International Journal of Power and Energy Systems, Vol. 46, No. 10, 2026
An Open Access Paper

A LIGHTWEIGHT TWO-FACTOR AUTHENTICATION SCHEME FOR POWER INTERNET OF THINGS

Xin Liang,* Shifeng Ou,* and Bin Wang*

Abstract

This study introduces a two-factor authentication mechanism for DTUs in power grids. The design leverages the IMEI of the communication module as a static device-bound factor and an SMS-based one-time verification code as a dynamic factor to enhance security under limited computational and storage resources. In our scheme, mutual authentication and session key agreement are realized through lightweight elliptic curve cryptography, combined with efficient hash and HMAC operations to reduce computational overhead. Meanwhile, no additional dedicated security hardware, persistent server-side verifier table, or time synchronization is required. We evaluated the computational and communication overhead under 160-bit elliptic curve parameters and the results demonstrate that our scheme remains within practical limits for resource-constrained DTU deployments.

Key Words

Authentication; international mobile equipment identity; power internet of things; distribution terminal units; network security

1. Introduction

The Internet of Things (IoT) is widely used to connect sensing devices, embedded terminals, communication modules, and cloud platforms. IoT objects range from wearable devices and vehicles to specialized equipment used in industrial settings, healthcare, agriculture, and other domains, enabling devices to communicate, share information, and perform tasks with limited human intervention [1]. Wireless sensor networks (WSNs), as a typical resource-constrained sensing infrastructure, support distributed data collection and status monitoring. Existing WSN studies show that time synchronization, self-organization, and energy efficiency are essential for sensing services [2]. In healthcare, wireless sensor nodes can collect and exchange health-related data for real-time patient monitoring [3]. Similar sensing and communication capabilities are also used in industrial control and smart-grid scenarios, including the power Internet of Things (PIoT), for equipment monitoring, data acquisition, remote control, fault detection, and so on.

The deployments of IoT and WSN also enlarge the attack surface. Thus, comprehensive IoT security frameworks emphasize that device protection, secure communication, access control, and authentication should be jointly considered when safeguarding connected IoT systems [4]. IoT authentication is essential for establishing trust, preventing unauthorized access, and safeguarding sensitive data in IoT ecosystems [1]. Existing studies show that authentication mechanisms must consider not only mutual authentication and key agreement, but also communication cost, storage overhead, scalability, and resistance to public-key, impersonation, replay, and node-cloning attacks [1]-[2]. IoT authentication schemes can be grouped into three technical directions: static identity-based methods, such as device-bound or PUF-bound identities, dynamic credential-based approaches, e.g., one-time passwords (OTPs), and hybrid constructions that combine both [5]-[6]. In 2023, Kim et al. [7] examined authentication threats in smart grids, and in the literature, they pointed out that the pure static mechanism is not enough to provide sufficient protection against active attacks, especially when facing man-in-the-middle interception and device cloning. Shamsoshoara et al.'s survey [8] for PUF-based IoT security solutions also supports this observation. They noted that environmental variations—such as temperature and voltage fluctuations—can affect reliability in edge deployments.

Many lightweight authentication protocols have been proposed for IoT environments. Ei-Hajj et al. [9] observed that certain designs focus primarily on reducing communication overhead without tightly integrating device-bound identity attributes. Such schemes provide limited resistance to device counterfeiting. Early password-based two-

* Guangxi Power Grid Co., Ltd. Electric Power Research Institute, Nanning, Guangxi, 530023, China; e-mail: liang_x.ggg@gx.csg.cn, ouqi012@163.com, wang_b.sy@gx.csg.cn
Corresponding author: Shifeng Ou

(DOI: 10.2316/J.2026.203-0711)

factor mechanisms [10]-[12] encounter challenges related to credential update limitations as well as high computation cost, and many of them used secret keys with lengths of thousands of bits. Trivedi and Patel [13] introduced the advantage of dynamic credentials, but their scheme requires increased computation and strict clock synchronization. Thus, Mathi and Dharuman [14] proposed a layered verification structure to mitigate desynchronization attacks. Unfortunately, this scheme increases signaling complexity and protocol overhead because an additional secondary verification process is incorporated.

Hardware-assisted approaches have also been investigated in our researches. Both Guo et al. [15] and Xie et al. [16] utilized PUF characteristics for lightweight mutual authentication. Guo et al.'s [15] scheme assumes the availability of trusted fog infrastructure and Xie et al.'s [16] scheme employed multi-PUF binding strategies. One of the shortages of PUF based scheme is quite dependent on hardware integrity, which make it a burden on finance.

In recent years, blockchain-supported authentication frameworks have attracted much attention. Schemes in [17, 18] improve tamper resistance and auditability for the decentralized architecture properties of blockchain. Wijesundara et al. [19] integrated decentralized identifiers (DIDs) and verifiable credentials (VCs) into firmware authentication, enhancing revocation transparency. But, as we know, blockchain may introduce additional communication and storage costs and it may be unacceptable for bandwidth-limited scenarios.

Although these studies provide useful background for lightweight IoT authentication, the present work focuses on Distribution Terminal Unit (DTU) authentication in the power Internet of Things (PIoT). A DTU is an important terminal device used in power grids and plays a significant role in data acquisition (telemetry and teleindication), remote control (telecontrol) and the communication process [20]. DTUs are commonly installed in roadside cabinets, pole-mounted enclosures, or other open scenarios, which increases exposure to physical tampering, device replacement, and unauthorized access attempts [21]-[23]. On the other hand, DTUs generally have limited processing capability and restricted storage resources, and facing stricter operational requirements than general sensing nodes, including long service lifetimes, difficult clock synchronization, and high reliability requirements for grid control, making heavyweight cryptographic mechanisms unsuitable for practical deployment. Maintaining long-term key or verifier tables at the server side also introduces additional security risks, since table leakage may directly compromise device credentials and increase management complexity at scale. Moreover, strict clock synchronization is difficult to guarantee in distributed power IoT scenarios, especially under unstable communication conditions. A single static identity is also insufficient, as it can be copied or cloned once exposed. Therefore, a practical PIoT-oriented authentication mechanism should reduce computation and communication overhead, avoid server-side verifier-table risks, remove strict time-synchronization dependence, and strengthen device identity authentication against cloning and impersonation. To address these requirements, this paper proposes a lightweight two-factor authentication scheme for DTU access in PIoT environments. Different from existing ECC-based lightweight authentication protocols that mainly rely on pre-shared long-term secrets, verifier tables, strict timestamp synchronization, PUF hardware, fognodes, or blockchain infrastructure, the proposed scheme combines a device-bound IMEI (International Mobile Equipment Identity) with an SMS-based one-time code to provide both static device binding and session-level freshness. Mutual authentication and session key establishment are completed within a single ECC-based protocol flow, while avoiding server-side verifier tables, additional trusted hardware, and strict clock synchronization. Therefore, the scheme is more suitable for practical PIoT deployment scenarios where DTUs are widely distributed, physically exposed, resource-constrained, and difficult to maintain frequently.

2. Scheme Design

In this section, we propose a lightweight DTU authentication scheme based on USIM cards with two authentication factors. The IMEI serves as a device-bound static factor, which binds the authentication process to a registered DTU and prevents unauthorized terminals from accessing the PIoT system. The SMS verification code serves as a dynamic factor, which provides session freshness and helps resist replay attacks. Their combination avoids the weakness of relying on a single IMEI, which may be copied, or a single SMS code, which may be intercepted, thus improving the practical robustness. This scheme consists of three phases: the initialization phase, the registration phase, and the authentication phase.

2.1 Symbol Description

The notations used in this paper are listed in Table 1.

Table 1
Notations

Notation	Meaning
S	The server
s	Master key of the Server
A	A client, i.e., a DTU (Distribution Terminal Unit)
ID	Identifier of a client
IMEI	The IMEI code of a communication module
SMS	Short Message Service (SMS) Code
q	A large prime number
$\mathbb{F}_q$	A finite field
$\mathcal{E}(\mathbb{F}_q)$	An elliptic curve
O	The infinite point of $\mathcal{E}(\mathbb{F}_q)$
G	A base point of $\mathcal{E}(\mathbb{F}_q)$
$R, r, r', \hat{r}$	Large random number
$\|$	Concatenation
$\oplus$	Exclusive OR
HMAC$(\cdot)$	A function for computing Hash Message Authentication Code
$h(\cdot)$	A collision-free one-way hash function
$X \Rightarrow Y : M$	Message sent from left to right via a secure channel
$X \rightarrow Y : M$	Message sent from left to right via an open channel
$x(G)$	The x-coordinate of ECC point G
KDF$(\cdot)$	Key derivation function
k	Session key

2.2 Initialization

During the initialization phase, the server administrator first selects an appropriate elliptic curve defined over a finite field, along with a base point G that generates a cyclic subgroup of prime order q. At the same time, a secure one-way hash function h(•) must be chosen. Then, a long master key s is generated and kept strictly confidential. It should be noted that all these parameters remain fixed within a protocol version. Parameter updates require a controlled re-registration procedure.

2.3 Registration

During the registration phase, the client selects an appropriate identifier ID and submits it along with its own IMEI number through a secure channel to the server. The detailed registration steps are shown in Figure 1, as described below:

(1) Client A generates a random number as its identifier ID and another random number $\hat{r}$ as a security parameter.
(2) A ⇒ S: ID, h(IMEI $||\hat{r}$). The client A calculates h(IMEI$||\hat{r}$) and sends it along with ID through the secure channel to the server S.
(3) After receiving the message, the server S calculates T=(HMAC$_s$ (ID)-h(IMEI$||\hat{r}$))·G.
(4) $S \Rightarrow A : T, q, G, E(F_q), h(\cdot)$. The server S returns T along with all public parameters (including $q, G, E(F_q)$ and $h(\cdot)$) through the same secure channel to the client A.
(5) Client A stores ID, $\hat{r}$ and T in the USIM card. These values will be permanently fixed after registration and cannot be modified anymore.

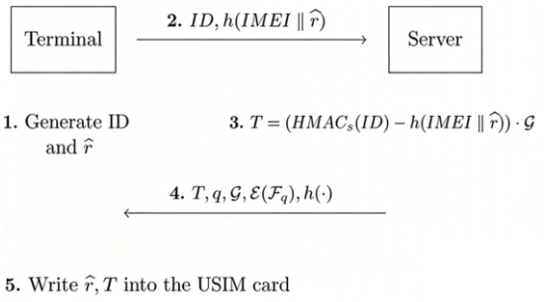


Figure 1. Registration Procedure

2.4 Authentication

During the authentication phase, the client and the server communicate through an open channel. The specific steps of this phase are shown in Figure 2 and are described as follows:

(1) Client A generates a random number r and calculates $r \cdot \mathcal{G}$.
(2) $A \rightarrow S : ID, r \cdot \mathcal{G}$. Client A sends ID and $r \cdot \mathcal{G}$ to Server S.
(3) After receiving the message from the client, the server S calculates T′, then generates a random number R, and computes $R \cdot \mathcal{G}$. Here, the calculation formula for T′ is as follows:

$$\begin{aligned} T' &= r \cdot \mathcal{G} + HMAC_s(ID) \cdot \mathcal{G} \\ &= (r + HMAC_s(ID)) \cdot \mathcal{G} \end{aligned}$$

Thus, the formula (1) is obtained:

$$T' = (r + HMAC_s(ID)) \cdot \mathcal{G} \quad (1)$$

Thereafter, the server S computes Auth $_S$ = $h\left(T' \| R \cdot \mathcal{G} \| ID \| r \cdot \mathcal{G}\right)$.
(4) $S \rightarrow A : Auth_S, R \cdot \mathcal{G}$. The server S sends $Auth_S$ and $R \cdot \mathcal{G}$ to Client A.
(5) S → A : SMS. The server S generates a random number (denoted as SMS) and transmits it to client A via the SMS (Short Message Service) channel.
(6) After receiving the response data, the client A reads the IMEI of communication module, and retrieves r̂ and T from the USIM card, and calculates T″ as follows:

$$\begin{aligned} T'' &= r \cdot \mathcal{G} + h(IMEI \| \hat{r}) \cdot \mathcal{G} + T \\ &= r \cdot \mathcal{G} + h(IMEI \| \hat{r}) \cdot \mathcal{G} \\ &+ (HMAC_s(ID) - h(IMEI \| \hat{r})) \cdot \mathcal{G} \\ &= r \cdot \mathcal{G} + HMAC_s(ID) \cdot \mathcal{G} \\ &= (r + HMAC_s(ID)) \cdot \mathcal{G} \end{aligned}$$

Thus, the formula (2) is obtained:

$$T'' = (r + HMAC_s(ID)) \cdot G \quad (2)$$

Thereafter, the client computes $Auth'_S$ = $h\left(T'' \parallel R \cdot \mathcal{G} \parallel ID \parallel r \cdot \mathcal{G}\right)$.
Subsequently, the client verifies the legitimacy of the server: if $Auth'_S = Auth_S$, the server verification is successful; otherwise, the connection is rejected.
Finally, the client calculates the verification tag: $V = h\left(R \cdot \mathcal{G} \| T'' \| SMS\right)$. At the same time, the client can calculate $k = KDF(x(r \cdot R \cdot \mathcal{G}) \| ID \| r \cdot \mathcal{G} \| R \cdot \mathcal{G})$, which is used as the session key, and computes $MAC_A = HMAC_k$ (”$'a_ok''\|ID\|r \cdot \mathcal{G}\|R \cdot \mathcal{G}\|V$)
(7) A → S : MAC_A, V. The client A sends MAC_A and V to Server S.
(8) After receiving V , the server S first calculates $k = KDF(x(R \cdot r \cdot \mathcal{G}) \| ID \| r \cdot \mathcal{G} \| R \cdot \mathcal{G})$ and $MAC'_A = HMAC_k$(” $a_ok'' \| ID \| r \cdot \mathcal{G} \| R \cdot \mathcal{G} \| V$), then checks whether $MAC'_A = MAC_A$ holds. If it does, the server continues tocalculate $V' = h\left(R \cdot \mathcal{G} \| T' \| SMS\right)$. Only if both $MAC'_A = MAC_A$ and $V = V'$, the login request is accepted and the server S computes $MAC_S = HMAC_k$(”s_ok” $\|ID\|r \cdot \mathcal{G}\|R \cdot \mathcal{G}\|V$); otherwise, the authentication is rejected.
(9) S → A : MAC_S. The server S returns MAC_S to the client Aafter accepting the login request.
(10) The client A calculates $MAC'_S = HMAC_k$(” $_ok'' \| ID \| r \cdot \mathcal{G} \| R \cdot \mathcal{G} \| V$), and checks whether $MAC'_S = MAC_S$ holds, and accepts the session if it does.

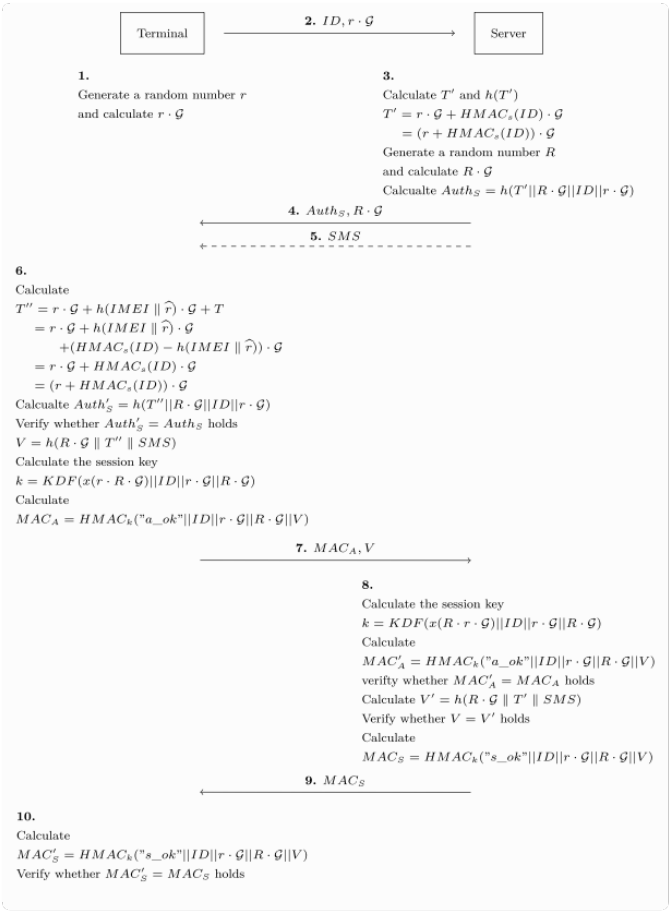


Figure 2. Authentication Procedure

3. Functional, Security, and Performance Analysis

3.1 Functional Analysis

3.1.1 Two-way Authentication

The proposed scheme realizes two-way authentication between the DTU client and the authentication server in the authentication stage. The client confirms the legitimacy of the server by verifying $Auth'_S = Auth_S$ in step 6; correspondingly, the server authenticates the client's identity by verifying both $MAC'_A = MAC_A$ and V = V ' in Step 8. Since $T' = r \cdot \mathcal{G} + HMAC_s(ID) \cdot \mathcal{G}$ can only be calculated by the legitimate server holding the master key s , an adversary cannot compute a valid T ' without knowing the server secret to construct the answer $Auth_S$. On the other hand, since V can only be generated by the legitimate client holding valid T and receiving fresh verification code SMS , successful verification of V allows the server to authenticate the DTU client.

3.1.2 Session Key Agreement

During the authentication stage of this scheme, the client calculates $k = KDF(x(r \cdot R \cdot \mathcal{G})\|ID\|r \cdot \mathcal{G}\| \ R \cdot \mathcal{G})$ in step 6, and the server calculates $k = KDF(x(R \cdot r \cdot \mathcal{G})\|ID\|r \cdot \mathcal{G}\|R \cdot \mathcal{G})$ in step 8. Because elliptic-curve scalar multiplication is commutative with respect to scalar multiplication order, the results of $r \cdot R \cdot \mathcal{G}$ and $R \cdot r \cdot \mathcal{G}$ are the same. Therefore, the value k can be safely used as the session key.

3.1.3 Immunity to Clock Synchronization Issues

This scheme uses random numbers (nonce) and one-time SMS verification codes instead of the timestamp mechanism to resist replay attacks, so there is no need for clock synchronization between the client and the server.

3.1.4 No Verification Table Required

In many table-based authentication schemes, the server needs to store verifier strings or temporary authentication values for each client with additional storage overhead. Once the table is inconsistent or lost, legitimate DTUs may fail to authenticate; once the table is leaked, an adversary may obtain reusable verifier values and launch impersonation or replay attacks.

The proposed scheme avoids maintaining a server-side verification table to store verifier strings for every DTU. This reduces storage cost, leakage risk, and long-term maintenance workload. This design fits large-scale DTU access in PIoT, where terminals may remain deployed for long periods and are difficult to maintain frequently. Even if the server database is exposed, it does not directly reveal reusable verification strings that can be replayed to pass authentication.

3.2 Security Analysis

3.2.1 Resistance to Verification String Theft

In the proposed design, the server does not maintain a persistent verification table storing the parameter T. Consequently, compromise of server-side storage does not directly expose client verification strings. The only long-term credential associated with the client is stored within the USIM card.

The USIM environment provides hardware-level access control and tamper-resistance mechanisms. Under the assumed threat model, extraction of T requires physical compromise of the USIM module. Even in the event that T is disclosed, authentication cannot be completed without the dynamically generated SMS verification code. Therefore, possession of the static parameter alone is insufficient to impersonate a legitimate client.

3.2.2 Resistance to Identity Forgery

In our scheme, an identity forgery attempt requires constructing a valid login message in the form $V = h\left(R \cdot \mathcal{G} \parallel T'' \parallel SMS\right)$ without access to the legitimate client's secret information. This requires the adversary to obtain both the derived parameter T″ and the SMS verification code. In the authentication phase, T″ depends on USIM-stored data and fresh random number r, and SMS is transmitted through an independent communication channel. Simultaneously compromising both factors

is non-trivial, thereby limiting the feasibility of message forgery.

3.2.3 Resistance to Server Impersonation

Impersonating the server requires an adversary to generate a valid response value $Auth_S$ that contains T' in Step 3. The master key s, which is known only to the legitimate server, is required to calculate the parameter $T' = r \cdot \mathcal{G} + HMAC_s(ID) \cdot \mathcal{G}$. Without knowledge of s, computation of a consistent T′ is computationally infeasible for the adversary.

3.2.4 Resistance to Replay

In the authentication phase, the DTU selects a fresh random number r, and the server selects a fresh random number R for the current session. Meanwhile, the SMS verification code SMS is generated dynamically and is valid only for the current authentication process. The verification tag is computed as $V = h(R \cdot \mathcal{G} \| T'' \| SMS)$ where $R \cdot \mathcal{G}, T''$, and SMS are all associated with the current session. Therefore, an old message containing previous values such as $r \cdot G, R \cdot G$, or V cannot pass verification in a new session. Once either the client-side random value, the server-side random value, or the SMS code changes, the corresponding verification tag becomes invalid. As a result, replayed authentication messages cannot be accepted even without using timestamps or requiring strict clock synchronization between the DTU and the server.

3.2.5 Resistance to SIM-Swap Attacks and SMS Interception

A SIM-swap attack refers to the case where an adversary uses copied SIM cards for the purpose of taking over a mobile service subscription and thereby accessing sensitive personal and financial data [24]. In an SMS interception attack, an adversary may eavesdrop on, forward, or obtain the one-time verification code transmitted through the SMS channel.

Possession of the SMS verification code alone is insufficient to complete authentication. The DTU identity is bound to the IMEI of the communication module and the authentication parameters stored in the legitimate USIM card. During authentication, the server verifies not only the dynamic SMS factor but also the device-bound identity and the protocol values generated from the registered authentication parameters. Therefore, even if an adversary succeeds in receiving the SMS verification code through SIM swapping or interception, the adversary still cannot pass authentication without the legitimate USIM-stored parameters and the corresponding IMEI binding.

3.2.6 Resistance to Physical Device Compromise

In PIoT, physical access by an adversary should be considered because DTUs are often deployed in open environments. If an attacker only obtains temporary physical access to the DTU, he must extract the USIM-stored authentication parameters, reproduce the IMEI-bound identity, and generate valid ECC protocol values within the authentication session. However, if the attacker can completely clone the USIM card, extract all internal authentication parameters, and reproduce the corresponding device identity, this should be regarded as a strong physical compromise beyond the normal lightweight authentication threat model.

To reduce this risk in practical PIoT deployment, DTUs should adopt tamper-resistant packaging, USIM PIN protection, secure storage for sensitive parameters, and access-control mechanisms for maintenance ports. The server side should also support abnormal IMEI re-registration detection, repeated authentication failure monitoring, and device replacement auditing. These operational measures help limit the impact of physical compromise of the proposed scheme.

3.3 Formal Security Verification

In this section, we analyze the security of the proposed scheme in a formal way by using BAN, which is widely used in authentication protocol analysis. The notations of BAN are specified in Table 2.

Table 2
BAN Notations

Notation	Meaning
P, Q	Network agents
X	Proposition
$P \mid\equiv X$	P believes X is true
$P \triangleleft X$	P received X
$A \mid\sim X$	P has previously sent X
$\#(X)$	X is fresh
$P \overset{k}{\leftrightarrow} Q$	P and Q share the session key k
P $\mid\Rightarrow$ X	P dominates and control X

From literature [25] we have the following BAN rules:

(1) Freshness-extension rule:

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}.$$

(2) Nonce-verification rule:

$$\frac{P\mid \equiv \#(X), P\mid \equiv Q \mid\sim X}{P\mid \equiv Q\mid \equiv X}.$$

Classical BAN logic does not directly model Diffie-Hellman and HMAC algebra. Therefore, in addition to the standard BAN rules in [25], we adopt the following two assumptions for the analysis.

(1) Diffie-Hellman key-agreement assumption

$$\frac{P \mid\equiv x \text{ is secret to } P, P\mid \equiv Q\mid \sim Y, k = KDF(x(Y),\ \text{other session messages }\ )}{P \mid\equiv P \leftrightarrow Q}.$$

(2) HMAC message-meaning assumption

$$\frac{P \mid\equiv P \overset{k}{\leftrightarrow} Q, P \triangleleft HMAC_k(X)}{P \mid\equiv Q \mid\sim X}.$$

3.3.1 Authentication Goals

To prove mutual authentication in the proposed scheme, the following four goals should be achieved.

(1) The client A believes that the session key k is established.

$$G1: \mathrm{A} \mid\equiv \mathrm{A} \overset{\mathrm{k}}{\leftrightarrow} \mathrm{S}.$$

(2) The server S believes that the session key k is established.

$$G2: \mathrm{S} \mid\equiv \mathrm{A} \overset{\mathrm{k}}{\leftrightarrow} \mathrm{S}.$$

(3) The client A believes that the server S believes the shared session key k is established.

$$G3: \mathrm{A} \mid\equiv \mathrm{S} \mid\equiv \mathrm{A} \overset{\mathrm{k}}{\leftrightarrow} \mathrm{S}.$$

(4) The server S believes that the client A believes the shared session key k is established.

$$G4: \mathrm{S} \mid\equiv \mathrm{A} \mid\equiv \mathrm{A} \overset{\mathrm{k}}{\leftrightarrow} \mathrm{S}.$$

3.3.2 Protocol Message Idealization

The protocol messages are idealized according to BAN logic, as follows:

$$M1: A \rightarrow S: \langle ID, r \cdot \mathcal{G} \rangle$$
$$M2: S \rightarrow A: \langle \text{ Auth }_S, R \cdot \mathcal{G} \rangle$$

where $Auth_S = h\left(T' \| R \cdot \mathcal{G} \| ID \| r \cdot \mathcal{G}\right)$ and $T' = r \cdot \mathcal{G} + HMAC_S(ID) \cdot \mathcal{G}$.

$$M3: S \rightarrow A: SMS.$$

$$M4: A \rightarrow S: \langle MAC_A, V \rangle,$$

where $V = h\left(R \cdot \mathcal{G} \| T'' \| SMS\right)$, $T'' = (r + HMAC_s(ID)) \cdot \mathcal{G}$, and $MAC_A = HMAC_k\left(''a_ok'' \| ID \| \ r \cdot \mathcal{G} \| R \cdot \mathcal{G} \| V\right)$.

$$M5: S \rightarrow A: MAC_S,$$

where $MAC_S = HMAC_k\left(\text{ }'' \ s_ok'' \| ID \| r \cdot \mathcal{G} \| R \cdot \mathcal{G} \| V\right)$.

From the registration phase, we know that $T = (HMAC_s(ID) - h(IMEI \| \hat{r})) \cdot \mathcal{G}$. Denote $C_{ID} = HMAC_S(ID) \cdot \mathcal{G}$. Therefore, $T' = \mathrm{r} \cdot \mathcal{G} + C_{ID} = T''$.

3.3.3 Initial Assumptions

Before conducting the BAN logic analysis, the following initial assumptions are defined.

The client and server believe that their own ephemeral values are fresh:

$$A1: A \mid\equiv \#(r),$$
$$A2: S \mid\equiv \#(R).$$

The parties believe that the SMS value is fresh and associated with the current authentication run:

$$A3: A \mid\equiv \#(SMS),$$
$$A4: S \mid\equiv \#(SMS).$$

The client and server believe that the registration-derived relation is available only to the legitimate protocol participants under the adopted threat model:

$$A5: A \mid\equiv A \overset{C_{ID}}{\longleftrightarrow} S,$$
$$A6: S \mid\equiv A \overset{C_{ID}}{\longleftrightarrow} S.$$

Each party believes that the other party controls its own ephemeral contribution:

$$A7: \mathrm{A} \mid\equiv \ \mathrm{S} \mid\Rightarrow \mathrm{R} \cdot \mathcal{G},$$
$$A8: S \mid\equiv A \mid\Rightarrow r \cdot \mathcal{G}.$$

The private scalars are assumed to remain secret:

$$A9: A \mid\equiv r \text{ is known only to } A,$$
$$A10: S \mid\equiv R \text{ is known only to } S.$$

The parties believe that only a holder of k can generate a valid $HMAC_k(\cdot)$:

A11: $A \mid\equiv HMAC_k(X)$can begenerated only by a holder of k,
A12: $S \mid\equiv HMAC_k(X)$can begenerated only by a holder of k.

3.3.4 BAN Logic Derivation

(1) Client authentication of the server
Based on $M2$, the client receives:

$$A \triangleleft \langle Auth_S, R \cdot \mathcal{G} \rangle.$$

The client reconstructs: $T'' = \mathrm{r} \cdot \mathcal{G} + C_{ID}$ and verifies Auth $_S = h\left(T' \| R \cdot \mathcal{G} \| ID \| r \cdot \mathcal{G}\right)$. The client believes that only S , who holds the master secret s, can generate a valid $Auth_s$ bound to C_{ID} and the current protocol values. The client infers:

$$A \mid\equiv S \mid\sim \langle ID, r \cdot \mathcal{G}, R \cdot \mathcal{G} \rangle.$$

From A1:

$$A \mid\equiv \#(r \cdot \mathcal{G}).$$

By the freshness-extension rule:

$$A \mid\equiv \#\langle ID, r \cdot \mathcal{G}, R \cdot \mathcal{G} \rangle.$$

According to the nonce-verification rule:

$$A \mid\equiv \ \mathrm{S} \mid\equiv \langle ID, r \cdot \mathcal{G}, R \cdot \mathcal{G} \rangle.$$

That is, the client A believes that $R \cdot \mathcal{G}$ was generated by the server for the current run. Using its secret r, the authenticated $R \cdot \mathcal{G}$, the client derives:

$$\mathrm{A} \mid\equiv \mathrm{A} \overset{\mathrm{k}}{\leftrightarrow} \mathrm{S}$$

Thus, G1 is achieved.

(2) Server authentication of the client
From M4, the server receives:

$$S \triangleleft \langle MAC_A, V \rangle .$$

The server first verifies: $V = h\left(R \cdot \mathcal{G} \,\|T'\|\, SMS\right)$. Because $T' = \mathrm{r} \cdot \mathcal{G} + C_{ID}$, a valid V demonstrates possession of the registration-derived client material and the fresh SMS value for the current $R \cdot \mathcal{G}$. The server may infer:

$$\mathrm{S}| \equiv A| \sim \langle ID, r \cdot \mathcal{G}, R \cdot \mathcal{G}, SMS, V \rangle.$$

From A2 and A4:

$$S \mid\equiv \#(R \cdot \mathcal{G}),$$
$$S \mid\equiv \#(SMS).$$

By the freshness extension:

$$S \mid\equiv \#\langle ID, r \cdot \mathcal{G}, R \cdot \mathcal{G}, SMS, V \rangle.$$

Applying the nonce-verification rule:

$$S| \equiv \ \mathrm{A}| \equiv \langle ID, r \cdot \mathcal{G}, R \cdot \mathcal{G}, SMS, V \rangle.$$

The server then derives k and verifies: $MAC_A = HMAC_k(\ "\ a_ok''\|ID\|r \cdot \mathcal{G}\|R \cdot \mathcal{G}\|V)$ and $V = h\left(R \cdot \mathcal{G} \,\|T''\|\, SMS\right)$. The successful verification of V associates the message with the legitimate registered client, while successful verification of MAC_A demonstrates possession of the key derived from $r \cdot R \cdot \mathcal{G}$. Under the "Diffie-Hellman key-agreement assumption", the server obtains:

$$\mathrm{S} \mid\equiv \mathrm{A} \overset{\mathrm{k}}{\leftrightarrow} \ \mathrm{S}.$$

Thus, G2 is achieved.
Because MAC_A is a current-session statement authenticated under k, the server also infers:

$$\mathrm{S}| \equiv \ \mathrm{A}| \equiv \mathrm{A} \overset{\mathrm{k}}{\leftrightarrow} \ \mathrm{S}.$$

Thus, G4 is achieved.
(3) Server-to client key confirmation
After verifying both MAC_A and V, the server sends $M5$:

$$S \rightarrow A : MAC_S.$$

The client receives:

$$A \triangleleft MAC_S.$$

Then the client verifies $MAC_S = HMAC_k(\ "\mathrm{s_ok}"\ \|ID\|r \cdot \mathcal{G}\|R \cdot \mathcal{G}\|V)$.
From G1, A11 and "HMAC message-meaning assumption", the client infers:

$$\mathrm{A}| \equiv \ \mathrm{S}| \sim \langle ID, r \cdot \mathcal{G}, R \cdot \mathcal{G}, V \rangle.$$

Because the authenticated statement contains the fresh current-session values $r \cdot \mathcal{G}$ and $R \cdot \mathcal{G}$, nonce verification gives:

$$\mathrm{A}| \equiv \ \mathrm{S}| \equiv \langle ID, r \cdot \mathcal{G}, R \cdot \mathcal{G}, V \rangle.$$

Successful verification of MAC_S demonstrates that the server derived and accepted the same session key. Therefore:

$$\mathrm{A}| \equiv \ \mathrm{S}| \equiv \mathrm{A} \overset{\mathrm{k}}{\leftrightarrow} \ \mathrm{S}.$$

Thus, G3 is achieved.

3.4 Deployment Considerations

DTUs may be installed in mountainous areas, roadside cabinets, pole-mounted enclosures, or other locations where cellular coverage is weak or unstable. In such environments, SMS delivery may be unstable, potentially increasing authentication latency or causing occasional authentication failures. In deployment, these risks can be mitigated by several operational mechanisms.

(1) Assign a reasonable validity window for the one-time verification code, so that normal network delay does not immediately invalidate the authentication process.
(2) Limit the number of re-transmission attempts to prevent abuse and repeated guessing.
(3) Use dedicated APN services or power private communication networks instead of ordinary public SMS delivery for the SMS channel.

3.5 Performance Analysis

The computational advantage of the proposed scheme mainly comes from the use of elliptic curve cryptography (ECC). Compared with traditional RSA-based or DH-based public-key mechanisms, ECC can provide the same security level with a much shorter key length. For example, a 160-bit ECC key is generally considered to provide a security strength roughly comparable to that of a 1024-bit RSA key. Therefore, ECC-based point multiplication requires smaller operands and introduces lower communication overhead than RSA modular exponentiation or traditional DH operations at the same security level.

The computational cost, storage requirements and communication overhead of the proposed scheme are analyzed below. Assuming that the system employs 160-bit elliptic curve cryptography (ECC), 160-bit hash function and 160-bit random numbers [26], and KDF is also implemented with a hash function. Since elliptic curve points are represented in the form of coordinate pairs, 320 bits are required. The performance indicators and communication overhead of the client and server sides in the registration stage and the authentication stage are evaluated separately below. The following symbols are used in the analysis: t_h : hash function or HMAC computation time; t_r : random number generation time; t_a : elliptic curve point addition time; t_m : elliptic curve point multiplication time.

3.5.1 Computational Overhead Analysis

(1) Registration Stage

Table 3
Performance

Stage	Client		Service term		Communication overhead
	Time	Space	Time	Space	
Registration Phase	$2t_r + t_h$	80B	$t_h + t_m$	0	220B
Authentication Phase	$t_r + 6t_h + 2t_a + 3t_m$	0	$t_r + 6t_h + t_a + 3t_m$	0	186B

Table 4
Performance Comparison

Scheme	Two-factor mechanism	ECC operation	No Verification Table	No clock synchronization required	Computational overhead	Communication overhead	Engineering Adaptability
Lin et al. [11]	√	×	√	×	Middle to high	Middle	Lower
Trivedi and Patel [13]	✓	✓	×	×	Low to medium	Low	Lower
Mathi and Dharum an [14]	✓	×	×	✓	Low to medium	Low	General
Guo et al. [15]	×	×	×	✓	Low	Low to medium	General
This proposal	✓	✓	✓	✓	Low	Low	High

In Step 1, the client generates two random numbers ID and r̂ and calculates $h(IEMI\|\hat{r})$. The total computational cost is: $t_r + t_h$.
On the server side, in Step 3, T is calculated. This process involves an HMAC operation and one point multiplication operation, with a total cost of: $t_h + t_m$.

(2) Authentication Phase
In Step 1, the client generates a random number r and calculates $r \cdot \mathcal{G}$ through a point multiplication operation. In Step 6, the client computesT", which involves one point multiplication and two point additions; then it calculates $Auth'_S$ and V with two hash, and uses one point multiplication and a KDF to obtain the session key k ; then it calculate MAC_A with an HMAC. In step 10, the client computes MAC'_S with an HMAC. Therefore, the total cost for the client is: $t_r + 6t_h + 2t_a + 3t_m$.
On the server side, in Step 3, T′ is calculated, including one point addition, one HMAC operation, and one point multiplication; then it computes $Auth_S$ with a hash. Then a random number R is generated, and $R \cdot \mathcal{G}$ is obtained through one point multiplication. In Step 8, the session key k is calculated through a point multiplication and a KDF; then MAC'_A is computed with an HMAC operation, and V ' is computed with a hash operation, and MAC_S is computed with an HMAC operation. Therefore, the total cost for the server side is: $t_r + 6t_h + t_a + 3t_m$.

3.5.2 Storage Overhead Analysis

(1) Registration Stage
The client persistently stores ID, r̂, and T, requiring 160 + 160 + 320 = 640 bits = 80 bytes. The server does not maintain a persistent per-client verifier table; however, temporary session state and device-to-SMS routing information are outside the storage estimate.

(2) Authentication Phase
This stage does not involve the storage of permanent data.

3.5.3 Communication Overhead Analysis

(1) Registration Phase
The client sends ID and $h(IEMI\|\hat{r})$ to the server, and the server returns, $T, q, \mathcal{G}, \varepsilon(\mathcal{F}_q)$, and $h(\cdot)$ to the client, where $\varepsilon(\mathcal{F}_q)$ is determined by three scalar coefficients of equation $y^2 = x^3 + ax + b \bmod p$. Therefore, the total communication volume is: 160 + 160 + 320 + 160 + 320 + 3 ∗ 160 + 160 = 1760 bits = 220 bytes.

(2) Authentication Stage
In Step 2, the client sends ID and $r \cdot \mathcal{G}$, a total of 160 bits + 320 bits = 60 bytes.
In Step 4, the server sends $Authh_S$ and $R \cdot \mathcal{G}$, also 160 bits + 320 bits = 60 bytes.
In Step 5, the SMS code contains no more than 6 ANSI characters, approximately 6 bytes.
In Step 7, the client sends MAC_A and V, a total of 160 bits + 160 bits = 40 bytes.
In Step 9, the server sends MAC_S, occupying 20 bytes.
Therefore, the total communication overhead of this stage is: 60 + 60 + 6 + 40 + 20 = 186 bytes.
The comprehensive performance analysis results are shown in Table 3.

3.6 Performance Comparison Analysis

Table 4 summarizes the performance of related researches. Lin et al.'s scheme [11] requires the highest computational and communication overhead for it is based on DLP. Though Trivedi and Patel's scheme [13] uses ECC to lower the computational and communication overhead, it needs both verification table and clock synchronization. Mathi and Dharuman's scheme [14] achieves low computational overhead, but reduces the engineering applicability for introducing an additional secondary verification. Guo et al.'s scheme [15] similarly reduced its practicality to some ex-

tent for reliance on the credibility of fog nodes while clock synchronization is required.

Lastly, by using IMEI and SMS as the two authentication factors, the scheme presented in this work provides competitive computational and communication overhead while remaining the engineering applicability and being free of verification table and clock synchronization.

To evaluate the computational cost under practical conditions, experiments were conducted on an industrial control computer configured as listed in Table 5. The elliptic curve used is $E(\mathbb{F}_p)$, where p is a 160 -bit prime number, and the order q of the point group defined on $E(\mathbb{F}_p)$ is also a 160 -bit prime number. The curve parameter set adopted in the experiment is secp160r1 for comparative evaluation.

Table 5
Hardware and Software Specification

Hardware/Software	Specification
CPU	Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz
RAM	4GB DDR3
OS	Kali Linux
Library	fastecdsa(python)

Execution time was recorded separately for the registration and authentication stages. Each measurement was repeated 100 times, and the average values were reported and all experiments were implemented in Python using the fastecdsa library. The measured timing results are presented in Figure 3.

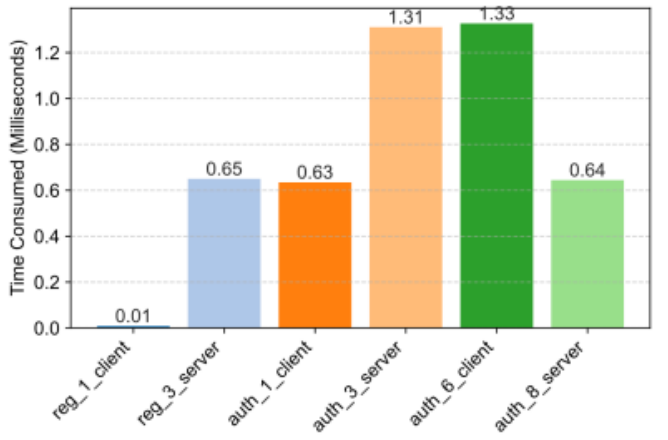


Figure 3. Time Consumed

4. Conclusion

This work describes a two-factor authentication mechanism designed for DTUs operating under constrained computational and storage conditions in power IoT environments. The scheme combines the IMEI and SMS-based one-time verification code to construct a dual-factor authentication process. Mutual authentication and session key establishment are achieved without introducing additional security hardware, persistent verification tables, or timestamp synchronization. Experiments were conducted with 160-bit elliptic curve parameters and the computational and communication overhead remains within the practical capacity of power grids terminal devices. Therefore, the proposed approach provides a feasible option for DTU authentication.

Funding

Research on Wireless Communication and Security Encryption Detection Technology for Distribution Terminals and Development of Corresponding Devices (Project No.:GXKJXM20240139).

References

[1] G. Zargar, H. Barati, and A. Barati, "Authentication methods in internet-of-things platform: A comprehensive review," *Majlesi Journal of Telecommunication Devices*, vol. 13, no. 4, pp. 239–258, 2024.

[2] L. Khajehzadeh, H. Barati, and A. Barati, "A lightweight authentication and authorization method in iot-based medical care," *Multimedia Tools and Applications*, vol. 84, no. 12, pp. 11137–11176, 2025.

[3] Z. G. Al-Mekhlafi, Z. M. Hanapi, and A. M. S. Saleh, "Firefly-inspired time synchronization mechanism for self-organizing energy-efficient wireless sensor networks: A survey," *IEEE Access*, vol. 7, pp. 115229–115248, 2019.

[4] Z. G. Al-Mekhlafi and S. A. Alfhaid, "Innovative security measures: A comprehensive framework for safeguarding the internet of things," in *AI-Driven: Social Media Analytics and Cybersecurity* (W. M. Yafooz and Y. Al-Gumaei, eds.), vol. 1180 of *Studies in Computational Intelligence*, pp. 175–185, Cham: Springer, 2025.

[5] J. S. Yalli, M. H. Hasan, L. T. Jung, and S. M. Al-Selwi, "Authentication schemes for internet of things (iot) networks: A systematic review and security assessment," *Internet of Things*, vol. 30, p. 101469, 2025.

[6] A. N. Alsheavi, A. Hawbani, W. Othman, X. Wang, G. Qaid, L. Zhao, and M. A. Al-Qaness, "Iot authentication protocols: Challenges, and comparative analysis," *ACM Computing Surveys*, vol. 57, no. 5, pp. 1–43, 2025.

[7] Y. Kim, S. Hakak, and A. Ghorbani, "Smart grid security: Attacks and defence techniques," *IET Smart Grid*, vol. 6, no. 2, pp. 103–123, 2023.

[8] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, "A survey on physical unclonable function (puf)-based security solutions for internet of things," *Computer Networks*, vol. 183, p. 107593, 2020.

[9] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (iot) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, 2019.

[10] M. Scott, "Authenticated id-based key exchange and remote login with simple token and pin number," 2002. Cryptology ePrint Archive.

[11] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[12] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic id-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.

[13] H. S. Trivedi and S. J. Patel, "Design of secure authentication protocol for dynamic user addition in distributed internet-of-things," *Computer Networks*, vol. 178, p. 107335, 2020.

[14] S. Mathi and L. Dharuman, "Prevention of desynchronization attack in 4g lte networks using double authentication scheme," *Procedia Computer Science*, vol. 89, pp. 170–179, 2016.

[15] Y. M. Guo, Z. F. Zhang, P. Xiong, and Y. J. Guo, "Puf-based lightweight authentication protocols for fog assisted iot," *Chinese Journal of Computers*, vol. 45, no. 7, pp. 1412–1430, 2022.

[16] F. Xie, K. Meng, W. Zhang, J. X. Xu, and Q. Y. Wang, "Research on identity generation and identity authentication mechanism based on multi-puf modules," *Computer Technology and Development*, vol. 34, no. 8, pp. 73–77, 2024.

[17] A. K. Al Hwaitat, M. A. Almaiah, A. Ali, S. Al-Otaibi, R. Shishakly, A. Lutfi, and M. Alrawad, "A new blockchain-based authentication framework for secure iot networks," *Electronics*, vol. 12, no. 17, p. 3618, 2023.

[18] W. H. Miao, J. X. Wang, and Z. H. Zheng, "Identity authentication scheme based on blockchain and multi factor combination," *Computer Simulation*, vol. 39, no. 5, pp. 402–408, 2022.

[19] W. M. A. B. Wijesundara, J. S. Lee, E. Aloupogianni, D. Tith, H. Suzuki, and T. Obi, "Didauth-iotfw: Decentralized firmware authentication for smart home iot devices using verifiable credentials," *Internet of Things*, vol. 34, p. 101788, 2025.

[20] Q. P. Song, H. L. Yang, Z. Li, N. Wang, L. P. Li, and Y. Y. Liu, "Integrated modeling of distribution terminals for distribution internet of things," *Modern Electric Power*, vol. 41, no. 1, pp. 1–9, 2024.

[21] J. Yuan and L. X. Wang, "Research and implementation of low cost dtu security protection scheme for distribution network," *Information Security and Communications Privacy*, no. 7, pp. 109–117, 2024.

[22] R. Yaben, N. Lundsgaard, J. August, and E. Vasilomanolakis, "Towards identifying neglected, obsolete, and abandoned iot and ot devices," in *8th Network Traffic Measurement and Analysis Conference*, (Dresden, Germany), IEEE, 2024.

[23] A. Srhir, T. Mazri, and M. Benbrahim, "Security in the iot: State-of-the-art, issues, solutions, and challenges," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023.

[24] M. Kim, J. Suh, and H. Kwon, "A study of the emerging trends in sim swapping crime and effective countermeasures," in *2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, (Danang, Vietnam), pp. 240–245, IEEE, 2022.

[25] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[26] D. X. Zhou, J. M. Zhao, F. Q. Zeng, X. Feng, and S. Li, "A method for generating and reconstructing power grid substation topology based on lstm lasso model," *International Journal of Power and Energy Systems*, vol. 45, no. 1, pp. 26–35, 2025.

Biographies



Xin Liang received a Bachelor of Engineering degree. He is currently an Engineer with Guangxi Power Grid Co., Ltd. Electric Power Research Institute. His research interests include power grid safety production management and the research and application of information security technologies.



Shifeng Ou received a Master of Engineering degree. He is currently a Senior Engineer with Guangxi Power Grid Co., Ltd. Electric Power Research Institute. His research interests include the research, promotion, and application of new distribution power technologies.



Bin Wang received a Master of Engineering degree. He is currently an Engineer with Guangxi Power Grid Co., Ltd. Electric Power Research Institute. His research interests include electrical measurement and thermal metering technologies.