Design and Implementation of an SM2-Based Security Authentication Scheme with the Key Agreement for Smart Grid Communications

Wei Li, Rui Li, Kehe Wu, Rui Cheng, Linping Su, Wenchao Cui
North China Electric Power University, Beijing, CO 102206 CN
Corresponding author: Rui Li (e-mail: 116227089@ncepu.cn)

This work was supported by the China State grid headquarters technology project "Research of new energy station network security protection key technology " under Grant 522722180007.

ABSTRACT Smart grids have a number of benefits; they provide a reliable and uninterruptible power supply and promote smart cities in a smart way. Multiple modes of communication are deployed in smart grids, including wired networks and wireless networks. More and more intelligent terminals are connected to the smart grid control center, such as RTU, charge pile control terminal, and centralized meter reading terminal etc. The communication between the smart grid control center (SC) and the smart grid terminals (STs) are susceptible to cyber-attacks due to a lack of inherent and effective security mechanisms. Some security communication schemes for smart grids have recently been proposed in the relevant literatures. However, published schemes have limited features. Given the security needs of smart grid communication networks and the inherent qualities of smart terminals, we proposed a novel authentication scheme that addresses cyber-attacks using a minimal number of computations. The proposed scheme is implemented based on mutual authentication as well as an improved SM2-based key agreement. Through detailed security analysis and experimental evaluation, this paper validates the security of the approach based on the lowest possible communication and computational cost.

INDEX TERMS Key agreement, mutual authentication, smart grids, SM2

I. INTRODUCTION

The smart grid is a platform with good development foreground that meets the needs of users in the twenty-first century using two-way high-speed communication network technologies and computer-based automation and distributed control. Current power grids are expanding with many advanced smart grid technology characteristics, such as advanced measurement and sensing technology, advanced decision support systems and advanced control methods [1]. The smart grid features intelligent transmission and distribution networks, and controls and optimizes the production and distribution of electricity through high-voltage and low-voltage networks. From power generators to energy storage systems and end-user consumers, the smart grid provides substantial advancements to the reliability, flexibility, efficiency, and load balancing/load adjustment of the electric system [2]. Features include the ability to self-heal, access to various forms of power generation, generation of an electricity market, and optimized and efficient operation of assets.

Several wired and wireless technologies are used to realize the communication network in the smart grid [3]. Various communication technologies, such as optical fiber communication, power line carrier communication, industrial Ethernet, private power communication networks, APON(APON, ATM Passive Optical Network), and wireless special networks are used in the smart grid communication network. The mixed-use of multiple communication modes is common practice in smart grid communication systems. However, the integration of communication networks with power systems exposes the smart grid to vulnerabilities and creates challenges in network safety [4]. The security challenges specifically include the potential for several types of cyber-attacks, such as replay attacks, man-in-the-middle (MITM) attacks, and impersonation attacks [2]. The security issues of the smart grid communication network affect the availability, reliability, and productivity of the power system.

As shown in Figure 1, power systems involve power generation, transmission, distribution and power consumption, comprising the entire smart grid communication network structure with the communication network. In this paper, we take into account general-purpose smart grid applications that have a smart grid terminal (ST)and a smart grid control center(SC).Here, the ST is near
the electrical devices and corresponds to a DTU, RTU, FTU, TTU, charge pile control terminal, centralized meter reading terminal and renewable energy power generation acquisition terminal. One of the example is the communication between RTU and the SC, which can monitor and measure sensors and equipment installed on remote sites [1]. Another example is the communication between the wind power generation acquisition terminal and the SC, collecting and sending the measurement and monitoring information of the components of the wind power generation unit to analyze the operation of the wind power station and report the power generation status of the generator set.

Communication between the SC and the ST may use a security scheme. The RSA algorithm was recommended for securing smart grid. ECC also has addressed various known security concerns, such as message integrity, non-authentication, and confidentiality of message information[5]. The elliptic curve cryptosystem was proposed in 1986, which is better than traditional cryptosystems (such as the RSA and DES cryptosystem) in terms of security and efficiency. It has become a mainstream algorithm of the public key algorithm. In December 2010, to satisfy the requirements of identity authentication in E-commerce and E-government, the Chinese Encryption Administration issued the SM2 elliptic curve public key algorithms, which included the SM2 elliptic curve key agreement algorithm[6]. The literature [7] demonstrated that the SM2 elliptic curve key agreement algorithm is more efficient and safer than the key agreement algorithm based on ECDH.

A. PAPER MOTIVATION

Currently, the main security measures between the terminal and the control center involve the secret transmission of data. Data security is typically achieved by encrypted transmission, while both sides of the network must use a shared key for encrypted communication. There are usually two ways to obtain the shared key: one is to preview the key used by the encrypted communication before transferring the communication and prefabricating in the communication device; the other is to negotiate the key before the beginning of the encrypted communication and then use the key to perform the subsequent encryption communication. In the above two ways, the advantage of the first method is that the two parties have no key negotiation resource consumption, but the prefabricated key has a greater risk of being cracked in communications over a long period of time, and the data in communications will be leaked. In addition, once the key is leaked, the process of replacing the prefabricated key is very tedious. The second method requires the two parties to exchange the key to initiate the communication each time, so that the key used in each
communication process is random, which can better protect the security of the communication data and reduce the cost of key management [8]. Therefore, the security and performance of key negotiation fundamentally determines the security and performance of the data encryption transmission, and affects the security of the smart grid communication network in terms of the security communication scheme.

Because of relatively weak security measures and enhancements in hacker attacks, widely distributed smart grid communication networks are faced with various attack risks, and this affects the electrical reliability and safety of the electric system[9]. Changes have been implemented in the current international security scenario. Attackers have launched tortuous attacks on the SC through the misinformation of the ST, causing a greater security threat. The insecurity of various STs and communication networks poses great risks for the SCs in smart grids. In the process where STs access the SC, data transmission links are faced with threats such as attacks, interference, destruction, interception of data and tampering. Once the ST is successfully connected to the SC, it is considered a trusted user of the resources of the SC. If an illegal ST is connected, it becomes a springboard from which the SC is infiltrated, which creates uncontrollable risk for the entire electric power information system [10]. Additionally, the attacker posing as the SC sends malicious control instructions to the STs, which also creates immeasurable consequences for the electric power system and its users.

Recently, the security of smart grid communications has been the subject of a substantial amount of attention. Related works have been performed. However, the published works have several limitations, including:(i) the SC frequently communicates with the STs to transmit measurement values, and the quantity of STs still increases, leaving much to be desired in the communication overload and efficiency terms of published schemes; (ii) the storage and computation of the ST must be considered during the design and realization of the security scheme for smart grid communication. Any overburdening of computational resources reduces the efficiency of the scheme [11]; (iii) although the schemes proved to have adequate performance, their security level is not sufficient. The transmission of information over the communication network between the SC and the STs must repel security attacks through a security solution, e.g., replay attacks, MITM attacks, and impersonation attacks [12]. Consequently, it is essential to properly design a lightweight smart grid communication scheme for key agreement and mutual authentication between the SC and the STs.

B. RELATED WORK
There are several published authentication and key agreement schemes that are applied in different network infrastructures. Global Mobility Network (GLOMONET) is very important for smart city. Li et al. [4] proposed a robust biometrics based three-factor authentication scheme for GLOMONET in smart city. Wireless Sensor Networks (WSN) are important infrastructures in the industrial Internet of Things. Considering the resource-constrained nature of sensor nodes, Li et al. [11] proposed a three-factor anonymous authentication scheme for WSNs, keeping computational efficiency.

The related work on the smart grid communication network security solutions and SM2-based protocol are introduced in this section. Yan et al. [13] proposed a hash-based message authentication code key agreement solution for smart grids. The goal of the solution was to ensure communication safety between the BAN gateway and smart meters. Chen et al. [14] proposed a key establishment and anonymous authentication solution for smart grids, which is based on the computational Daffier-Hellman problem and bilinear maps. Boudia et al. [15] proposed a solution that considers multidimensional aggregation of privacy preservation and a valid confirmation of smart grid information. The proposed solution is based on ECC with unpaired holomorphic encryption. However, cryptographic algorithms used in the above scheme are inefficient due to the lightweight and efficiency needs of smart grid communications. Furthermore, they only achieve one-way authentication, which cannot avoid the malicious control and operation of intelligent terminals by fake intelligent control center attackers.

Recently, Li et al. [16] designed an authentication architecture for the smart grid’s advanced metering infrastructure (AMI), which proves that the architecture is capable of fault diagnosis. However, it does not address the key agreement problem along with the authentication scheme. Li et al. [17] presented the multicast authentication scheme, which reduces the size of the signature and overcomes the memory overhead through a one-time signature. Although the authentication delay and computational complexity of the proposed solution are very low, it does not consider the key agreement. The schemes in [16, 17] do not prevent impersonation attacks and fail to prevent eavesdropping due to the absence of key agreement.

Fouda et al. [18] achieves hash-based authentication and generation of a joint session key between distributed smart terminals (such as smart meters) for the smart grid network. The proposed solution provides key agreement and mutual authentication for the smart grid network. However, the complexity of exponential time in the authentication phase leads to high computational complexity of their scheme. In [19], an ECC-based authentication solution is proposed by Mahmood et al. , which not only provides identity authentication but also exchanges a shared session key to achieve later secure communication. Although the schemes in [18, 19] proved to have adequate security, their performance is not sufficient in practice due to the high speeds and minimal requirements.

In order to ensure the wireless communication security between the reader and the tag in the RFID system, Li et al.
[20] proposed a two-way authentication protocol based on the SM2 and zero-knowledge proof. Although the communication efficiency is high, it fails to prevent eavesdropping due to the absence of key agreement. Yang et al. [21] used the indistinguishability-based Bellare-Rogaway model to prove the security of the SM2 protocol. They also presented a simplified and more efficient version of the SM2 protocol with an accompanying security proof. However, they achieved key agreement without mutual authentication. In the smart grid communication network, secure communications between STs and the SC can be guaranteed only when the two-way identity authentication and the session key agreement are simultaneously implemented. Identity authentication ensures that STs and the SC are not impersonated. Session key agreement ensures that the session key is updated in time, so that subsequent encrypted communication is guaranteed.

C. OUR CONTRIBUTION
A lightweight authentication and key agreement scheme is proposed to address the challenges and issues presented in Section I-A. Our solution is based on the SM2 elliptic curve key agreement algorithm and bidirectional identity authentication based on PKI (PKI, public Key Infrastructure) technology. In our scheme, the key agreement and mutual authentication are triggered once the connection between the SC and the ST is established, which ensures that the granularity of the key agreement and mutual authentication is in every TCP connection. Furthermore, the scheme determines whether the connection has gone over time after each smart grid communication service is over. The key agreement and mutual authentication are restarted for the timeout connection. A shared secret key is generated that is used to encrypt and decrypt the network packets communicated between the SC and the STs. The major contributions of this paper include:

1) Improvement and simplification of the SM2 key agreement algorithm. Due to the ST’s limitations in memory and computational capability, the communication and computational cost of the security scheme between the SC and ST should be as low as possible. We improved and simplified the SM2 key agreement algorithm without changing or decreasing its security.

2) Implement mutual authentication along with the key agreement. The proposed solution achieves lightweight mutual authentication between SCs and STs with identity authentication technology based on the X.509 in the public key cryptosystem.

3) Detailed security analysis and performance evaluation of the proposed scheme. Through an attack scenario and computational complexity analysis of the proposed scheme, as well as a comparison analogy with similar schemes, its security and light weight are verified.

4) Experimental analysis with a security gateway and simulated terminals. Under the network environment of the developed security gateway and the built terminal simulator server, the proposed algorithm is implemented to achieve simulated communication between the SC and STs, and the experimental results are obtained.

D. PAPER ORGANIZATION
The paper is organized as follows: Section II presents related work from the literature. Section III presents preliminaries, including key generation and the derivation function, data type conversion and the notation guide. Section IV addresses the realization of the proposed scheme based on SM2. A security analysis is given in Section V. Section VI practically evaluates the performance of the proposed solution. Finally, Section VII concludes the paper.

II. PRELIMINARIES
This section elaborates the key generation and derivation function of elliptic curve cryptography, the fundamentals of data type conversion and the notation guide [22].

A. KEY GENERATION AND DERIVATION FUNCTION
1) KEY GENERATION
An elliptic curve is a curve in a coordinate system that corresponds to the elliptic curve equation, which is called the Weiertrass equation, as shown below,

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$  

All elliptic curves have their Weiertrass equation. These elliptic curves have common mathematical characteristics, but not all curves can be called an elliptic curve. Algebraically closed fields can be divided into complex fields, real number fields, rational number fields and finite fields. The closed domain of elliptic curve algebra is a finite field. There are two finite fields that can be used to define an encrypted elliptic curve: prime finite field and two element extension field, expressed as $F_p$, $F_{2^m}$. The SM2 algorithm based on a prime finite field is adopted in this paper. The elliptic curve equation on the prime number finite field can be simplified and the prime number $p$ is greater than 3. The equation is as follows: $y^2 = x^3 + ax + b (mod\ p)$, where $a, b \in F_p$, and $4a^3 + 27b^2 \neq 0$.

The SM2 algorithm has the same mathematical theory as the elliptic curve algorithm. It is also based on the operation of multiple points on the elliptic curve, and has been optimized and improved. It takes some effort to find the right elliptic curve. Fortunately, the National Cryptographic Administration has recommended elliptic curve parameters, including $p, a, b$, and base point $G(x_G, y_G)$. The SM2 algorithm uses elliptic curve equations in the 256-bit prime field. Equations and recommended parameters are as follows; $y^2 = x^3 + ax + b (mod\ p)$

a) recommended parameter $p$

$\begin{array}{l}
\text{FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF} \\
\end{array}$
b) recommended parameter \( a \)

\[
\text{FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 00000000} \quad \text{FFFFC}
\]

\( b \)

\[
28E9FA9E 9D9F5E34 4D5A9E4B \quad \text{CF6509A7 F39789F5} 15A88F92 \quad \text{DDBCBDA4} \quad \text{4D940E93}
\]

c) recommended parameter \( b \)

\[
\text{BC3736A2 F4F6779C 59BDCE3E} 68692153 \quad \text{D0A9877C} \quad \text{C62A4740} \quad \text{02DF32E5} \quad \text{2139F0A0}
\]

d) recommended parameter \( x_G \)

\[
\text{FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 7203DF6B} \quad \text{21C6052B} \quad \text{53BBF409} \quad \text{39D54123}
\]

The key pair of a user includes its public key \( P \) and private key \( d \), and \( G \) is the base point, \( n \) is the order of the base point is \( G \).

The generation of the user key pair and the public key verification is as follows:

Key generation:

Input: parameters of an elliptic curve system valid \( F_p \).

Output: a key pair \((d, P)\) associated with the parameters of the elliptic curve system.

a) using an integer random number generator to generate integers \( d \in [1, n-2] \);

b) compute point \( P = (x_p, y_p) = [d]G \);

c) the key pair is \((d, P)\), where \( d \) is the private key and \( P \) is the public key.

Verification of the public key of the elliptic curve on \( F_p \):

Input: a valid \( F_p \) \((p > 3 \text{ and } p \text{ is prime})\) on the elliptic curve system parameter set and a related public key \( P \).

Output: if the public key \( P \) is valid, output "valid"; otherwise, the output is invalid.

a) verify that \( P \) is not infinite point \( O \);

b) verify that the coordinates \( x_p \) and \( y_p \) of the public key \( P \) are elements in the domain \( F_p \) (That is to verify that \( x_p \) and \( y_p \) are the integers in the interval \([0, p-1] \));

c) verify \( y_p^2 \equiv x_p^3 + ax_p + b \mod p \);

d) verify \([n]P = O\);

e) if all validation is passed, the output is valid; otherwise, the output is invalid.

2) KEY DERIVATION FUNCTION

The function of the key derived function is to deduce the key data from a shared secret bit string. During the key exchange process, the key derived function acts upon the secret bit string shared by the key agreement and generates the required session key or the key data needed to further encrypt the key[23].

The cryptographic hash function \( H_v() \) is called by the key

derivative function, and its output is the hash value of the length, which is \( V \) bit.

The key derivative function \( K(Z, L_k) \) process is as follows:

Input: integer \( L_k \) (representing the bit length of the key data to be obtained, and the value is less than \((2^{32} - 1) V \)), bit string \( Z \).

Output: a key data bit string \( K \) with a length of \( L_k \)

a) set a counter \( ct = 0 \times 000000001 \) composed of 32 bits.

b) execute \( i_1 \) from 1 to \( \left[ \frac{L_k}{V} \right] \):

b.1 calculation of \( H_{a_1} = H_v(Z \| ct) \);

b.2 \( ct + + \);

c) if \( \left[ \frac{L_k}{V} \right] \) is integer, \( H_{a_{\left[ \frac{L_k}{V} \right]}} = H_{a_{\left[ \frac{L_k}{V} \right]}} \), otherwise \( H_{a_{\left[ \frac{L_k}{V} \right]}} \) is the leftmost \((L_k - (V \times \left[ \frac{L_k}{V} \right]) \) bit of \( H_{a_{\left[ \frac{L_k}{V} \right]}} \).

d) \( K = H_{a_1} \| H_{a_2} \| \ldots \| H_{a_{\left[ \frac{L_k}{V} \right]-1}} \| H_{a_{\left[ \frac{L_k}{V} \right]}} \).
Input: type of base field $F_p$, byte string $S$ of length 

$$\text{len} = \lceil a/8 \rceil,$$

where $a = \lceil \log_2 p \rceil$

Output: element $\alpha$ in $F_p$

- a) if $p$ is an odd prime number, $S$ is converted to integer $\alpha$ according to the details of Section II.B.2. If $\alpha$ is not in $[0, p-1]$, it is wrong.

- b) if $p = 2^n$, transform the $S$ to bit string $\alpha$ of length $m$ according to the details of Section II.B.4.

7) CONVERSION OF DOMAIN ELEMENTS TO INTEGER

Input: the element $\alpha$ in the domain $F_q$

Output: integer $x$

A) if $q$ is an odd prime number, then $x = \alpha$ (does not need to be converted);

B) if $q = 2^n$, then the $\alpha$ must be a bit string of length $m$, $s_{m-1}, s_{m-2}, \ldots, s_0$ is the bit of $\alpha$ from the leftmost to the rightmost, and transform $\alpha$ to integer $x$:

$$x = \sum_{i=0}^{m-1} 2^i s_i$$

### C. NOTATION GUIDE

The primitive notations related to the proposed scheme are given in Table I.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A,B$</td>
<td>Two users using the public key cryptosystem</td>
</tr>
<tr>
<td>$G$</td>
<td>The base point of the elliptic curve, order of $G$ is a prime number where $G = (x_G, y_G)$</td>
</tr>
<tr>
<td>$E(F_q)$</td>
<td>The set of all points(including infinite point $O$) on the elliptic curve $E$ defined over $F_q$</td>
</tr>
<tr>
<td>$d_A,d_B$</td>
<td>Private key of user $A$ and user $B$</td>
</tr>
<tr>
<td>$F_q$</td>
<td>The prime field $F$ includes $q$ elements</td>
</tr>
<tr>
<td>$H()$</td>
<td>Hash function</td>
</tr>
<tr>
<td>$H_v()$</td>
<td>Cipher hash function whose message digest length is $V$ bit</td>
</tr>
<tr>
<td>$ID_A,ID_B$</td>
<td>Identification of user $A$ and user $B$</td>
</tr>
<tr>
<td>$K_A,K_B$</td>
<td>Session shared key confirmed from the key agreement</td>
</tr>
<tr>
<td>$K()$</td>
<td>Key derivation function</td>
</tr>
<tr>
<td>$n$</td>
<td>The order of the base point $G$</td>
</tr>
<tr>
<td>$h$</td>
<td>Cofactor, $h = #E(F_q)/n$</td>
</tr>
<tr>
<td>$q$</td>
<td>The number of elements in a prime field $F_q$</td>
</tr>
</tbody>
</table>
| $O$       | A special point on the elliptic curve is called the infinite point or zero point. It is the unit element of the elliptic curve.
D. SECURITY MODEL

We consider that attacker $\delta$ knows the public identity of STs and SC, and can also access the public communication channel. He can eavesdrop on data transmitted in the smart grid communication network, and try to identify cipher text content. For example, $\delta$ steals vehicle location information, user information, user payment information, charging pile processing information when electric vehicles enter the network, thus inferring the user's range of activity, driving path and driving distance and other life habits information. $\delta$ can also execute active attacks, including tampering, replaying, injecting messages, etc. For example, attacker $\delta$ tampers with the user's power consumption records sent to the control center by a smart meter, which can lead to erroneous decision-making by the control center. The attacker $\delta$ injects false control commands into the data transmitted by the control center, causing RTU to make incorrect actions. In addition, we assume that the trusted third-party CA (CA, certificate Authority) is secure and $\delta$ cannot obtain its key.

III. PROPOSED SCHEME

The SM2-based lightweight authentication and key agreement scheme for the smart grid is presented in this section. First, the SM2 key agreement protocol is elaborated, and it is also depicted in Figure 2.

A. SM2 KEY AGREEMENT PROTOCOL

The length of the key obtained by the key agreement process between user A and user B is $L_\delta$ bits, user A is the initiator, and user B is the responder.

To obtain the same key, both user A and B should implement the following calculation steps:

User A:

A1: randomly selects $a \in [1, n-1]$;

A2: computes elliptic curve point $M_A = [a]G = (x_A, y_A)$;

A3: sending $M_A$ to B;

User B:

B1: randomly selects $b \in [1, n-1]$;

B2: computes the elliptic curve point $M_B = [b]G = (x_B, y_B)$;

B3: takes out the domain element $x_B$ from $M_B$, converts the data type of $x_B$ to an integer according to Section II.B.7, computes $x_B = 2^\omega + (x_B \& (2^\omega - 1))$, where $\omega = \lfloor (\lfloor \log_2(n) \rfloor / 2) \rfloor - 1$;

B4: computes $t_B = (d_B + x_B \cdot b) \mod n$;

B5: verifies that $R_A$ satisfies the elliptic curve equation. If it is not satisfied, the key agreement fails.

Unless the domain element $x_i$ is taken out of $R_A$, the data type of $x_i$ is converted to an integer according to Section II.B.7, which computes $x_i = 2^{\omega_i} + (x_i \& (2^{\omega_i} - 1))$, where $\omega_i = \lfloor (\lfloor \log_2(n) \rfloor / 2) \rfloor - 1$;

B6: computes elliptic curve point $\beta = \{h \cdot t_B\} (M_A + [x_A]M_A) = (x_\beta, y_\beta)$.

If $V$ is an infinity point, then user B fails in the agreement, unless the data type of $x_V, y_V$ is converted to a bit string according to Section II.B.5 and Section II.B.4;

B7: computes $K_B = K(x_B \parallel y_\beta \parallel Z_A \parallel Z_B \parallel L_\delta)$;

B8: (optional for key confirmation) converts the data type of $x_A, y_A$ and $x_B, y_B$ to bit string according to Section II.B.5 and Section II.B.4, computes $W_B = H(0 \times 02 \parallel \{x_\beta \parallel H(x_\beta \parallel Z_A \parallel Z_B \parallel x_A \parallel y_A \parallel x_B \parallel y_B)\}$;

B9: sends $M_B$ (optional $W_B$) to user A

User A:

A4: takes out the domain element $x_A$ from $M_A$, converts the data type of $x_A$ to an integer according to the method given in Section II.B.7, and computes $x_A = 2^{\omega} + (x_A \& (2^\omega - 1))$, where $\omega = \lfloor (\lfloor \log_2(n) \rfloor / 2) \rfloor - 1$;

A5: computes $t_A = (d_A + x_A \cdot a) \mod n$;

A6: verifies that $M_B$ satisfies the elliptic curve equation. If it is not satisfied, the key agreement fails.

Unless the domain element $x_B$ is taken out of $M_B$, the data type of $x_B, y_B$ is converted to an integer according to the method given in Section II.B.7, and it computes $x_B = 2^{\omega} + (x_B \& (2^\omega - 1))$, where...
\[ \omega = \left[ \left( \log_2(n) / 2 \right) \right] - 1; \]

A7: computes elliptic curve point \( \alpha = [h \cdot t_A \cdot (P_B + [x_B]M_B)] = (x_\alpha, y_\alpha) \). If \( U \) is an infinity point, then user \( A \) fails in the agreement, unless the data type of \( x_\alpha, y_\alpha \) is converted to a bit string according to the method given in Section II.B.5) and Section II.B.4):

A8: computes \( K_A = K(x_\alpha \parallel y_\alpha \parallel Z_A \parallel Z_B \parallel L_A) \);

A9: (optional for key confirmation) converts the data type of \( x_A, y_A \) and \( x_B, y_B \) to a bit string according to the method given in Section II.B.5) and Section II.B.4):

A10: (optional for key confirmation) computes

\[ W_A = H(0x03 \parallel y_A \parallel H(x_A \parallel Z_A \parallel Z_B \parallel x_A \parallel y_A \parallel x_B \parallel y_B)) \]

, and sends \( W_A \) to client \( B \);

User B:

B10: (optional for key confirmation) computes \( W_B = H(0x03 \parallel y_B \parallel Hash(x_B \parallel Z_A \parallel Z_B \parallel x_A \parallel y_A \parallel x_B \parallel y_B)) \)

, verifies whether \( W_B = W_A \), and if it returns false, user \( A \) fails in the agreement.

At this point, the entire SM2 key exchange protocol is complete.
process ends. The session key established is \( K_A = K_B \).

**B. PROPOSED SECURITY COMMUNICATION SCHEME**

In the algorithm described above, user A and user B can obtain a shared secret key on the unsafe communication channel. However, due to the high concurrency of the SC and the limited computing power and storage capacity of the ST, we simplified the SM2 key agreement scheme. In addition, the mutual identity authentication is implemented along with the simplified SM2 key agreement scheme. As a result, a lightweight secure communication scheme based on PKI (PKI, public Key Infrastructure) technology for smart grid communication is proposed. Separately, the scheme is described in three stages including initialization, mutual authentication and key agreement. It is represented in Fig. 3.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>((d_A, \ E(F_{P_1}), H(\cdot), K(\cdot), n, P_A, P_B, Z_A, Z_B))</td>
<td>((d_B, \ E(F_{P_2}), H(\cdot), K(\cdot), n, P_A, P_B, Z_A, Z_B))</td>
</tr>
</tbody>
</table>

Randomly selects \( r_A, S_A, ID_A \) | Randomly selects \( r_B, S_B, ID_B \)

Computes Request=\( E_{ID_A}(r_A, S_A, ID_A) \) | Computations \( D_{ID_B}(r_B, S_B, ID_B) \)

Replay | Computations \( D_{ID_A}(H(r_A) \oplus S_A) \)

Verifies \( D_{ID_B}(E_{ID_A}(H(r_A) \oplus S_A)) = H(r_A) \oplus S_A \) | Computations \( E_{ID_A}(H(r_A) \oplus S_A) \)

Selects \( P_A = (x_{A1}, y_{A1}) \) | Selects \( P_B = (x_{B1}, y_{B1}) \)

Computes \( X_{A2} = x_A \cdot X_{P2} + Z_A \cdot (2^{e-1}) \) | Computes \( X_{B2} = x_B \cdot X_{P2} + Z_B \cdot (2^{e-1}) \)

Computes \( t = (d_A \cdot X_{P2} + r_A) \mod n \) | Computes \( V = (t \cdot b) \cdot P_2 \cdot X_{P2} \cdot P_2 = (x_B, y_B) \)

Computes \( \bar{r}_B = r_A \cdot r_B \) | Computes \( K_A = K \oplus x_A \oplus Z_A \oplus t \oplus r_B \)

\( R_B \rightarrow \) | \( R_A \rightarrow \)

Session key established is \( K_A = K_B \) | Session key established is \( K_B = K_A \)

**FIGURE 3 Proposed scheme**

1) **INITIALIZATION AND MUTUAL AUTHENTICATION**

In the proposed scheme, the private key \( d_A \) and \( d_B \) depend on the security chips held by user A and user B respectively. The private key is generated by a random number generator and stored in the chip storage unit before the security chip is released, and it cannot be exported. User A and user B send certificates request files to the CA, and the CA issues their signature certificates to them. The respective certificates include public key \( P_A \) and \( P_B \). Next, user A and user B will be able to perform the following mutual authentication.

The length of the key obtained by the key agreement process between users A and B is \( L_\alpha \) bits, user A is the initiator, and user B is the responder. To obtain the same
key, both user A and B should implement the following initialization and mutual authentication:

User A:

A1: randomly selects \( r_A, Se, ID_A \)

User B:

B1: randomly selects \( r_B, ID_B \)

A2: computes \( E_{p_A}(r_A \| Se \| ID_A) \), \( E_{d_A}(H(r_A \| Se \| ID_A)) \), sends Request= \( E_{p_A}(r_A \| Se \| ID_A) \| E_{d_A}(H(r_A \| Se \| ID_A)) \) to user B.

B2: computes \( D_{p_B}(E_{p_B}(r_B \| Se \| ID_A)) \) to obtain \( r_A \cdot Se \cdot ID_A \), and computes \( H(r_B \| Se \| ID_B) \)

B3: verifies \( D_{p_B}(E_{d_B}(H(r_B \| Se \| ID_B))) \).

A4: verifies \( D_{p_A}(E_{d_B}(H(r_B \| Se \| ID_B))) \).

Request= \( E_{p_A}(r_A \| Se \| ID_A) \| E_{d_B}(H(r_B \| Se \| ID_B)) \) to user B.

A5: computes \( D_{d_A}(E_{p_A}(r_B \| Se \| ID_A)) \) to obtain \( r_B \cdot Se \cdot ID_B \), and computes \( H(r_B \| Se \| ID_B) \)

B4: sends Replay= \( E_{p_A}(r_A \| Se \| ID_A) \| E_{d_B}(H(r_B \| Se \| ID_B)) \) to user B.

User A:

A6: computes \( t_A = (d_A + x_{p_A} \cdot r_A) \mod n \).

User B:

B6: takes out the domain element \( x_{p_2} \) from \( P_2 \), converts the data type of \( x_{p_2} \) to an integer according to the function given in Section II.B.7), computes \( x_{p_2} = 2^\omega + (x_{p_2} & (2^\omega - 1)) \), where \( \omega = \left\lceil \left( \left\lfloor \log_2(n) \right\rfloor / 2 \right\rfloor \right\rceil - 1 \).
At this point, the entire scheme process ends. The session key established is \( K_A = K_B \).

The granularity of authenticated negotiation between users is a TCP connection. Each connection requires authentication negotiation communication. After closing the connection, the session key will be deleted immediately and will not be used again. Each communication must determine whether the connection is timeout. If the timeout is required, authentication negotiation is required again.

**IV. SECURITY ANALYSIS**

**A. MUTUAL AUTHENTICATION**

In the proposed scheme, the user’s private key is generated by the random number generator inside the security chip held by the user. It is stored in the storage unit of the security chip and cannot be exported. The trusted third-party CA publishes the user's signature certificate through PKI technology. The certificate includes the user's public key. The security model states that the CA is secure. Therefore, only legitimate users can have the correct signature private key corresponding to the signature public key. User \( A \) determines \( (r_A \| Se \| ID_A) \) and sends \( E_{p_b}(r_A \| Se \| ID_A) \| E_{d_A}(H(r_A \| Se \| ID_A)) \) to user \( B \).

User \( B \) authenticates user \( A \) by verifying \( D_{p_b}(E_{d_A}(H(r_A \| Se \| ID_A))) = H(D_{d_A}(E_{p_b}(r_A \| Se \| ID_A))) \). The computation of user \( B \) involves user \( A \) ’s public key \( P_A \). If the return result is true, the message is sent by the legitimate user \( A \) involving user \( A \) ’s secret key \( d_A \). For the same reason, user \( A \) authenticates user \( B \) by verifying \( D_{p_b}(E_{d_A}(H(r_b \| Se \| ID_B))) = H(D_{d_A}(E_{p_b}(r_b \| Se \| ID_B))) \). Hence, both user \( A \) and \( B \) achieve mutual authentication for each other.

**B. REPLAY ATTACK**

User \( A \) sends an authentication request to \( B \) containing random number \( r_A \), which is not only sent in a cipher-text, but it is also concealed in \( E_{p_b}(r_A \| Se \| ID_A) \| E_{d_A}(H(r_A \| Se \| ID_A)) \). Therefore, if the adversary \( \delta \) replays a former message from user \( A \), user \( B \) can identify and check the recentness/freshness of random number \( r_A \) by verifying \( D_{p_b}(E_{d_A}(H(r_A \| Se \| ID_A))) = H(D_{d_A}(E_{p_b}(r_A \| Se \| ID_A))) \). Therefore, if \( \delta \) replays a former message from user \( A \), \( B \) checks the recentness/freshness of random number \( r_A \) to identify it and fails the authentication request [24]. As a result, the user in our solution can resist a replay attack.

**C. IMPERSONATION ATTACK**

Users can authenticate the message source. As described in Section IV, user \( B \) verifies \( D_{p_b}(E_{d_A}(H(r_A \| Se \| ID_A))) = H(D_{d_A}(E_{p_b}(r_A \| Se \| ID_A))) \). If the verification result is true, the message source is identified as a legitimate user \( A \). In contrast, an impersonation attack is found. Because the security model states that the CA is secure. Only legitimate users can have the correct signature private key corresponding to the signature public key. The illegitimate user \( \delta \) cannot pass the authentication process.

**D. MESSAGE INJECTION ATTACK**

Our scheme provides an encrypted channel over the transmitted data sent between the SC and STs after the key agreement and authentication. The session key is known only by users who have achieved the authentication and key agreement. The attacker \( \delta \) cannot obtain the shared session secret key. It is not possible to inject malicious information, since \( \delta \) cannot correctly decrypt the network packets without the shared session secret key. If \( \delta \) sends malicious data encrypted by the SC or the ST, the SC or the ST will not correctly decrypt the malicious data with the correct shared session key, and the error data is discarded[25].

**E. MAN-IN-THE-MIDDLE ATTACK**

The adversary \( \delta \) may receive message \( E_{p_b}(r_A \| Se \| ID_A) \| E_{d_A}(H(r_A \| Se \| ID_A)) \) from user \( A \). The adversary \( \delta \) cannot obtain \( (r_A \| Se \| ID_A) \) without the secret key of user \( B \), and cannot make a reply to user \( B \). For the same reason, \( \delta \) cannot make a correct reply to user \( A \). Thus, the proposed scheme protects against MITM attacks.

**F. EAVESDROPPING FOR A LONG TIME**

Suppose that the attacker \( \delta \) performs a brute-force attack or an offline dictionary attack to crack the shared session key. Because of the granularity of mutual authentication, the key agreement occurs with every TCP connection. The scheme judges whether the connection is over time after each smart grid communication service is over. The key agreement and mutual authentication are restarted to obtain the new shared session key for the timeout connection. The adversary cannot obtain the new session key due to the shared session secret key refresh and updating in a short-time. As a result, once the shared session key becomes effective, \( \delta \) cannot eavesdrop for a long time on the session between the SC and the ST in our scheme.

**G. PERFECT FORWARD Secrecy**

The elements used to generate the key are changed one at a time, and no other keys can be generated; one key is cracked and does not affect the security of other keys. Our scheme.
has this characteristic, called perfect forward secrecy (PFS) [19]. The proposed scheme will periodically create new key based on the random numbers provided by both parties. Since both sides provide a random value that only they know at the time of the key agreement, each new key generated is different from the previously created key. Even if the adversary δ intercepts the key, it cannot use the intercept key for a long time. In addition, since the fresh key through key agreement is not obtained from the previously intercepted key, δ must start a new brute-force calculation to get the key that is in use. Hence, it is somewhat strenuous for an adversary to obtain the session keys.

Finally, we compare several features of the proposed scheme with similar schemes. The comparison result is explained in Table II , and shows that the proposed scheme has more security characteristics.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Proposed</th>
<th>[1]</th>
<th>[19]</th>
<th>[26]</th>
<th>[13]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key agreement</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mutual authentication</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Replay attack</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Impersonation attack</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Information injection</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Shared key update</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MIMT</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Perfect forward secrecy</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**V. PERFORMANCE EVALUATION AND EXPERIMENTAL ANALYSIS**

### A. PERFORMANCE EVALUATION

In this section, the computational complexity of the proposed scheme is compared with similar solutions by identifying the primitive operations and computing their frequency separately. The related primitive operations notations are listed in Table III. The execution time of the primitive operations is presented in [27]. Furthermore, the communication overhead in our scheme is evaluated by selecting a 16-bit-long ID, an 8-bit random number r, an 8-bit session identifier Se, a 64-bit public keys and a 160-bit ECC operation size. Table IV shows that our scheme outperforms the other similar schemes both in computational complexity and communication overhead terms.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computation cost</th>
<th>Communication cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>[13]</td>
<td>14T_h=26.5123ms</td>
<td>960 bits</td>
</tr>
<tr>
<td>[1]</td>
<td>5T_VS+2T_ESM=10.4968ms</td>
<td>640 bits</td>
</tr>
<tr>
<td>[26]</td>
<td>T_SE+T_SD+2T_ESM+2T_SE+2T_SD=19.8658ms</td>
<td>608 bits</td>
</tr>
<tr>
<td>[19]</td>
<td>5T_SE+5T_HMAC+1+1.1703ms</td>
<td>576 bits</td>
</tr>
<tr>
<td>Proposed</td>
<td>2T_SE+2T_ESM+2T_SD+2T_VS+2T_ME+2T_ESM=9.1328ms</td>
<td>422 bits</td>
</tr>
</tbody>
</table>

### B. EXPERIMENTAL ANALYSIS

For the support of our proposal, the smart grid control center can deploy a security gateway to achieve secure communication between the SC and the intelligent terminal. Thus, the level of service in the smart grid communication system is clearer, and the control center is in the business layer and responsible for all business processing; the security gateway is in the security layer and is responsible for the authentication of the smart grid terminal and the encryption of the data link. Therefore, our proposal is the implementation with the security gateway.

In the experimental analysis, we use a security gateway and a terminal simulator server. The security gateway and the terminal simulator server run on the same allocation servers with an Intel (R) Xeon CPU E3-1230 V3 @ 3.40GHz processor, which uses GCC with Linux C and Putty to implement the communication scenarios in our scheme. The hardware parameters of the processor are as follows: the memory size is 16GB, the hard disk size is 1 TB HDD, and the operating system is CentOS 6.6 * x64. Figure 4 shows the network topology diagram of the experimental evaluation.

**FIGURE 4. Network topology diagram of the experimental evaluation**

First, we implement the authentication and key agreement process. The terminal simulator server simulates a legitimate terminal making an authenticate request to the security gateway. As shown in Figure 5.a, both sides successfully authenticate and obtain the session key. By modifying the
certificate, the terminal emulator server simulates an illegal terminal. As shown in Figure 5.b, the authentication fails, and the key agreement process terminates. Figure 6.a-g shows that the terminal simulation server simulates the terminals separately, ranging from 1 to 5000 terminals, and obtains the execution time of our scheme.

**FIGURE 5.** A GUI of Putty: The authentication and key agreement process of the proposed scheme.(a) Both sides successfully authenticate and obtain the session key.(b) The authentication fails, and the key agreement process terminates

**FIGURE 6.** A GUI of Putty: The terminal simulation server simulates the terminals separately, ranging from 1 to 5000 terminals, and obtains the execution time of our scheme. (a) The execution time of 1 terminal. (b) The execution time of 200 terminals. (c) The execution time of 500 terminals. (d) The execution time of 1000 terminals. (e) The execution time of 2000 terminals. (f) The execution time of 5000 terminals.
To study the execution time of our scheme, experiments have been conducted. Figure 7 shows the execution time of our scheme, and the scheme in [1, 13, 19, 26]. According to Figure 7, with the increasing number of STs, the execution time markedly increases in the scheme of [13, 26]. In contrast, the execution time is relatively low in our scheme and [1, 19]. Obviously, our scheme achieves more efficiency than the scheme in [1, 13, 19, 26].

FIGURE 7. The execution time in related schemes.

VI. CONCLUSION

In this paper, an SM2-based security communication scheme for key agreement and authentication in smart grid is proposed. The proposed scheme is also applicable for smart grid communication networks due to its lightweight operations. The scheme not only obtains a secret session key for subsequent encrypted transmission but also achieves mutual authentication between the SC and STs. Furthermore, the proposed algorithm is analyzed to confirm its robustness against cyber-attacks. Comparison with contemporary related schemes and experimental analysis further shows that our scheme incurs the lowest computation and communicational cost.

REFERENCES

[20] Li Zichen (Dean's Office, Beijing Inst. of Graphic Commun., Beijing, China); Liu Boya; Wang Peidong; Yang Yatao Source: Computer Engineering, v 43, n 6, p 97-100, 15 June 2017 Language: Chinese.
Wei Li received a B.S. degree in software engineering from the University of North China Electric Power University, Beijing, China, in 1987. She is a professor at the School of Control and Computer Engineering, University of North China Electric Power University. Her interests focus on smart grid software technology, network security, and machine learning.

Rui Li received a B.S. degree in information security from the University of North China Electric Power University, Beijing, China, in 2016. She is currently working towards a M.S. degree at the Beijing Engineering Research Center of Electric Information Technology, School of Control and Computer Engineering, University of North China Electric Power University. Her interests focus on information security and smart grid.

Kehe Wu received a Ph.D. degree from the University of North China Electric Power University, Beijing, in 2009. He is a professor at North China Electric Power University, the director of the Chinese Association for Artificial Intelligence and Beijing Engineering Research Center of Electric Information Technology, and a committee member of the China Electric Power Information Standardization Committee and Professional Electric Power Information Committee of the Chinese Society for Electrical Engineering. His research interests include smart grid software technology, power information security, and deep learning.

Rui Cheng received a M.S. degree in computer technology and application from the University of North China Electric Power University, Beijing, China, in 2017. He is currently working at the Beijing Engineering Research Center of Electric Information Technology, School of Control and Computer Engineering, University of North China Electric Power University. His interests focus on power information security.

Liping Su received a B.S. degree in computer science and technology from the Tianjin University, Tianjin, China, in 1989. She is an associate professor at the School of Control and Computer Engineering, University of North China Electric Power University. Her interests focus on smart grid software technology, network security, and machine learning.

Wenchao Cui received a Ph.D. degree from the University of North China Electric Power University, Beijing, in 2014. He works at the School of Control and Computer Engineering, University of North China Electric Power University. His interests focus on smart grid software technology, network security, and machine learning.