timestamp size is 80 bytes, so it takes 600 bytes for the SM_i to report one reading. PPETD [15] uses secure multiplication, $sigmoid(\cdot)$ security evaluation, and garbled circuits to protect the privacy evaluation of the CNN model, which results in a high communication overhead of about 1900 MB per SM. Yao et al.'s scheme [11] requires sending a ciphertext, signature, and timestamp to two institutions to complete the aggregation and detection, and we assume that it generates 2048 bits of ciphertext, 40 bytes of signature, and 40 bytes of timestamp, the total size required is 672 bytes. Richardson et al. [14] and Ibrahem et al.'s schemes [12] only sends 40 bytes, and 256 bytes of ciphertext, respectively. Meanwhile, Fig. 10 gives a comparison of the communication overhead with other schemes. It can be seen that the proposed scheme achieves more security within an acceptable range of communication overhead.

*2) **Computation overhead**:* In the proposed scheme, the computations mainly include three phases: reporting phase, aggregating phase, and electricity theft detection phase. In the reporting phase, the main computation overhead comes from the encryption, signature, decryption keys generation and timestamp generation operations of the SM, therefore, the total time cost of the reporting phase is 59.488 ms. In the aggregation phase, MN achieves aggregating readings, decrypting, and verifying signatures, the total time cost is 129.635 ms. In the detection phase, the computation cost of decrypting to obtain $\boldsymbol{rW}$ is 49.63 ms. The computation costs of required functions are listed in Table VI. Experimental results show that this is feasible in a real-world environment.

In model inference speed, the total evaluation time of ETDFE for a 15-layer FFN model with 3,391,634 parameters is about 1.94 seconds and PPETD MD1 takes 48 minutes to evaluate the model, our model has only 1,095,022 parameters and its evaluation time is only 56.03 ms. In addition, the proposed scheme is more efficient compared to other schemes because it performs electricity theft detection after identifying the suspected theft area.

TABLE VI
AVERAGE COMPUTATION COST OF BASIC FUNCTIONS.

Notations	Description	Time(ms)
T_C	Time cost of encryption	0.096
T_{agg}	Time cost of aggregating 200 readings	2.21
T_{decAgg}	Time cost of decrypting aggregated readings	0.135
T_{DM}	Time cost of public key generation DW	45.36
T_{decDW}	Time cost of decrypting to obtain $\boldsymbol{rW}$	49.63
TS_t	Time cost of generating timestamp	0.852
T_{sig}	Time cost of signature operation	13.18
T_{versig}	Time cost of the verify signature operation	127.29
T_m	Time cost of model detection	56.03

E. Blockchain simulations

Block time is a measure of the time it takes for the miners or validators in the network to verify the transactions within a block and generate a new block in that blockchain. Very short block times may lead to abnormal behavior, because nodes may not have enough time to send transactions, and synchronize their transaction pool or blockchain. Very long block time wastes arithmetic power and reduces the security of the system. Therefore, an appropriate block time is important. As shown in Fig. 11, average block time is simulated in the blockchain simulation system [36] for the number of SMs in the detection region from 50 to 300. The block time should be as much as possible less than the period of the SM reporting power consumption readings, and the SO can select the number of SMs in the area based on the reporting period.

VII. System Analysis

In this section, we aim to demonstrate that the proposed scheme can achieve the following security and privacy guarantee, while it can resist the attacks in Section IV-B. In addition, in order to prove that the proposed scheme is more secure than the existing schemes, we perform a comparison of system characteristics.

A. Security analysis

Scenario 1: The privacy of consumers' power consumption cannot be inferred by any attacker.

Proof: The consumers' fine-grained data $r_i[t]$ is encrypted and sent to the MN. The confidentiality of $r_i[t]$ is achieved by an elliptic curve over finite fields. Specifically, in order to analyze the consumer's private information, the attacker needs to crack the consumer's continuous long-term encrypted data $[\cdots, \mathbb{C}_i[t-1], \mathbb{C}_i[t], \mathbb{C}_i[t+1], \cdots]$, but the attacker can only get the public parameters, which is infeasible in cracking the computation. In the electricity theft detection stage, the input encrypted power consumption data $\mathbb{C}_i[t], t = [1, 2, \cdots, d]$ is decrypted to get the output result $[(\boldsymbol{r_i} \cdot \boldsymbol{w_1^T}) + \boldsymbol{b_1}, (\boldsymbol{r_i} \cdot \boldsymbol{w_2^T}) + \boldsymbol{b_2}, \cdots, (\boldsymbol{r_i} \cdot \boldsymbol{w_n^T}) + \boldsymbol{b_n}]$ of the first layer of neural network. Since n is less than d, n d-element equations cannot be solved, therefore the SO cannot obtain the original consumers' power consumption data $[r_i[1], r_i[2], \cdots, r_i[d]]$, while still complete the electricity theft detection. Therefore, the proposed scheme preserves the privacy of consumers.

Scenario 2: Consumers' fine-grained power consumption data cannot be falsified and forged during transmission and storage, etc.

Proof: The messages $\mathbb{C}_i[t]||TS_t^i||DW_i||PK_i$ sent by each SM_i in the scheme are BLS signed as $\delta_i = x_i \cdot H_2(\mathbb{C}_i[t]||TS_t^i||DW_i||PK_i)$ to ensure the integrity of the data and prevent falsification. After accepting the message, the MN creates a block after establishing the Merkle tree, and each SM_i can access the block to verify whether its data has been falsified. Meanwhile, since all data transfers in the blockchain have timestamps and cannot be changed when added to the blockchain, so the proposed scheme can resist data falsification and forgery.

Scenario 3: The proposed scheme does not require a third party and also can resist collusion attacks by smart grid entities.

Proof: In the proposed scheme, the whole process does not require the participation of a third party, which makes the scheme more reliable and convenient. In the keys security aggregation process, each SM_i negotiates the masks $sk_{i,o} = x_i s_o^{pk} (i \in \mathcal{U}; o \in \mathcal{U}, o \neq i)$ with all other SMs, and the mask agreement is based on the computational Diffie-Hellman hard problem. Suppose the SO wants to get the private key s_i of the SM_i after colluding with the MN, it still needs to collude with $m-2$ SMs, which is not achievable in practice. Therefore, our scheme resists collusion attacks.

B. Effective defense evaluation

In this sub-section, we calculate the probability of successful attacks by the attacker in two scenarios and illustrate the effectiveness of the scheme through mathematical proofs.

1) Scenario 1: Network attackers may destroy data before it is transmitted, during its transmission, and after it is received by the MN to render the system inoperable.

2) Scenario 2: Network attackers may tamper with the original data before it is transmitted, during data transmission, and after it is received by the MN (before it is broadcast) to allow false data to be verified.

The attack methods and success probabilities of data being destroyed and tampered with before, during, or after transmission are summarized in Table VII.

TABLE VII
Probability of successful attacks.

Stages		Scenario 1	Scenario 2
Pre-data transmission	Attack method	Hack into m SMs	Hack into m SMs; Get the keys
	Probability	$\prod_{i=1}^{m} P_{SM_i}$	$\prod_{i=1}^{m} P_{SM_i} \cdot \prod_{i=1}^{m} P_{k_i}$
Data in transit	Attack method	Hack m channels	Hack m channels; Get the keys
	Probability	$\prod_{i=1}^{m} P_{C_i}$	$\prod_{i=1}^{m} P_{C_i} \cdot \prod_{i=1}^{m} P_{k_i}$
Data received	Attack method	Hack into MN	Hack into m SMs; Get the keys
	Probability	P_{MN}	$\prod_{i=1}^{m} P_{SM_i} \cdot \prod_{i=1}^{m} P_{k_i}$

For scenario 1, we suppose that the probability of an attacker hacking into a smart meter is denoted as P_{SM}, $0 < P_{SM} < 1$, and the probability of an attacker hacking into a channel is denoted as P_C, $0 < P_C < 1$. In order to make the system unworkable, the attacker needs to attack m smart meters with success probability $\prod_{i=1}^{m} P_{SM_i}$ before data transmission, m channels with success probability $\prod_{i=1}^{m} P_{C_i}$ during data transmission, and after the MN accepts the data, the success probability of the attack is P_{MN}. However, because m is large, the attacker's probability of hacking into the smart meters is extremely low, and even if it is destroyed during the data transmission phase, it can still be detected from the data signature to discover and eliminate this attack, and meanwhile, when the MN is attacked and the data is destroyed, all other SMs will find the wrong data in the consensus phase and re-vote to select a new MN, so our scheme has good defense capability under scenario 1.

For scenario 2, we suppose that the probability of an attacker stealing the private key of the smart meter is denoted as P_k, $0 < P_k < 1$. When the attacker wants to tamper with the data in the smart grid, the probability of successful attack before data transmission is $\prod_{i=1}^{m} P_{SM_i} \cdot \prod_{i=1}^{m} P_{k_i}$, during data transmission is $\prod_{i=1}^{m} P_{C_i} \cdot \prod_{i=1}^{m} P_{k_i}$, after the MN receives the data, the probability of successful attack is $\prod_{i=1}^{m} P_{SM_i} \cdot \prod_{i=1}^{m} P_{k_i}$. Compared with scenario 1, scenario 2 can be attacked with more demanding requirement conditions and lower probability of successful attack. From the above probabilistic analysis, it can be demonstrated that our scheme can perform the basic tasks in a more secure environment.

C. System characteristic comparison

The proposed scheme is compared with several other representative privacy-preserving electricity theft detection schemes for smart grid in terms of non-reliance on any trusted third party (TTP), data non-falsifiability (DNF), data non-repudiation (DNR), and data non-tamperability (DNT). As shown in Table VIII, the related work does not achieve all the desired characteristics of the smart grid, while only the proposed scheme achieves it.

TABLE VIII
SYSTEM CHARACTERISTICS COMPARISON.

	No TPP	DNF	DNR	DNT
Joker et al. [7]	Yes	No	No	No
Yao et al. [11]	No	Yes	Yes	No
I.Ibrahem et al. [12]	No	No	No	No
Richardson et al. [14]	No	No	No	No
Our Scheme	Yes	Yes	Yes	Yes

VIII. CONCLUSION

In this paper, we propose a more secure blockchain-based privacy-preserving electricity theft detection scheme. The proposed scheme does not require a third party, which avoids the security and privacy issues brought about by a third party. Meanwhile, the blockchain's distributed storage of electricity theft detection scheme is used to solve the problems such as data tampering due to centralized storage data resulting in the inability to perform electricity theft detection. In addition, a real dataset and environment are used for simulation evaluation. The experimental results show that the proposed scheme can detect malicious consumers more accurately with acceptable communication and computational overhead. System analysis shows that the proposed scheme is more secure compared to existing schemes. For our future work, we intend to improve the proposed scheme by reducing communication and computation overhead.

REFERENCES

[1] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "A survey on smart grid potential applications and communication requirements," *IEEE Transactions on industrial informatics*, vol. 9, no. 1, pp. 28–42, 2012.

[2] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2018.

[3] Z. Zeng, X. Wang, Y. Liu, and L. Chang, "Msda: multi-subset data aggregation scheme without trusted third party," *Frontiers of Computer Science*, vol. 16, no. 1, pp. 1–7, 2022.

[4] X. Xia, Y. Xiao, and W. Liang, "Sai: A suspicion assessment-based inspection algorithm to detect malicious users in smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 361–374, 2019.

[5] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE security & privacy*, vol. 7, no. 3, pp. 75–77, 2009.

[6] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3953–3962, 2018.

[7] P. Jokar, N. Arianpoo, and V. C. Leung, "Electricity theft detection in ami using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, 2015.

[8] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2017.

[9] J. E. Rubio, C. Alcaraz, and J. Lopez, "Recommender system for privacy-preserving solutions in smart metering," *Pervasive and Mobile Computing*, vol. 41, pp. 205–218, 2017.

[10] R. Hoenkamp, G. B. Huitema, and A. J. de Moor-van Vugt, "The neglected consumer: The case of the smart meter rollout in the netherlands," *Renewable Energy Law and Policy Review*, pp. 269–282, 2011.

[11] D. Yao, M. Wen, X. Liang, Z. Fu, K. Zhang, and B. Yang, "Energy theft detection with energy privacy preservation in the smart grid," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7659–7669, 2019.

[12] M. I. Ibrahem, M. Nabil, M. M. Fouda, M. M. Mahmoud, W. Alasmary, and F. Alsolami, "Efficient privacy-preserving electricity theft detection with dynamic billing and load monitoring for ami networks," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1243–1258, 2020.

[13] M. Wen, R. Xie, K. Lu, L. Wang, and K. Zhang, "Feddetect: A novel privacy-preserving federated learning framework for energy theft detection in smart grid," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6069–6080, 2021.

[14] C. Richardson, N. Race, and P. Smith, "A privacy preserving approach to energy theft detection in smart grids," in *2016 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2016, pp. 1–4.

[15] M. Nabil, M. Ismail, M. M. Mahmoud, W. Alasmary, and E. Serpedin, "Ppetd: Privacy-preserving electricity theft detection scheme with load monitoring and billing for ami networks," *IEEE Access*, vol. 7, pp. 96 334–96 348, 2019.

[16] H. Fu, P. Hu, Z. Zheng, A. K. Das, P. H. Pathak, T. Gu, S. Zhu, and P. Mohapatra, "Towards automatic detection of nonfunctional sensitive transmissions in mobile applications," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 3066–3080, 2020.

[17] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3162–3173, 2018.

[18] J. Song, T. Gu, Z. Fang, X. Feng, Y. Ge, H. Fu, P. Hu, and P. Mohapatra, "Blockchain meets covid-19: a framework for contact information sharing and risk notification system," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2021, pp. 269–277.

[19] M. R. Hamouda, M. E. Nassar, and M. Salama, "A novel energy trading framework using adapted blockchain technology," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2165–2175, 2020.

[20] H. Fan, Y. Liu, and Z. Zeng, "Decentralized privacy-preserving data aggregation scheme for smart grid based on blockchain," *Sensors*, vol. 20, no. 18, p. 5282, 2020.

[21] M. R. Hamouda, M. E. Nassar, and M. M. Salama, "Centralized blockchain-based energy trading platform for interconnected micro-grids," *IEEE Access*, vol. 9, pp. 95 539–95 550, 2021.

[22] X. Zhang, L. You, and G. Hu, "An efficient and robust multidimensional data aggregation scheme for smart grid based on blockchain," *IEEE Transactions on Network and Service Management*, 2022.

[23] S. Chen, Z. Shen, L. Zhang, Z. Yan, C. Li, N. Zhang, and J. Wu, "A trusted energy trading framework by marrying blockchain and optimization," *Advances in Applied Energy*, vol. 2, p. 100029, 2021.

[24] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[25] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Journal of cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[26] H. Moniz, N. F. Neves, and M. Correia, "Byzantine fault-tolerant consensus in wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2441–2454, 2012.

[27] Y. Zou, D. Yu, P. Hu, J. Yu, X. Cheng, and P. Mohapatra, "Jamming-resilient message dissemination in wireless networks," *IEEE Transactions on Mobile Computing*, 2021.

[28] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1997, pp. 256–266.

[29] C. Keras, "Theano-based deep learning librarycode: https://github. com/fchollet," *Documentation: http://keras. io*, 2015.

[30] "Irish social science data archive," http://www.ucd.ie/issda/data/commissionforenergyregulationcer/.

[31] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.
[32] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A time series forest for classification and feature extraction," *Information Sciences*, vol. 239, pp. 142–153, 2013.
[33] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "Hive-cote 2.0: a new meta ensemble for time series classification," *Machine Learning*, vol. 110, no. 11, pp. 3211–3243, 2021.
[34] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.
[35] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.
[36] L. Stoykov, K. Zhang, and H.-A. Jacobsen, "Vibes: fast blockchain simulations for large-scale peer-to-peer networks," in *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos*, 2017, pp. 19–20.