Computer Systems Science & Engineering
DOI:10.32604/csse.2022.017657
Article

Tech Science Press

A CPK-Based Identity Authentication Scheme for IoT

Mingming Zhang[1], Jiaming Mao[1,*], Yuanyuan Ma[2], Liangjie Xu[3], Chuanjun Wang[1], Ran Zhao[1], Zhihao Li[3], Lu Chen[4] and Wenbing Zhao[5]

[1]State Grid Jiangsu Electric Power Co., Ltd. Information & Telecommunication Branch, Nanjing, 210008, China
[2]State Grid Key Laboratory of Information & Network Security, Institute of Information and Communication, Global Energy Interconnection Research Institute, Nanjing, 210003, China
[3]Anhui Jiyuan Software Co. Ltd, SGITG, Hefei, 230088, China
[4]Engineering Research Center of Post Big Data Technology and Application of Jiangsu Province, Research and Development Center of Post Industry Technology of the State Posts Bureau (Internet of Things Technology), Engineering Research Center of Broadband Wireless Communication Technology of the Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China
[5]Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, 44115, USA
*Corresponding Author: Jiaming Mao. Email: seumao@qq.com
Received: 06 February 2021; Accepted: 09 April 2021

Abstract: As the power Internet of Things (IoT) enters the security construction stage, the massive use of perception layer devices urgently requires an identity authentication scheme that considers both security and practicality. The existing public key infrastructure (PKI)-based security authentication scheme is currently difficult to apply in many terminals in IoT. Its key distribution and management costs are high, which hinders the development of power IoT security construction. Combined Public Key (CPK) technology uses a small number of seeds to generate unlimited public keys. It is very suitable for identity authentication in the power Internet of Things. In this paper, we propose a novel identity authentication scheme for power IoT. The scheme combines the physical unclonable function (PUF) with improved CPK technology to achieve mutual identity authentication between power IoT terminals and servers. The proposed scheme does not require third-party authentication and improves the security of identity authentication for power IoT. Moreover, the scheme reduces the resource consumption of power IoT devices. The improved CPK algorithm solves the key collision problem, and the third party only needs to save the private key and the public key matrix. Experimental results show that the amount of storage resources occupied in our scheme is small. The proposed scheme is more suitable for the power IoT.

Keywords: Power Internet of Things; terminal equipment; identity authentication; CPK; PUF

1 Introduction

In recent years, the development of the power Internet of Things (IoT) has entered the security construction stage. Scholars are increasingly concerned about security issues in the IoT [1–3]. Smart terminals with specific sensing, computing and execution functions are deployed in power IoT. The secure identity authentication of a terminal device in the perception layer is an essential prerequisite for establishing a secure connection to the power IoT. Technologies for identity authentication of the terminal devices can be divided into cryptography-based authentication technologies and non-cryptography authentication technologies. Traditional cryptography identity authentication method is the public key infrastructure/certificate of authority (PKI/CA) method [4]. However, this method is difficult to support many users, and it is challenging to implement in some low-end smart devices.. These limitations have hindered the wider use of PKI/CA in IoT security. Because the PKI/CA system needs to issue a digital certificate for each device.In the PKI/CA system, many certificates need to be maintained and managed. Online exchanges are required, and the overall construction and maintenance costs are very high.

Identity-based encryption (IBE) is a public key encryption technology that can use any string as a valid public key [5–7]. The recipient's identity information is bound to the public key, and the trusted key generation center (KGC) calculates the corresponding private key. The key is then distributed without the issuance and management of certificates. In 2003, Boneh and Franklin realized the first IBE scheme based on bilinear mapping [8] and proved its security under the random oracle model. Compared with PKI authentication technology, IBE has lower overhead and can simplify PKI certificate management. In addition to Boneh and Franklin's method, the combined public key (CPK) technology was proposed by the Chinese scholar Nan Xianghao [9]. The CPK technology uses a small number of seeds to generate unlimited public keys. The current implementation method is mainly based on the elliptic curve [10]. CPK uses the user's identification number as a parameter to perform a certain number of mappings, obtains the private key according to the mapping value, and then calculates the user's public key according to the mapping value and the seed matrix. A significant advantage of CPK is that it has no need for third-party certification. The third party only needs to save the key matrixes. The amount of storage resource occupied is small, and the public key does not need to be transmitted. It is convenient and fast. Thus, CPK can meet the large-scale authentication needs of massive power IoT devices.

In addition, with hardware security research development, non-cryptography identity authentication technology has become a hot spot. Scholars such as Pappu provided the first formal conceptualizations of physical unclonable functions (PUFs) [11]. PUFs are widely present in existing mobile and Internet terminal devices.PUF can realize the unique function of excitation signal and response signal. Existing PUF implementation methods mainly include non-electronic PUFs, analog circuit PUFs, and digital circuit PUFs. Digital circuit PUF technology mainly uses the random process deviations in the internal circuit or lighting process in the chip manufacturing process to generate the device's physical fingerprint. The technology cannot be cloned, and it has uniqueness, reliability, security, unpredictability, and low computational costs. The implementation methods for PUF technology include storage-based PUF methods and delay-based PUF methods. Onboard storage devices for the IoT node include the dynamic random access memory (DRAM) and static random access memory (SRAM) [12]. These devices can be used as endogenous PUFs. Different DRAMs of the same model can obtain different response values after using the same excitation. Therefore, the response value can be used as the IoT node's inherent and unique identification information. For SRAM, each logic unit has a different state bias for process reasons, so the memory unit address can be used as an excitation input to realize node identification.

Because of the lightweight and secure identity authentication requirements of power IoT devices, our scheme combines PUF with improved CPK technology to solve the secure identity authentication and resource limitation problems of power IoT devices. Furthermore, the scheme improves the security of identity authentication for power IoT devices.

2 Related Work

Currently, the identity authentication methods of devices in the IoT perception layer mainly include cryptography-based authentication and non-cryptography authentication. Similar to the IBE mechanism, Nan Xianghao first proposed CPK identification. CPK uses a small number of parameters to construct a key seed matrix. Additionally, CPK generates many public and private key pairs by combining the seed matrix elements to achieve super-large-scale identification-based key production and distribution [13–15]. The CPK system does not require a third party to stay online when used, so the verification efficiency is higher than that of the PKI. With the development of identification technology, CPK has become widely used in identity authentication. To address the RFID [16] identity authentication problem, Ying Cui et al. proposed a CPK-based RFID authentication protocol [17] to realize the mutual security authentication of an RFID reader and a tag. The protocol is very suitable for an RFID system's hardware environment with limited performance. It solves the power IoT's key management problem in terminal devices. Guangquan Xu et al. [18] proposed an improved CPK algorithm based on single and double mixed matrixes called HyCPK. The key management center(KMC) generates a basic key matrix and an auxiliary key matrix simultaneously. Both matrixes are used to calculate the user's key. Additionally, the CPK identity is composed of the user identity and corresponding effective date, making the verification process more convenient. This algorithm can solve the large-scale authentication problem. Xiaoting Ma et al. [19] combined the the identification key generation algorithm with the blockchain's decentralized characteristics to build a PKI and IBC alliance chain model to achieve safe and efficient cross-domain authentication and re-authentication of the external domain ISE. Moreover, efficiency analysis proved the practicality of the scheme.

However, the CPK identity system may face key collision problems when generating identification keys [20]. It poses a security threat to the system and hinders the large-scale application of CPK authentication. One method to solve the key collision problem [21,22] is to set the seed matrix's private key factor's selection condition or constrain the elliptic curve parameters.

In recent years, PUF technology has been widely used in lightweight identity authentication. These methods include directly using the sequence generated by a PUF as a lightweight authentication identifier and using a PUF to generate and store keys [23–26]. Liu et al. [27] designed a lightweight mutual authentication protocol, called MPUF-HB. It uses the 2-level PUF circuit to address the defect that the HB protocol cluster can only perform one-way authentication. This protocol has high security with low resource usage. V. P. Yanambaka proposed a PUF-based authentication scheme for medical IoT (IoMT) devices [28]. The scheme uses PUF to generate encryption keys, reducing the need for processors. There is no storage in the server memory related to MIoT devices. The time required to verify the device is 1.2 s to 1.5 s.

Kocabas et al. [29] proposed a reverse PUF authentication protocol to realize mutual authentication and key exchange. However, this protocol also requires a large number of CRPs to be extracted in advance. Many searches and XOR operations need to be performed during each authentication process. So it consumes considerable computing resources. To reduce the risk of authentication key exposure and the authentication server's burden, Byoungkoo Kim et al. [30] proposed a PUF-based authentication scheme for IoT devices. Through interaction with the device, only a single CRP is stored and updated. Itis different from the existing technology of storing all CRPs generated by PUF technology in the authentication server. The scheme can also use the key generated by a CRP to encrypt the authentication message to achieve more secure device authentication. At present, the primary authentication technologies that can be applied to the IoT are authentication based on public key algorithms. Key management in these methods incurs considerable costs. Combining hardware chips and lightweight cryptography to support IoT devices' identity authentication can improve authentication security without consuming too many resources.

3 Preliminaries

3.1 CPK Identification Key Algorithm Principle

3.1.1 Building a Seed Key Matrix

Suppose the elliptic curve equation $E\colon y^2 = (x^3 + ax + b) mod p$. We build the parameters $T = \{a, b, G, n, p\}$, where p is an integer, a and b are integers belonging to the finite field F_P, G is the base point on the elliptic curve $E(F_P)$, and $G = (X_G, Y_G)$. The prime number n is the order of the base point G. *SSK* is the private key seed matrix, and *PSK* is the public key seed matrix.

$$SSK = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{m1} \\ r_{21} & r_{22} & \cdots & r_{m2} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mh} \end{pmatrix} \tag{1}$$

$$PSK = \begin{pmatrix} (x_{11}, y_{11}) & (x_{12}, y_{12}) & \cdots & (x_{1h}, y_{1h}) \\ (x_{21}, y_{21}) & (x_{22}, y_{22}) & \cdots & (x_{2h}, y_{2h}) \\ \cdots & \cdots & \cdots & \cdots \\ (x_{m1}, y_{m1}) & (x_{m2}, y_{m2}) & \cdots & (x_{mh}, y_{mh}) \end{pmatrix} \tag{2}$$

3.1.2 Generation of Identification Keys

Let the public key be *PK*, let the private key be *sk*, and let the identification key calculation process be as follows:

(1) Hash the user's identity through the hash algorithm to obtain a fixed-length result value, which is recorded as *data*; *data* is then used as an intermediate variable: $Hash(identity) = data = MAP$.

(2) Adopt the block cipher algorithm as a row mapping algorithm. Let the encryption algorithm be E, let the key be *ROWKEY*, and encrypt *data* circularly. The input of the block cipher is MAP_i and the output is MAP_{i+1}, that is, $E_{ROWKEY}(MAP_i) = MAP_{i+1}$. Assuming that the length of the output by the hash algorithm is 160 bits, when $m = 2^\wedge 5 = 32$, the result generated by the hash algorithm is recorded as *YS*. Then,

$$YS = Hash(identity) = i_1, i_2, \ldots, i_{32} \tag{3}$$

Here, $i_k (1 \geq k \geq 32)$ is a 5 bits string that can represent a number from 0 to 31. i_1 to i_{32} represent the respective row coordinates. The column coordinate starts from 0 to generate the row coordinate sequence $\{(i_1, 1), (i_2, 2), \ldots, (i_h, h)\}$.

(3) The row coordinate sequence $\{(i_1, 1), (i_2, 2), \ldots, (i_h, h)\}$ generated by the row mapping algorithm selects the corresponding matrix element in the private key matrix as the private key factor. For all private key factors, a modular addition operation is performed to obtain the user's private key:

$$sk = \left(\sum_{j=1}^{h} r_{i_j,j}\ mod\, n\right) mod\, n = \left(r_{i_1,1} + r_{i_2,2} + \ldots + r_{i_h,2}\right) mod\, n \tag{4}$$

The purpose of the modulo operation is to eliminate the linear relationship between the private key factors. However, for all private keys in the same domain, it is possible to satisfy

$$sk = \left(r_{i_1,1} + r_{i_2,2} + \ldots + r_{i_h,h}\right) mod\, n = \left(r_{i_1,1} + r_{i_2,2} + \ldots + r_{i_h,h}\right) \tag{5}$$

At this time, users will face the threat of conspiracy attacks by conspirators.

Similarly, the row coordinate sequence$\{(i_1, 1), (i_2, 2), \ldots, (i_h, h)\}$ generated by the row mapping algorithm selects the corresponding matrix element in the public key matrix PSK as the public key factor. Elements are combined and added to obtain the user's public key.

$$PK = \left(R_{i_1,1} + R_{i_2,2} + \ldots + R_{i_h,h}\right) = \left(r_{i_1,1} + r_{i_2,2} + \ldots + r_{i_h,h}\right) \times \mathrm{G} = (\sum_{j=1}^{h} r_{i_j,j}\ modn) \times G = sk \times G \tag{6}$$

3.2 Basic Principles of a PUF

The input and output of a PUF have a complicated functional relationship. Given a stimulus C_i, by inputting the stimulus into the PUF circuit, the corresponding response R_i can be generated:

$$R_i = PUF(C_i) \tag{7}$$

Due to the differences in PUFs, the response generated by each PUF cannot be cloned. PUFs have uniqueness, reliability, safety, and unpredictability. There are many ways to implement PUFs, such as PUFs based on ring oscillators [31], PUFs based on arbiters [32], and SRAM PUFs [33,34]. An SRAM PUF is a delayed PUF with a simple structure and low resource usage. When the SRAM is powered on and not initialized, its bistable logic unit will first enter an unstable state and then return to a stable state after hovering in the unstable state. For process reasons, each logic unit in SRAM has a different state bias. The memory cell address can be used as the excitation input, and the value in the corresponding address unit can be read after the SRAM is powered on as the PUF output response. The specific working principle is shown in Fig. 1. By observing the stable state of an SRAM unit, SRAM PUF can be realized.

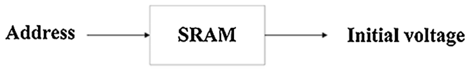

Figure 1: The working principle of SRAM PUF

4 Identity Authentication for Power IoT devices

4.1 Authentication Model

The power IoT device identity authentication model based on the cryptography identification algorithm and PUF does not require the CA center. The identity authentication model of the IoT terminal devices is shown in Fig. 2. It consists of the IoT terminal device, the authentication server, and the KMC:

IoT terminal devices: The IoT terminal devices are the authentication initiators. Each device has a unique identification ID, and each device has an external SRAM storage chip.

Authentication server: The authentication server is the verifier that can complete registration, identification generation, and mutual authentication for power IoT terminal devices.

KMC: The KMC generates a public-private key pair of the device according to its unique identification ID. The KMC discloses the public key and sends the private key to the terminal device through a secure channel.

4.2 Power IoT Device Identity Authentication Scheme Based on CPK and PUFs

This scheme combines the CPK with PUFs to achieve the mutual authentication of power IoT devices. It includes the registration phase and the server authentication phase, improves the security of identity authentication of power IoT devices, and reduces the terminal device's resource consumption. The program flow is shown in Fig. 3.

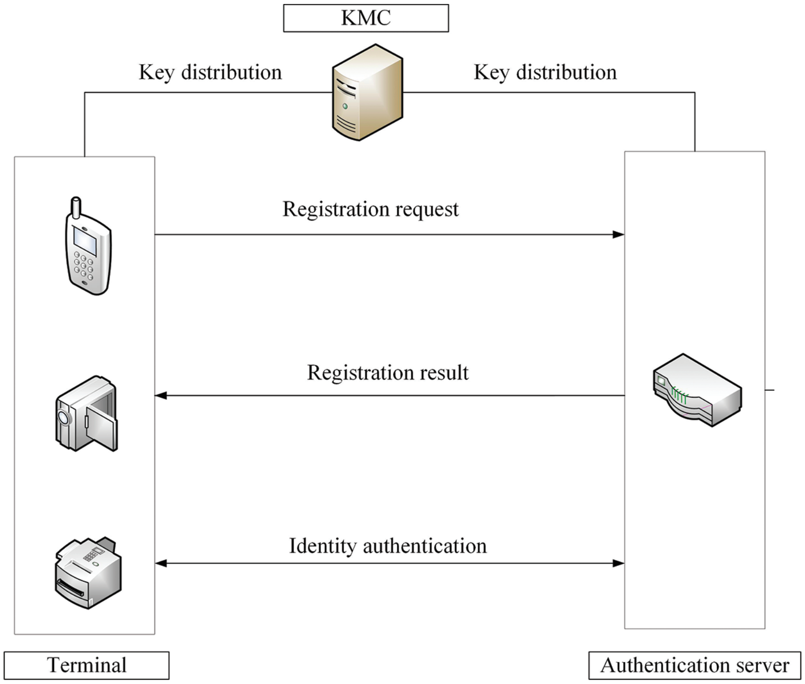

Figure 2: Identity authentication model of power IoT devices

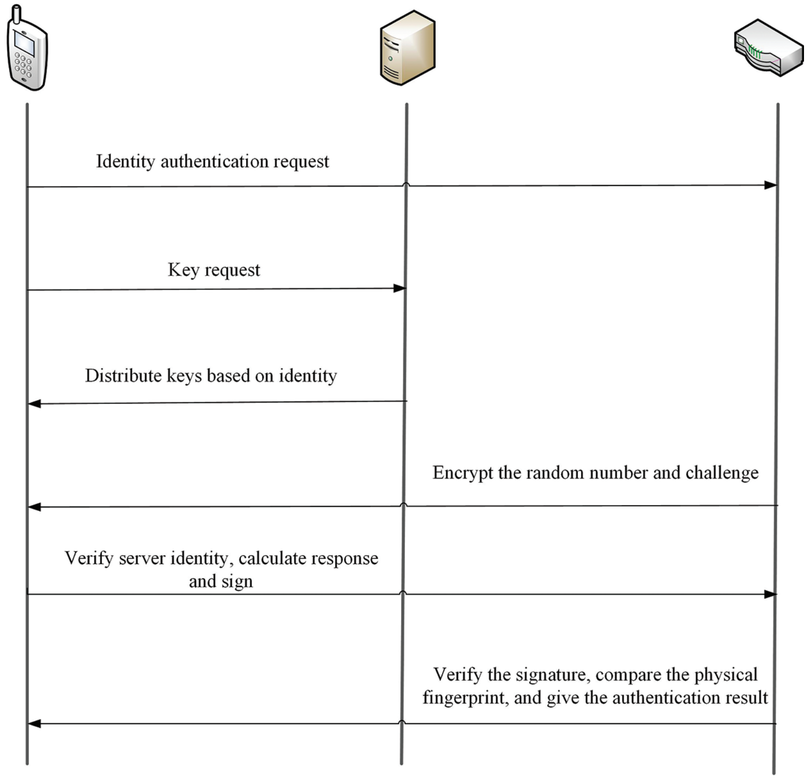
Identity authentication request
Key request
Distribute keys based on identity
Encrypt the random number and challenge
Verify server identity, calculate response and sign
Verify the signature, compare the physical fingerprint, and give the authentication result

Figure 3: The process of the IoT identity authentication scheme based on CPKs and PUFs

4.2.1 Registration Phase

Step 1. The device to be registered submits a series of device attributes to the server.

Step 2. The server extracts the device's attributes to be registered, including the terminal device's IP, MAC address, machine name, operating system type, and other information. It also extracts the terminal behavior characteristics, uses deep learning methods for data analysis, and obtains the unique identification of the device through calculation.

Step 3. The server stores the device identification ID in the database and sends it to the terminal device.

Step 4. The device to be registered sends a registration application to the server and submits the unique identification ID of the device.

Step 5. The registration module verifies the correctness of the device identification ID, selects a series of challenge information, and sends the challenge information to the device to be registered.

Step 6. The device to be registered enters the challenge information into the PUF to obtain the response information, which is the device's physical fingerprint and returns it to the server. The server stores the device ID-challenge-response information in the device fingerprint database.

4.2.2 Certification Phase

Step 1. The terminal device sends an authentication application to the server and submits the identification ID of the device.

Step 2. The terminal device applies to the KMC to obtain the key. The KMC generates the terminal public-private key pair$\{SK_u, PK_u = SK_u \times G\}$ according to the identity of the device and sends it to the terminal through a secure channel. The terminal device saves its private key.

Step 3. The server selects a challenge of the device to be authenticated. The random number generator generates a random number a and sends it to the device to be authenticated and the server. The server obtains PK_u of the terminal device, encrypts the challenge and a, and sends them to the terminal device: $E_{PK_u}(challenge, a) = C$.

Step 4. The terminal device uses its SK_u to decrypt the information sent by the server: $D_{SK_u}(C) = (challenge, a)$. In addition, the terminal device obtains the challenge and the random number a, verifies whether the random number is correct, and completes the server's authentication if it is correct. Then, the device generates response information based on the challenge. The terminal device hashes the information and signs it with the private key $SK_u : Sign_{SK_u}(data) = \sigma$. Then, the terminal device sends the information to the server.

Step 5. The server receives the information sent by the terminal device and uses PK_u of the device to verify it: $Vrfy_{PK_u}(\sigma) = data$. If the verification fails, the device request is rejected, and the authentication fails. Otherwise, the response value in the fingerprint database is matched by the same hash operation. If the result's distance exceeds the preset threshold, the device request is rejected, and the authentication fails. If the calculation result's distance is less than or equal to the preset threshold, the authentication is successful. In addition, the server deletes the corresponding challenge-response pair in the fingerprint database.

5 Improved CPK Identification Key Generation

5.1 Key Collision Problem

CPK identification key generation is completed by mapping the identity. In the process of generating public-private key pairs, key collisions may occur. As shown in Eq. 8, the public-private key pairs of different users are the same, resulting in the nonrepudiation of communication between users. In a large-scale power IoT environment, the number of terminal devices is large, and the key collision problem will affect the entire authentication system's security.

$$\sum_{i=1}^{t} r_{a_i,i} mod(n) = \sum_{i=1}^{t} r_{b_i,i} mod(n) \tag{8}$$

According to Formula 8, there are two reasons for key collisions:

1. $a_i = b_i$. The row sequence obtained after mapping different device identifiers is the same, which means that the hash results of the identifiers are the same.
2. The row sequence is different, but the combined private key calculation result is the same.

The security of the hash algorithm is not within the security of the CPK algorithm. Therefore, the source of the key collision problem is that the row coordinate sequences for generating the user ID are different, and the corresponding key factors are combined to obtain the same private key.

5.2 CPK Identification Key Generation Algorithm Without Key Collision

For the key collision problem, we propose an improved CPK solution without key collision. The improved CPK identification key generation algorithm is as follows:

(a) Device A applies for registration to the KMC and submits a unique identification ID.

(b) The KMC sequentially completes initialization, generates parameters $T = \{a, b, G, n, p\}$, and builds a seed key matrix. According to the terminal device's identity ID and seed key matrix, the device's initial combined public key PK_A and initial private key sk_A are calculated through the hash transformation and row mapping algorithm:

$$sk_A = (\sum_{j=1}^{h} r_{i_j,j}) mod\, n = \left(r_{i_1,1} + r_{i_2,2} + \cdots + r_{i_h,h}\right) mod\, n \tag{9}$$

$$PK_A = \left(R_{i_1,1} + R_{i_2,2} + \cdots + R_{i_h,h}\right) = sk_A \times G \tag{10}$$

(c) The Bloom filter is initialized, and the parameters $\{m, n, k, p, n\}$ are selected. The memory size is defined as m bits, k hash functions are selected, the error rate is set to p, and the number of combined public keys is set to n. Then, each bit position is set to 0.

(d) After the combined public key of the terminal device in the system is processed by a series of mapping algorithms, the corresponding positions of the Bloom filter are set to 1. According to the selected k hash functions, the KMC maps the generated combined public key to the Bloom filter for comparison and detects whether it is duplicated.

(e) If the detection is repeated, $x > 0$ is randomly selected, and the public key change factor $Y = x \times G$ is calculated. Then, the new device combined public key is $PK = PK_A + Y$. Step (c) is repeated until the detection result is not repeated.

(f) The device identification private key $sk = sk_A + x$ is computed.

(g) The public identification key of the terminal device is added to the Bloom filter through a series of operations, and the corresponding positions to are set 1.

(h) The KMC sends the combined private key sk and public key change factor Y to the device through a secure channel.

5.3 Correctness

Theorem 1 The generated device private key satisfies the definition of the private key in the combined public key cryptography.

According to the improved CPK identification key generation algorithm, the key collision problem can be solved adequately. The combined public key and private key of user A have the following relationship:

$$PK = PK_A + Y = sk_A \times G + x \times G = (sk_A + x) \times G = sk \times G \tag{11}$$

The equation shows that after the public key change factor is added, the relationship between the user's combined public key and the private key does not change. As the public key change factor is added and compared with the Bloom filter, the key collision problem is solved.

6 Experiment and Analysis

6.1 Security Analysis

6.1.1 Confidentiality of Information Transmission

The proposed scheme is based on CPK and SRAM PUF. It adopts a two-factor authentication method. When the CPK signature is successfully verified and the solved physical fingerprint is consistent with the database's fingerprint information, the device can be successfully authenticated. If only one of these conditions is met, authentication cannot be successfully completed, and thus authentication is more secure than authentication using cryptographic methods or PUFs alone. The KMC uniformly generates the public and private keys of devices, and the unique identity of the device determines the public key. By encrypting and signing the information, the confidentiality of the transmitted information can be guaranteed, and the information can be protected from being stolen and manipulated.

6.1.2 Communication Key Security

The improved CPK identification cryptography algorithm used in our scheme is based on the discrete logarithm problem (ECDH) on the elliptic curve [35]. Given a base point G on the elliptic curve and a random integer r, it is easy to solve $Q = r * G$. Given a base point G, knowing $Q = r * G$, it is difficult to find the integer r. The security of this scheme is the same as that of the elliptic curve cryptosystem. This paper improves the CPK identification key generation method and eliminates the key collision problem in the CPK cryptosystem. Therefore, the security threat caused by key collisions when applied in a large-scale power IoT environment is eliminated. During the authentication process, the two parties also agreed on a random parameter a and a PUF response. Even if the attacker performs a replay attack, identity authentication cannot be passed.

6.2 Comparison

Our solution combines CPKs and SRAM PUFs to achieve dual and mutual authentication between IoT devices and the server. The PKI-based identity authentication scheme requires the online operation of the certificate catalog, and the generation of public keys is not scaled. Thus, PKI cannot satisfactorily meet the needs of power IoT device authentication. Private keys and certificates are securely issued to users. The public key of the verified party can be obtained only through the public key matrix, and the storage of the public key matrix is small and does not occupy too many resources. In addition, the physical fingerprint is generated by the endogenous or external SRAM PUFs of terminal devices; the characteristics of uniqueness, reliability, security, unpredictability, and low computational costs are exploited to achieve dual authentication of the devices. Therefore, the method enhances the security of device authentication. Tab. 1 compares the characteristics of the different identity authentication methods of IoT terminal devices.

Table 1: Comparison of the characteristics of identity authentication schemes for IoT devices

	PKI-based identity authentication	CPK-based identity authentication	Our method
Key generation	Public keys and certificates are randomly generated in a distributed manner	Uniformly generated by the KMC based on identity	Uniformly generated by the KMC based on the identity calculated by the server
Key storage	The key is stored in the online database, which requires a large amount of maintenance and takes up a large amount of storage space	Stored centrally or decentrally and occupies little storage space	Stored centrally or decentrally and occupies little storage space
Key distribution and management	Static distribution and dynamic management; the device retains the private key	Static distribution, static management, and distribution through a secure channel	Static distribution, static management, and distribution through a secure channel
Authentication method	Traditional public key cryptography	Lightweight cryptography	Lightweight cryptography and PUF physical fingerprint

The key collision problem usually occurs when the device identity is used to generate the row coordinate sequence for the selection and combination operation to obtain the public-private key pair. Our method uses an improved anti-key collision CPK identification key generation algorithm. When large-scale applications are not considered, most methods [21,22] restrict the selection of key seeds to eliminate the key collision problem of CPKs. A database deduplication method is used to conduct public key selection in Li [36]. However, this method requires a database to store all public keys for comparison, which requires considerable storage space and takes up more resources. Our method uses the Bloom filter to solve the key collision problem of the CPK without taking up too many resources. The following compares the key collision problem solved in this paper with the solution in [36]. First, we define some parameters in Tab. 2.

Table 2: List of parameters

Parameter	Description
n	The number of public keys stored
p	The false-positive rate of the Bloom filter
m	The size of storage space required (bits)
k	The number of Bloom filter hash functions
L	The length of the combined public key (bits)
x	The number of forks when using the index query

According to the number of combined public keys and the false-positive rate, first, calculate the required storage space m:

$$m = -\frac{n \ln p}{(\ln 2)^2} \tag{12}$$

Then, calculate the required number of hash functions k according to m and n:

$$k = \ln 2 \cdot \frac{m}{n} \tag{13}$$

Add n combined public keys to the Bloom filter and let the length of each combined public key be L bits. The storage space required by the method in [36] is $m = n \cdot L$ (bits), and the storage space required by this scheme is (when the false-positive rate is 1%):

$$m = -\frac{n \ln(1\%)}{(ln2)^2} \approx 9.585n \tag{14}$$

The result shows that the storage space required by the anti-key collision CPK identification key generation algorithm of this scheme is only related to the number of combined public keys and is unrelated to the length. The storage space required by the method in Li [36] is linearly related to the length of the combined public key. Fig. 4 shows the storage space occupied by our method (the false-positive rate is 0.01) and by the method in [36]. Figs. 5 and 6 shows the storage space occupied by our method (the false positive rates are 0.02 and 0.05, respectively) and by the method in Li [36].

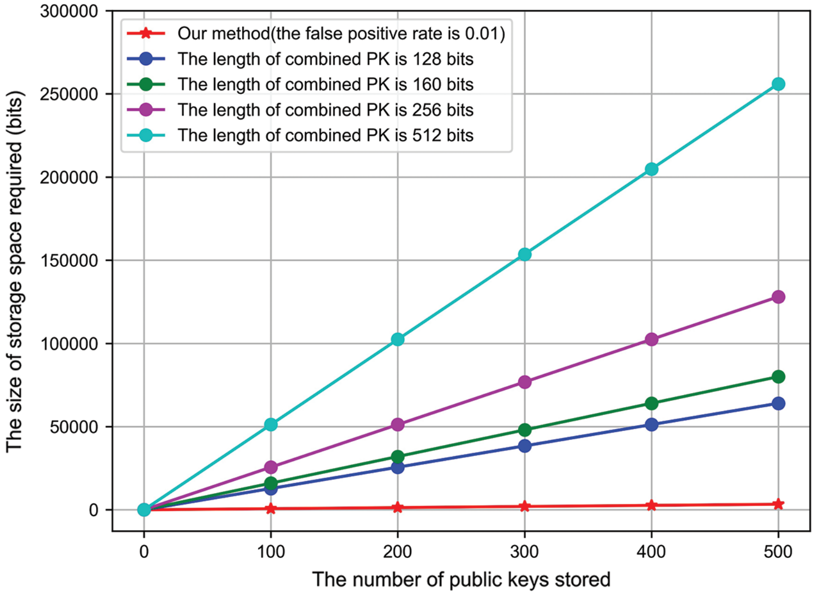

Figure 4: Comparison of the storage space occupied (the false-positive rate of our method is 0.01)

We selected the case where the length of the public key is 128, 256, and 512 bits. The storage space required by this scheme is less than that of [36], and it takes up fewer resources without key collisions. In addition, the query time complexity of the method in [36] is related to the number of combined public keys n. In contrast, our method is only related to the number of hash functions, which is constant. Tab. 3 compares the characteristics of the anti-key collision CPK algorithm of this scheme with the characteristics of other methods.

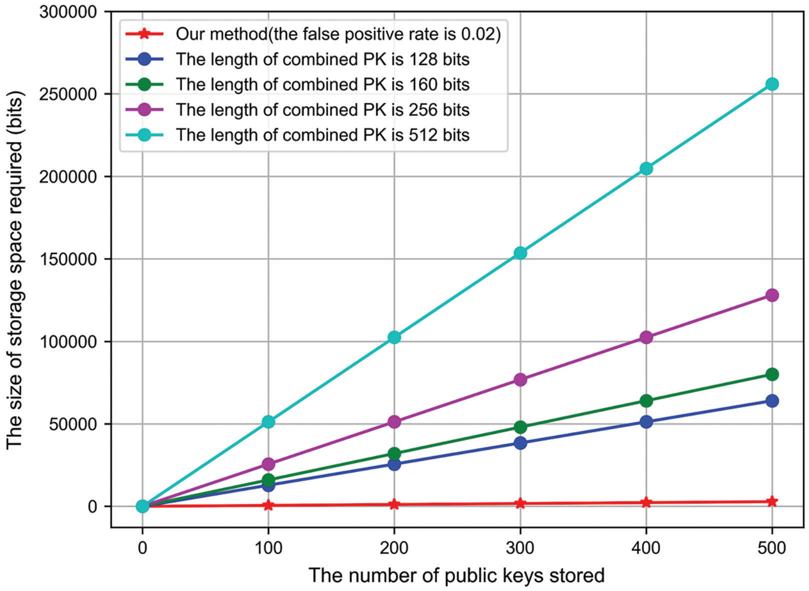

Figure 5: Comparison of the storage space occupied (the false-positive rate of our method is 0.02)

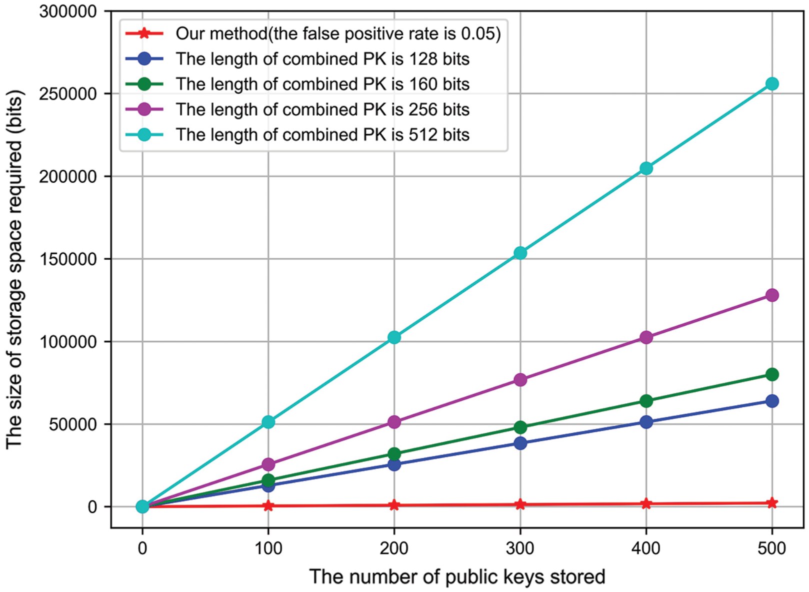

Figure 6: Comparison of the storage space occupied (the false-positive rate of our method is 0.05)

Table 3: Feature comparison

	[21,22]	[36]	Our method
Seed key selection space	The selection of the limited seed key is simple and fast, but the selection space is small and not suitable for large-scale applications	The large selection of key factors makes it easy to realize large-scale applications	The large selection of key factors makes it easy to realize large-scale applications

Table 3 (continued).

	[21,22]	[36]	Our method
Deduplication speed	No need for database support and fast deduplication	Relatively slow	Relatively slow
Key storage	No need to store the combined public key	The database needs to store the legal combined public key; and the required storage space is large, which is related to the number and length of the public key	The storage space size is unrelated to the combined public key length, and the required storage space is small
Query time complexity	–	$\mathrm{O}\left(log_x^n\right) \sim \mathrm{O}(n)$	$\mathrm{O}(1)$
Applicability	When the number of devices or users is small	Large-scale application	Large-scale application, few storage resources required, and suitable for a power IoT environment with a large number of terminal devices

7 Conclusion and Future Work

This paper proposes an IoT device identity authentication scheme based on CPK and PUFs. The proposed scheme improves the security of the identity authentication of power IoT terminal devices and reduces the resource consumption of terminal devices. A method for generating a unique device identification combining CPK and PUF physical fingerprints to achieve dual identity authentication between power IoT terminal devices and servers is designed, which improves the security of power IoT device authentication. In addition, the CPK identification key generation algorithm is improved, and the possible key collision problem is solved, making this solution more suitable for power IoT environments with large numbers of terminal devices. Our future work will provide concrete and rigorous security proof of our scheme and improve the anti-collusion attack capability.

Acknowledgement: We thank the anonymous reviewers and editors for their very constructive comments.

Funding Statement: This work is supported by the Science and Technology Project of State Grid Jiangsu Electric Power Co., Ltd. under Grant No. J2020068.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. Nguyen, E. L. Lydia, M. Elhoseny, I. V. Pustokhina, D. A. Pustokhin *et al.,* “Privacy preserving blockchain technique to achieve secure and reliable sharing of IoT data,” *Computers, Materials & Continua*, vol. 65, no. 1, pp. 87–107, 2020.

[2] T. Poomagal, G. A. S. Kumar and D. Mehta, “Multi level key exchange and encryption protocol for Internet of Things (IoT),” *Computer Systems Science and Engineering*, vol. 35, no. 1, pp. 51–63, 2020.

[3] D. Kim, H. Kim and J. Kwak, "Secure sharing scheme of sensitive data in the precision medicine system," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1527–1553, 2020.

[4] D. Díaz-Sánchez, A. Marín-Lopez, F. A. Mendoza, P. A. Cabarcos and R. S. Sherratt, "TLS/PKI challenges and certificate pinning techniques for IoT and M2M secure communications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3502–3531, 2019.

[5] C. Ge, L. Fang and J. Xia, "Key-private identity-based proxy re-encryption," *Computers, Materials & Continua*, vol. 63, no. 2, pp. 633–647, 2020.

[6] Y. Xie, F. Xu, X. Li, S. Zhang, X. Zhang *et al.,* "EIAS: An efficient identity-based aggregate signature scheme for WSNs against coalition attack," *Computers, Materials & Continua*, vol. 59, no. 3, pp. 903–924, 2019.

[7] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the Theory and Application of Cryptographic Techniques*. Berlin, Heidelberg, pp. 47–53, 1984.

[8] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *IEEE Trans Wireless Commun*, vol. 32, no. 3, pp. 213–229, 2003.

[9] X. H. Nan, "CPK algorithm and identity authentication," *Information Security and Communications Privacy*, vol. 000, no. 009, pp. 12–16, 2006.

[10] X. Fang, G. Yang and Y. Wu, "Research on the Underlying Method of Elliptic Curve Cryptography," in *Proc. ICISCE*, Changsha, pp. 639–643, 2017.

[11] R. Pappu, B. Recht, J. Taylor and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[12] M. Kang, S. K. Gonugondla, A. Patil and N. R. Shanbhag, "A multi-functional in-memory inference processor using a standard 6T SRAM array," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 2, pp. 642–655, 2018.

[13] X. H. Nan, H. P. Chen and Z. Chen, "Combined Public Key (CPK) Cryptosystem Standard (v5.0)," *Computer Security*, vol. 10, pp. 1–2, 2010.

[14] X. H. Nana and H. P. Chen, "Combined Public Key (CPK) Cryptosystem Standard (v3.0)," *Computer Security*, vol. 11, pp. 1–2, 2009.

[15] X. H. Nana and H. P. Chen, "Combined Public Key (CPK) Cryptosystem Standard (v2.1)," *Computer Security*, vol. 9, pp. 1–2, 2008.

[16] P. Gope, J. Lee and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.

[17] Y. Cui, Y. Yao and G. N. Xu, "Research of ubiquitous power Internet of Things security authentication method based on CPK and RIFD," *Proc. ITNEC*, vol. 1, pp. 1519–1523, 2020.

[18] G. Xu, Y. Ren, G. Zhang, B. Liu, X. Li *et al.,* "HyCPK: Securing Identity Authentication in Ubiquitous Computing," in *Proc. UIC-ATC-ScalCom*, Beijing, pp. 239–246, 2015.

[19] X. T. Ma, W. P. Ma and X. X. Liu, "A cross domain authentication scheme based on blockchain technology," *Acta Electronica Sinica*, vol. 46, no. 11, pp. 2571–2579, 2018.

[20] G. H. Wang, M. Wang and W. U. Duo, "Analysis of the CPK random collision probability," *Information Security & Communication Privacy*, vol. 11, pp.87–88, 2008.

[21] X. Ma, X. Long and X. Fan, "Construction and selection scheme of seeded-key database of collision-free CPK," *Computer Engineering & Applications*, vol. 48, no. 27, pp.99–104, 2012.

[22] T. Li, H. Y. Zhang, J. Yang and D. Yu, "Optimized construction scheme of seeded-key matrices of collision-free combined public key," *Journal of Computer Applications*, vol. 35, no. 1, pp. 83–87, 2015.

[23] B. Chatterjee, D. Das, S. Maity and S. Sen, "RF-PUF: Enhancing IoT Security Through Authentication of Wireless Nodes Using In-Situ Machine Learning," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 388–398, 2019.

[24] H. Boyapally, P. Mathew, S. Patranabis, U. Chatterjee, U. Agarwal *et al.,* "Safe is the new Smart: PUF-based authentication for load modification-resistant smart meters," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[25] U. Chatterjee, V. Govindan, R. Sadhukhan, M. Debdeep and S. C. Rajat, "Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, pp. 424–437, 2019.

[26] M. H. Ameri, M. Delavar and J. Mohajeri, "Provably secure and efficient PUF-based broadcast authentication schemes for smart grid applications," *International Journal of Communication Systems*, vol. 32, no. 8, pp. 3935.1–3935.13935, 19, 2019.

[27] W. Liu, H. Q. He and W. Y. Dong, "Provable secure lightweight mutual authentication: MPUF-HB," *Journal of Chinese Computer Systems*, vol. 38, no. 11, pp. 2454–2457, 2017.

[28] V. P. Yanambaka, S. P. Mohanty, E. Kougianos and D. Puthal, "PMsec: Physical unclonable function-based robust and lightweight authentication in the Internet of Medical Things," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 388–397, 2019.

[29] U. Kocabas, A. Peter, S. Katzenbeisser and A. R. Sadeghi, "Converse PUF-based authentication," in *Proc. TRUST*, pp. 142–158, 2012.

[30] B. Kim, S. Yoon, Y. Kang and D. Choi, "PUF based IoT Device Authentication Scheme," in *Proc. ICTC*, pp. 1460–1462, 2019.

[31] E. Avaroğlu, "The implementation of ring oscillator based PUF designs in Field Programmable Gate Arrays using of different challenge," *Physica A: Statistical Mechanics and its Applications*, vol. 546, 2020. [Online]. http://doi.org/10.1016/j.physa.2020.124291.

[32] S. S. Zalivaka, A. A. Ivaniuk and C. H. Chang, "Reliable and modeling attack resistant authentication of arbiter PUF in FPGA implementation with trinary quadruple response," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1109–1123, 2019.

[33] A. Miller, Y. Shifman, Y. Weizman, O. Keren and J. Shor, "A Highly Reliable SRAM PUF with a Capacitive Preselection Mechanism and pre-ECC BER of 7.4 E-10," in *Proc. CICC*, pp. 1–4, 2019.

[34] P. Urien, "Innovative ATMEGA8 Microcontroler Static Authentication Based on SRAM PUF," in *Proc. CCNC*, pp. 1–2, 2020.

[35] C. M. Cheng, K. Kodera and A. Miyaji, "Differences among summation polynomials over various forms of elliptic curves, IEICE transcation on fundamentals of electronics," *Communications and Computer Sciences*, vol. 102, no. 9, pp. 1061–1071, 2019.

[36] P. C. Li, "Research and implementation on mobile Internet identity authentication based on improved combined public key," M.S. thesis, Nanjing University of Posts and Telecommunications, China, 2016.