An Authorizable Threshold Signature Scheme for Secure Sharing and Mining of Power IoT Data

Wei Chen, XiaoTian Xu, Cong Hou, RanXin Gao, GuanLin Si, and Yue Zhang
North China Electric Power Research Institute Co., Ltd
Beijing, China
wchen3@gmail.com, 123792338@qq.com, 591119439@qq.com,
1093187022@qq.com, m18811612766@163.com, 741473522@qq.com

***Abstract*—The rapid development and wide popularity of electric power IoT has brought convenience to people's daily lives. Comprehensive informatization and intelligent transformation of the electric power system are carried out to support the interconnection and interoperability among devices, as well as to facilitate the effective collection, transmission, processing, and analysis of data. Through electric power IoT, real-time monitoring and analysis of power equipment status, operation and load situation can be realized to improve the reliability, safety and economy of the power system. The Internet of Things (IoT) technology has enabled seamless integration between the physical and digital worlds by connecting and interconnecting devices. It provides people with smarter and more convenient ways of living. On the other hand, social computing technology focuses on utilizing social network and social media data for computation and analysis, revealing relationships and behavioral patterns among individuals. Data mining technology, in turn, aims to discover patterns and trends from large-scale data. The fusion of these three technologies promises exciting innovations and transformative changes that will have a profound impact on our society.However, Cyberspace encompasses pervasive security vulnerabilities across the entirety. The transmission process of power data entails the potential for corruption or unauthorized tampering to occur, thus compromising the integrity. To solve this problem, we propose an authorizable threshold signature scheme for the power IoT. In our scheme, the perception nodes in the power IoT authorize the storage nodes with signatures in time periods. Only those storage nodes possessing proper authorization are capable of generating valid signatures. As such, our scheme effectively fulfills the requirements of non-forgeability, non-repudiation, and forward security.**

***Index Terms*—Power IoT, threshold signature, signature authorization, data integrity.**

I. Introduction

THE integration of the Internet of Things (IoT) in smart grids, particularly in the electricity sector, represents a culmination of advancements in information and communication technology. This integration has reached a stage of development and maturity, fostering an effective amalgamation of communication infrastructure resources with power system infrastructure resources. As a result, it significantly elevates the information technology level within the power system, enhancing the efficiency of existing infrastructure and providing vital technical support across power generation, transmission, transformation, distribution, and consumption within the power grid.

Within the Power Internet of Things, the process begins with smart meters collecting power consumption data, which is then transmitted to a concentrator located in a building or park. This concentrator undertakes the task of packaging and uploading the power consumption data to the power consumption data acquisition master station [1]. The voluminous data collected from various electricity-consuming devices serves as a valuable resource, offering insights into user behavior, preferences, and habits. Leveraging advanced data mining techniques, these insights pave the way for the discovery of user needs and behavior patterns. This, in turn, facilitates the delivery of personalized recommendations and services to users. Furthermore, the integration of social computing technologies enables a deeper understanding of residents' electricity consumption habits and lifestyles, providing essential inputs for strategic planning within the distribution IoT and optimizing services for end-users.

Despite these advancements, the communication channels in the Power Internet of Things, particularly those involving smart meters, concentrators, and the power data acquisition master station, pose inherent security risks. The use of public or wireless network communication exposes these channels to potential eavesdropping during data transmission [2]. To address these secu-

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

rity concerns, existing communication schemes employ multi-layer encryption techniques throughout the transmission process, effectively mitigating risks such as data compromise, unauthorized manipulation of electricity data, and illegitimate access to the power IoT through the utilization of counterfeit identities by potential adversaries.

However, the rapid evolution of big data technology introduces new challenges. Attackers may exploit potential vulnerabilities by conducting comparative analyses on a large dataset, potentially reversing cracks in existing encryption systems. This introduces the risk of user data tampering, compromising the integrity of user data and, consequently, threatening user privacy and security [3]. Recognizing this emerging threat, there is an urgent need for a robust data security mechanism to prevent user data tampering during the transmission process, ensuring the integrity of the data.

In response to the problem of insufficient data integrity security protection capabilities in communication links in untrusted environments of the power internet of things, this paper proposes a secure transmission scheme for electricity data based on threshold signature technology. This scheme utilizes threshold signature technology to achieve integrity verification of user data, ensuring the privacy and security of user electricity data while significantly reducing verification costs.

II. Related work

Digital signature technology is developed from public key cryptography. It plays an important role in identity authentication, data integrity, nonrepudiation, anonymity and other security aspects. Today, it has become an important security guarantee in digital society. Below is a brief introduction to several classic traditional digital signature algorithms. In 1978, Ronald Rivest, Adi Shamir, and Len Adleman first proposed the RSA algorithm, which is currently one of the classic algorithms in computer cryptography and the most widely used digital signature algorithm [4]. It has played a significant role in the fields of information security and authentication. It is worth noting that the key implementation of the RSA digital signature algorithm is the same as that of the RSA encryption algorithm; hence, it is collectively referred to as the RSA algorithm. The security of the algorithm depends on the difficult problem of large number decomposition in number theory, that is, the multiplication of two large prime numbers makes it very easy to obtain a large integer, but it is very difficult to decompose a large integer factorization into two large prime numbers. In 1985, Tather ElGamal of Stanford University proposed the ElGamal digital signature algorithm using the EIGamal public key cryptosystem, which is one of the classic digital signature algorithms. Its security depends on the difficulty of computing discrete logarithms over finite fields. At present, many digital signature algorithms are based on the expansion or improvement of this algorithm, which has high practicality [5]. In 1989, C. Schnorr proposed the Schnorr digital signature algorithm based on the ElGamal digital signature algorithm, and its security was also based on the difficulty of computing discrete logarithms over finite fields [6]. In 1991, the National Institute of Standards and Technology (NIST) of the United States proposed the digital signature algorithm (DSA), which is a variant of the ElGamal algorithm, and its security also depends on the difficulty of solving discrete logarithms. In May 1994, NIST's digital signature standard DSS [7] adopted the digital signature algorithm DSA. In 1992, Scott and Vanstone first proposed the elliptic curve digital signature algorithm (ECDSA) [8], which combines ECC elliptic curve cryptography and DSA signature algorithms. This algorithm has the characteristics of small key storage space and high security. In 1999, ECDSA became the standard of ANSI and became the standard of IEEE and NIST in 2000. At present, Bitcoin generally uses the ECDSA algorithm to generate key pairs for trading users and signs the message digest of the data information in the transaction, using the private key of the trading account for signature authentication.

The threshold signature, as a special group digital signature technology, has the authentication function of digital signatures, anti-tampering, and power dispersion characteristics of secret sharing. Group signature is an application in cryptography where only certain members of a group can cooperate with each other to generate a signature that represents the entire group, while verifiers who do not belong to the group can use the public key of the group to verify the generated group signature. The threshold signature algorithm can be divided into the following four stages in terms of composition. The first step is to initialize the system. For algorithms with third-party trusted centers, the trusted center selects system parameters and calculates the partial private keys of each member responsible for distribution. In algorithms without a trusted center, all members of the algorithm jointly agree on system parameters and then use a self-certified public key system to generate their own key. The second step is to generate partial signatures. When a group decides to sign a message, all members of the group will use their own partial private key to sign the message. During this process, some threshold signature algorithms require some members to communicate with each other to exchange certain parameters to generate partial signatures, while others do not. The next step is to aggregate the partial signatures into a complete signature. After all members of the group in the algorithm sign, the generated partial signatures will be sent to a member with a synthetic signature function. The synthesizer of the signature synthesizes the overall signature according to a specific algorithm after receiving the number of partial signatures greater than the threshold value. The last step is to verify the complete signature. After receiving the overall signature, the signature recipient uses a signature verification algorithm to verify whether the overall signature is valid and decide whether to receive the message.

The threshold signature mechanism is more secure than traditional digital signatures. In traditional digital signature mechanisms, once the private key is obtained by the attacker, the security of the entire signature cannot be guaranteed. In the threshold signature mechanism, attackers need to simultaneously break through some private keys of at least t group members to generate a valid signature, which is much more difficult than traditional signature mechanisms. The current threshold signature schemes have generated many threshold signatures with different functions based on different encryption methods, mainly through secret sharing of existing mature signature algorithms such as RSA, ElGamal and DSS [9], [10].

III. Preliminaries

A. Bilinear pairings

Let there exist p-order multiplicative cyclic groups G and $\mathrm{G_T}$, i.e., $|\mathrm{G}| = |\mathrm{G_T}| = p$. g are generating elements of G. The pairing $e : G \times G \to G_T$ is called a bilinear pairing if it satisfies:

1) *Bilinear:* For $\forall a, b \in Z_p$ and $\forall u, v \in G$, there is $e\left(u^a, v^b\right) = e(u, v)^{ab}$;
2) *Non-degeneracy:* $e\left(g, g\right) \neq 1$, where 1 is the unit element of G_T;
3) *Computability:* For $\forall u, v \in G$, there exists an efficient algorithm capable of computing $e\left(u, v\right)$.

B. Discrete logarithm problem

Let G be a cyclic group of order p and g is a generating element of G. The binary group $\left(g, g^x\right)$ is known and solving $x \in Z_p$ is called the discrete logarithm problem, denoted as $DL_g\left(g, g^x\right) = x$. Before quantum computing was applied, the discrete logarithm problem was difficult.

C. Lagrangian interpolation

Let $f\left(x\right)$ be a polynomial function and $\left(x_0, y_0\right), \left(x_1, y_1\right), \cdots, \left(x_n, y_n\right)$ be $n+1$ points, where $x_0, x_1, \cdots, x_n, y_0, y_1, \cdots, y_n \in Z_p$. If $f\left(x_i\right) = y_i$ holds for $\forall \left(x_i, y_i\right), i \in [0, n]$, $f\left(x\right)$ can be uniquely determined according to the Lagrange interpolation method:

$$f\left(x\right) = \sum_{i=0}^{n} \left(y_i \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \right),$$

where the Lagrangian coefficient of the element x_k on the set $\mathrm{S} = \{x_0, x_1, \cdots, x_n\}$ is:

$$\Delta_{k,S}\left(x\right) = \prod_{x_i \in S, x_k \neq x_i} \frac{x - x_i}{x_k - x_i}.$$

IV. System model

As shown in Fig 1, each subdomain in the power IoT model includes three entities: management nodes, perception nodes and storage nodes.

Management node (M): One or more management nodes exist in a subdomain of the force IoT. They have rich network resources and are responsible for communicating with the management nodes of other subdomains for data interconnection. Upon receiving data requests from the management nodes of other subdomains, the management nodes select t of the n storage nodes in the subdomain to generate partial signatures. Then, the management node aggregates the complete threshold signatures from them and returns them along with the data to the management node that initiated the request. It is worth noting that the management node does not participate in secret sharing among the storage nodes, so it does not have the ability to forge signatures.

Perceptual node (P): Perceptual nodes have only limited storage space and computational resources and are responsible for collecting and producing data and handing them over to storage nodes for redundant storage. The perception layer nodes will sign and authorize the storage nodes at each time slot but will not participate in the signing process themselves.

Storage node (S): The storage node has sufficient storage resources to be responsible for storing the data collected by the perception node and generating partial signatures for the data. If the perception node does not provide signature authorization, it cannot generate a valid partial signature in that time slot. In the early

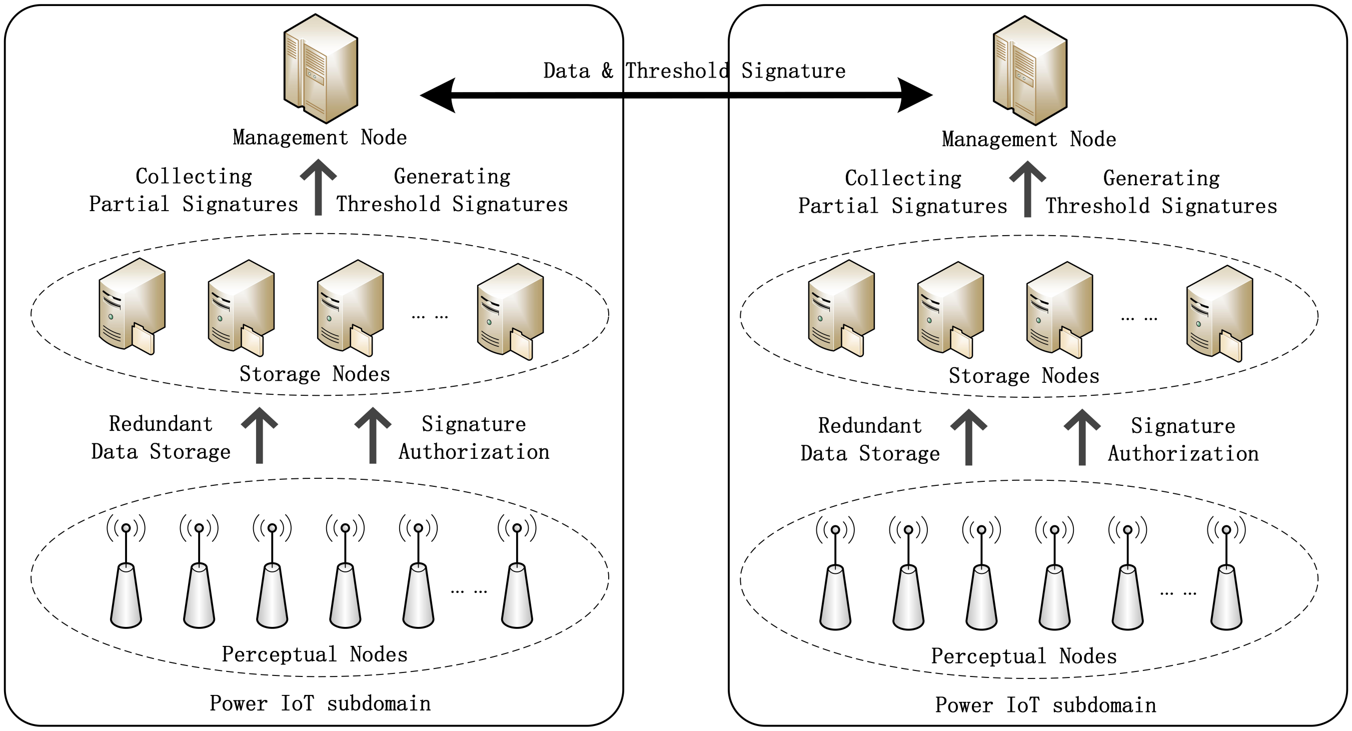

Fig. 1: The system model of subdomain in the power IoT.

stage of system establishment, all storage nodes need to share secretly to generate signature public keys for subdomains.

V. Construction

Assume that the number of stored nodes in the power IoT subdomain is n and the identity serial number is $i \in [1, n]$. The signature threshold is t. Our scheme consists of the following steps.

A. System initialization

Let be G, G_T two p-order multiplicative cyclic groups and g be the generating element of G. Define the bilinear mapping $e : G \times G \to G_T$. Select the collision-resistant one-way hash function $H : \{0,1\}^* \to Z_p$. Generate and expose the public parameters $\{p, g, e, H\}$.

B. Perceptual node initialization

The perception node randomly selects the private key $sk_P \overset{R}{\leftarrow} Z_p$ and computes the public key $pk_P = g^{sk}$. The perception node picks $\varphi \overset{R}{\leftarrow} Z_p$ and computes and publishes $\Phi = g^{\varphi}$. Then, the perception node picks a t-1th order polynomial $F(x) = sk_P + \varphi + a_1 x + \ldots + a_{t-1} x^{t-1}$, $a_i \in Z_p$. For $\forall i \in [1, n]$, the perception node computes $d_i = F(i)$ and publishes $D_i = g^{d_i}$. At time slot 0, the perception node picks $r_0 \overset{R}{\leftarrow} Z_p$. For $\forall i \in [1, n]$, the perception node computes the initial authorization component $\beta_{i,0} = \Phi^{d_i + r_0}$ and hands it to the storage node S_i.

C. Storage node initialization

Storage node S_i randomly selects $sk_i \overset{R}{\leftarrow} Z_p$ as the private key and computes the public key $pk = g^{sk_i}$. Storage node S_i picks a t-1th degree polynomial $f_i(x) = z_i + k_i + a_{i,1} x + \ldots + a_{i,t-1} x^{t-1}$, where $z_i, k_i, a_{i,l} \in Z_p$, $i \in [1, n]$, $l \in [1, t-1]$. S_i computes $Z_i = g^{z_i}$, $K_i = g^{k_i}$, $A_{i,l} = g^{a_{i,l}}$ and $f_i(j)$ and sends them to S_j. Storage node S_j verifies the correctness of $f_i(j)$ by whether the equation $g^{f_i(j)} \overset{?}{=} Z_i K_i \prod_{l=1}^{t-1} {A_{i,l}}^{j^l}$ holds. If all $f_i(j)$ are correct, S_j computes $K = \prod_{l=1}^{n} K_i$, $v_j = \sum_{l=1}^{n} f_l(j)$, $V_j = g^{v_j}$ and publishes them. Thus, the signed public key of the subdomain is then $PK = \prod_{l=1}^{n} Z_i$.

D. Signature authorization

At time slot L, the perception node updates $\beta_{i,L} = \Phi^{r_L} \beta_{i,L-1} = \Phi^{d_i + \sum_{l=0}^{L} r_l}$ for each storage node S_j, where $r_L \in Z_p$. The perception node then computes $R_L = R_{L-1} g^{r_L}$. The perception node sends the authorized update component $(\beta_{i,L}, R_L)$ to the corresponding storage node. The storage node determines whether the update information for that time slot is correct by equation $e(\beta_{i,L}, g) = e(D_i R_L, \Phi)$.

E. Signature

Let the requested data be m. The management node selects t storage nodes to obtain the data and partial signatures. The storage node calculates $\tau_i = \prod_{j=1,j\neq i}^{t} (j/(j-1))$ and $\sigma_i = \beta_{i,L}^{\tau_i H(m)} \Phi^{(sk_i+v_i\tau_i)H(m)} = (\beta_{i,L}^{\tau_i} \Phi^{sk_i+v_i\tau_i})^{H(m)}$, and sends it to the management node along with the data m.

F. Signature aggregation

When all the partial signatures of t storage nodes are collected, the management node aggregates the threshold signatures $\sigma = \prod_{i=1}^{t} \sigma_i$, $\tau = \sum_{i=1}^{t} \tau_i$. The final data signature set (m, σ, τ) is generated.

G. Verification

When the management nodes of other subdomains want to verify the signature of the data, they first aggregate the public key of the storage node to $PK_t = \prod_{i=1}^{t} pk_i$ and then verify whether equation $e(\sigma, g) = e({R_L}^{\tau} Kpk_{DO}PK_tPK, \Phi^2)^{H(m)}$ holds. If the equation holds, it indicates the validity of the threshold signature.

VI. Security Analysis

A. Proof of correctness

The correctness of our scheme is proved as follows.

$$
\begin{aligned}
& e({R_L}^{\tau} Kpk_{DO}PK_tPK, \Phi)^{H(m)} \\
& = e(g^{\varphi+sk_{DO}} \prod\nolimits_{i=0}^{L} {g^{r_i}}^{\tau} \prod\nolimits_{l=1}^{n} Z_i K_i \prod\nolimits_{i=1}^{t} pk_i, \Phi)^{H(m)} \\
& = e(\prod\nolimits_{i=1}^{t} g^{sk_i}, \Phi)^{H(m)} e(g^{F(0)} \prod\nolimits_{i=0}^{L} {g^{r_i}}^{\sum_{i=1}^{t} \tau_i}, \Phi)^{H(m)} \\
& \cdot e(g^{\sum_{l=1}^{n} f_i(0)}, \Phi)^{H(m)} \\
& = e((\prod\nolimits_{i=1}^{t} \Phi^{sk_i+\tau_i(v_i+d_i+\sum_{l=0}^{L} r_l)}), g)^{H(m)} \\
& = e((\prod\nolimits_{i=1}^{t} \beta_{i,L}^{\tau_i} \Phi^{sk_i+v_i\tau_i}), g)^{H(m)} \\
& = e(\sigma, g)
\end{aligned}
$$

B. Non-forgeability

Each partial signature requires the use of the private key of the storage node sk and the authorized component $\beta_{i,L}$. But the storage node only discloses its public key $pk = g^{sk}$. Because the discrete logarithm problem is intractable, no one can infer the private key from the public key. As for the authorization value, it contains the $d_i = f(i)$ generated by the perception node for each storage node. As with the private key, no one can infer $D_i = g^{d_i}$ from d_i. Therefore, no one can forge a valid partial signature, and hence a valid threshold signature for the creation.

C. Non-repudiation

The user's private key and public key correspond to each other, and the public keys of t stored nodes are required to verify the threshold signature. Since the signature is not forgeable, the threshold signature can be verified successfully if and only if the t public keys used correspond to the storage nodes one by one. Therefore, the user cannot deny his participation in the generation of the threshold signature.

D. Forward security

In each time slot, the perception node updates the authorization component for the storage node $\beta_{i,L} = \Phi^{r_L} \beta_{i,L-1} = \Phi^{d_i+\sum_{l=0}^{L} r_l}$. $\beta_{i,L}$ in which both the user's identity is bound by $d_i = f(i)$ and the random values r_i of all historical time periods are accumulated. But the perceptual nodes are only publicly available to the public as $D_i = g^{d_i}$ and $R_L = R_{L-1} g^{r_L}$. Because the discrete logarithm problem is difficult, no one can be informed of d_i and $\sum_{l=0}^{L} r_l$ from D_i and R_L, and thus cannot falsify $\beta_{i,L}$. At time slot L, for the storage node that has the authorized component $\beta_{i,L-1}$ at time slot L-1, he also cannot compute r_L by R_L and R_{L-1}, and thus Φ^{r_L} . Therefore, at time slot L, he will be unable to generate a valid partial signature because of the missing authorization component for that time slot.

VII. Performance Evaluation

In this section, we analyze the computational cost and communication cost of the scheme. In cryptography, the exponential operation and the bilinear mapping operation are the two operations with the highest computational complexity. Other operations (e.g., point multiplication operation, number multiplication operation, hash operation) have negligible computational cost. t denotes the size of the threshold, E denotes the exponential operation in G, E_T denotes the exponential operation in G_T, P denotes the bilinear pairing operation, $|G|$ denotes the length of the element in G, $|Z_p|$ denotes Z_p the length of the elements in G. The execution times of the mathematical operations are given in Table 1. The arithmetic tests were performed on Ubuntu 16.04 LTS using Python 3.8.5 and the Charm-Crypto framework [11]. The test platform was a desktop computer configured with an 11th generation Intel Core i5 processor. In addition, in terms of communication cost, $|G| = 1024$bit and $|Z_p| = 160$bit.

TABLE I: Execution Time of Mathematical Operations

Type of operation	Average computation time (ns)	Standard deviation (ns)
Exponential operations in G	743.92	29.88
Exponential operations in G_T	75.81	10.53
Bilinear mapping	479.40	19.65
Point product operation	4.61	2.38
Hash function	4.12	3.69

TABLE II: Comparison of Computation and Communication Costs

Scheme	Time-sharing authorization	Partial signature generation	Signature aggregation	Signature verification	Signature size
[12]	-	tE	tE	$2E$	$2\|G\|+\|Z_p\|$
[13]	-	tE	-	$2E$	$2\|Z_p\|$
Our scheme	$2tE$	$2tE$	-	$E+E_T+2P$	$\|G\|+\|Z_p\|$

Table 2 shows the comparison of our scheme with schemes [12] and [13]. Schemes [12] and [13] are two practical and efficient threshold signature algorithms. Therefore, we show the computation and communication performance of our schemes by scheme comparison. Since our scheme allows perceptive nodes to authorize storage nodes in time slots for forward security, there is a computational cost of authorization. In terms of computational cost of signature generation, our scheme is higher than schemes [12] and [13], also because each storage node needs to perform one additional exponential operation on the authorization component provided by the perception node. In terms of signature verification, our scheme is slightly higher than schemes [12] and [13]. This is because in order to verify whether the signature is generated at time L, one exponential operation needs to be performed at verification to compute ${R_L}^{\tau}$. In terms of communication cost, our scheme is slightly lower than that of scheme [12] but higher than that of scheme [13]. In scheme [13], the signature is cropped once after it is generated. First, only the horizontal coordinates of the signature are kept, and then the modulo operation is taken. Therefore, the actual length of the signature passed in [13] is compressed from $|G|+|Z_p|$ to $2|Z_p|$. Our scheme could also use similar compression to reduce the signature size to $2|Z_p|$. But this would increase the risk of signature forgery.

VIII. Conclusion

With the rapid development of IoT technology, the deep integration of power system and IoT has formed an intelligent power IoT. However, Ensuring the integrity and authenticity of transmitted data stands as a critical imperative for safeguarding the operation of power Internet of Things (IoT), given the pervasive presence of data security concerns across the networked environment. In this paper, we first reviewed the development of digital signature technology and introduce the advantages of threshold signature. Then, we proposed an authorizable threshold signature scheme for IoT to protect the secure transmission of power data. Subsequently, we proved the correctness of the scheme. The security analysis showed that our scheme satisfies non-forgeability, non-repudiation and forward security. The performance analysis compared the computational and communication costs of our scheme with existing schemes.

References

[1] M. Zeng, Y. Wang, M. Li, H. Dong, X. Zhang, H. Wang, X. Huo, and Z. Zhang, "Preliminary study on the architecture and implementation plan of the universal power internet of things system," *Smart Power*, vol. 47, no. 4, pp. 1–7, 2019.

[2] C. Dong, S. Xia, B. Sun, H. He, H. Yu, and L. Zhang, "Vulnerability analysis of concentrators in power consumption information collection systems," *Electrical Measurement and Instrumentation*, vol. 57, no. 8, pp. 128–134, 2020.

[3] G. Tong, H. Cao, and W. Wang, "Safety is technology - technological progress is the prerequisite and foundation for ensuring power safety," *Smart Power*, vol. 47, no. 8, pp. 12–17, 2019.

[4] E. A. Adeniyi, P. B. Falola, M. S. Maashi, M. Aljebreen, and S. Bharany, "Secure sensitive data sharing using rsa and elgamal cryptographic algorithms with hash functions," *Information*, vol. 13, no. 10, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/10/442

[5] Y. S. Prakash, P. H. Narayan, R. Ramakrishna, G. S. Sandeep, V. S. S. Ramesh, and I. Balaraju, "Digital signatures and el gamal scheme integration for secure data transmission in digital transaction survey," in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*. IEEE, 2022, pp. 892–898.

[6] J. Blocki and S. Lee, "On the multi-user security of short schnorr signatures with preprocessing," in *Annual International*

Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2022, pp. 614–643.

[7] R. Khurana and E. Narwal, “Analysis of code-based digital signature schemes,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 5, pp. 5534–5541, 2023.

[8] H. Xiong, C. Jin, M. Alazab, K.-H. Yeh, H. Wang, T. R. Gadekallu, W. Wang, and C. Su, “On the design of blockchain-based ecdsa with fault-tolerant batch verification protocol for blockchain-enabled iomt,” *IEEE journal of biomedical and health informatics*, vol. 26, no. 5, pp. 1977–1986, 2021.

[9] C. Komlo and I. Goldberg, “Frost: flexible round-optimized schnorr threshold signatures,” in *Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers 27*. Springer, 2021, pp. 34–65.

[10] R. Gennaro and S. Goldfeder, “Fast multiparty threshold ecdsa with fast trustless setup,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1179–1194.

[11] 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, pp. 111–128, 2013.

[12] L. Wang, M. Hu, Z. Jia, Y. Cheng, J. Fu, Y. Wang, and B. Gong, “Identity-based threshold group signature scheme of blockchain verification,” in *Trusted Computing and Information Security: 13th Chinese Conference, CTCIS 2019, Shanghai, China, October 24–27, 2019, Revised Selected Papers 13*. Springer, 2020, pp. 144–158.

[13] H. Wang, W. Ma, F. Deng, H. Zheng, and Q. Wu, “Dynamic threshold ecdsa signature and application to asset custody in blockchain,” *Journal of Information Security and Applications*, vol. 61, p. 102805, 2021.