IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 10, 2018, accepted October 2, 2018, date of publication October 15, 2018,
date of current version December 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875681

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, AND WENCHAO CUI
North China Electric Power University, Beijing 102206, China

Corresponding author: Rui Li (1162227089@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 (SC), such as RTU, charge pile control terminal, centralized meter reading terminal, and so on. The communication between the SC and the smart grid terminals is 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 literature. 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 costs.

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, security, 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

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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

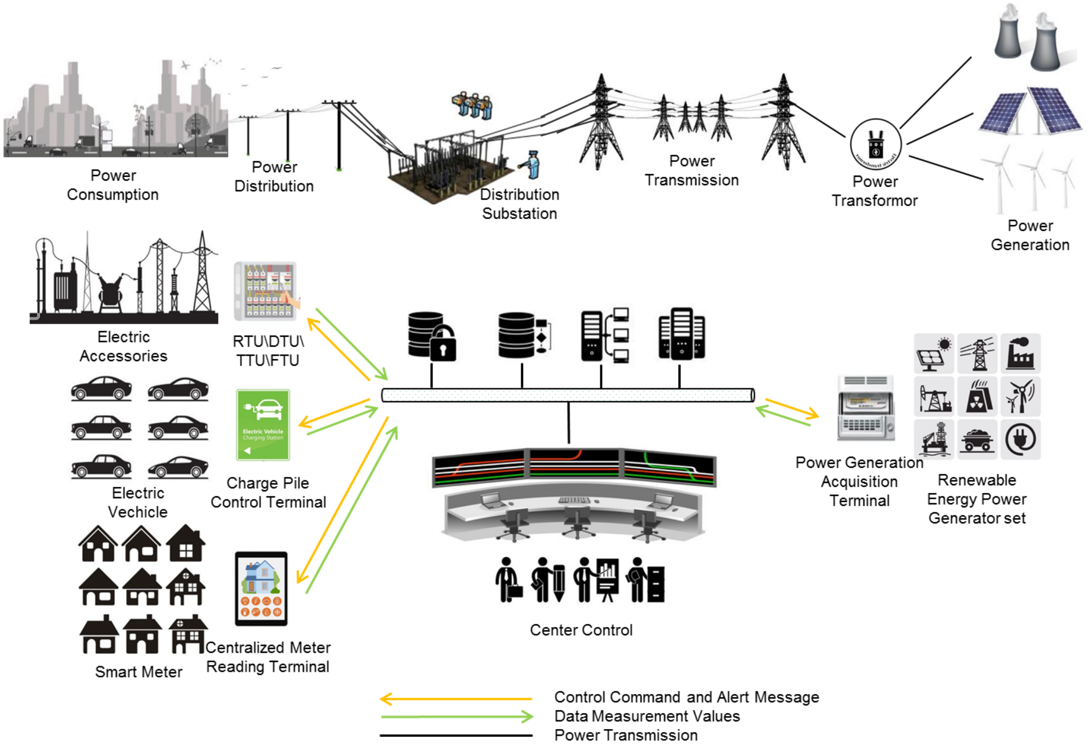


FIGURE 1. Smart grid communication network structure.

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 multi-dimensional 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 and Cao [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], and [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], and [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 [21].

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: $y^2 = x^3 + ax + b(\mathrm{mod}p)$, 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;

$$y^2 = x^3 + ax + b(\mathrm{mod}p)$$

a) recommended parameter p
FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF
b) recommended parameter a
FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFC
c) recommended parameter b
28E9FA9E 9D9F5E34 4D5A9E4B CF6509A7 F39789F5 15AB8F92 DDBCBD41 4D940E93

d) recommended parameter x_G
BC3736A2 F4F6779C 59BDCEE3 68692153 D0A9877C C62A4740 02DF32E5 2139F0A0

e) recommended parameter y_G
FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF 7203DF6B 21C6052B 53BBF409 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 ($p > 3$ and p 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 $y_P^2 \equiv x_P^3 + ax_P + b(\text{mod} p)$;

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 derivative 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 [22].

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 $ct = 0x00000001$ composed of 32 bits.

b) execute i from 1 to $\lceil L_k/v \rceil$:

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

b.2) $ct++$;

c) if $\lceil L_k/v \rceil$ is integer, $H_{a_{!\lceil L_k/v \rceil}} = H_{a_{\lceil L_k/v \rceil}}$, otherwise $H_{a_{!\lceil L_k/v \rceil}}$ is the leftmost $(L_k - (v \times \lfloor L_k/v \rfloor))$ bit of $H_{a_{\lceil L_k/v \rceil}}$.

d) $K = H_{a_1}||H_{a_2}|| \cdots ||H_{a_{\lceil L_k/v \rceil -1}}||H_{a_{!\lceil L_k/v \rceil}}$.

3) OTHER FUNCTIONS

User A has a discernable identifier ID_A with a length of len_A bits, and L_A is a two byte data converted from an integer len_A; user B has an identifiable identification ID_B with a length of len_B bits, and a L_B is a two-byte data converted from integer len_B. In the elliptic curve key exchange scheme, both user A and user B participating in the key agreement get the hash value Z_A of user A and the hash value Z_B of user B by the cryptographic hash function. According to the method given in the first part of Section II.B.5) and Section II.B.4), the data type of the elliptic curve equation parameters a, b, G coordinates x_G, y_G and P_A's coordinates x_A, y_A are converted to a bit string; then, $Z_A = H_{256}(L_A||ID_A||a||b||x_G||y_G||x_A||y_A)$; the data type of the elliptic curve equation parameters a, b, G coordinates x_G, y_G and P_B's coordinates x_B, y_B are converted to the bit string; then, $Z_B = H_{256}(L_B||ID_B||a||b||x_G||y_G||x_B||y_B)$.

B. DATA TYPE CONVERSION

1) CONVERSION OF INTERGER TO BYTE STRING

Input: non negative integer x, and byte string's target length k (where k satisfies $2^{8k} > x$).

Output: a byte string M with a length of k

a) $M_{k-1}, M_{k-2}, \cdots, M_0$ is the bytes of M from the leftmost to the rightmost.

b) The bytes of the M meet:

$$x = \sum_{i=0}^{k-1} 2^{8i} M_i$$

2) CONVERSION OF BYTE STRING TO INTERGER

Input: a byte string M with a length of k

Output: integer x

a) $M_{k-1}, M_{k-2}, \cdots, M_0$ is the bytes of M from the leftmost to the rightmost.

b) convert M to an integer x:

$$x = \sum_{i=0}^{k-1} 2^{8i} M_i$$

3) CONVERSION OF BIT STRING TO BYTE STRING

Input: a bit string s with a length of m.

Output: a byte string M with a length of k, of which $k = \lceil m/8 \rceil$

a) $s_{m-1}, s_{m-2}, \cdots, s_0$ is the bit of s from the leftmost to the rightmost.

b) $M_{k-1}, M_{k-2}, \cdots, M_0$ is the bytes of M from the leftmost to the rightmost, $M_i = s_{8i+7}s_{8i+6} \cdots s_{8i+1}s_{8i}$, where $0 \le i \le k$, when $8i + j \ge m$ and $0 < j \le 7$, $s_{8i+j} = 0$

4) CONVERSION OF BYTE STRING TO BIT STRING

Input: a byte string M with a length of k

Output: a bit string s with a length of m, where $m = 8k$

a) $M_{k-1}, M_{k-2}, \cdots, M_0$ is the bytes of M from the leftmost to the rightmost.

b) $s_{m-1}, s_{m-2}, \cdots, s_0$ is the bit of s from the leftmost to the rightmost, then s_i is the $i - 8j + 1$ bit on the right of M_j, where $j = \lfloor i/8 \rfloor$

5) CONVERSION OF DOMAIN ELEMENTS TO BYTE STRING

Input: element α in F_p
Output: the byte string S of length $len = \lceil a/8 \rceil$, where $a = \lceil \log_2 p \rceil$

a) if p is an odd prime number, α is an integer in interval $[0, p-1]$, and transform α into the byte string S of length l according to the details of Section II. B.1).

b) if, $p = 2^n$ the α is a bit string of length n, and transform the α to the byte string S of length l according to the details of Section II.B.3).

6) CONVERSION OF BYTE STRING TO DOMAIN ELEMENTS

Input: type of base field F_p, byte string S of length $len = \lceil a/8 \rceil$, where $a = \lceil \log_2 p \rceil$
Output: element α in F_p

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

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

7) CONVERSION OF DOMAIN ELEMENTS TO INTERGER

Input: the element α in the domain F_p
Output: integer x

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

B) if $p = 2^m$, then the α must be a bit string of length m, $s_{m-1}, s_{m-2}, \cdots, s_0$ is the bit of α from the leftmost to the rightmost, and transform α 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 1. Notation guide.

Notations	Description
A, B	Two users using the public key cryptosystem
G	The base point of the elliptic curve, order of G is a prime number where $G = (x_G, y_G)$
$E(F_q)$	The set of all points(including infinite point O) on the elliptic curve E defined over F_q
d_A, d_B	Private key of user A and user B
F_q	The prime field F includes q elements
$H()$	Hash function
$H_v()$	Cipher hash function whose message digest length is v bit
ID_A, ID_B	Identification of user A and user B
K, K_A, K_B	Session shared key confirmed from the key agreement
$K()$	Key derivation function
n	The order of the base point G
h	Cofactor, $h = \# E(F_q) / n$
q	The number of elements in a prime field F_q
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 additive group
P_A, P_B	Public key of user A and user B
$x \| y$	Concatenation of two strings x and y ,where x, y can be a bit string or a byte string
Z_A	The hash value of user A identifiable identifier, user A public key and partial elliptic curve system parameters
Z_B	The hash value of user B identifiable identifier, user B public key and partial elliptic curve system parameters
$[k]P$	The k times point of the point P on the elliptic curve, k is a positive integer

D. SECURITY MODEL

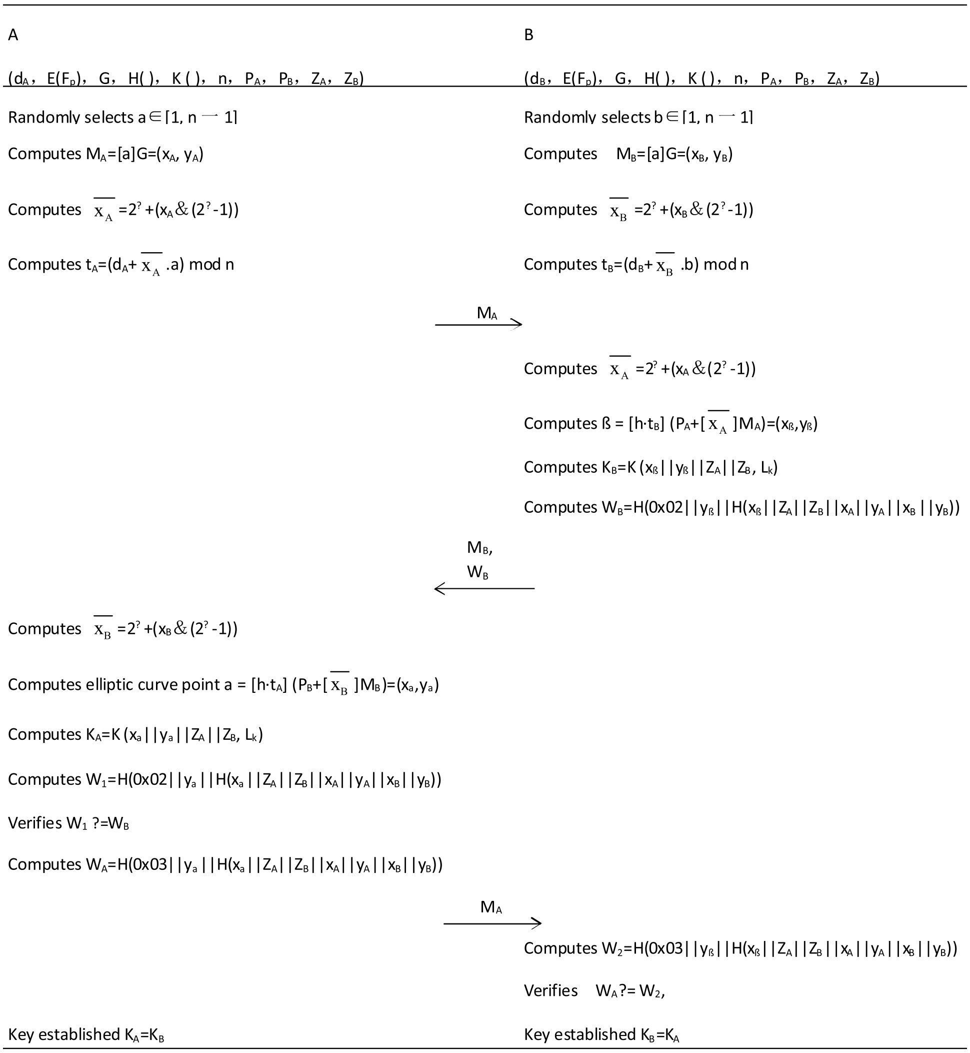
We consider that attacker δ knows the public identity of STs and SC, and can also access the public communication channel. He can eavesdrop on data transmitted in the smart grid communication network, and try to identify cipher text content. For example, δ steals vehicle location information, user information, 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. δ can also execute active attacks, including tampering, replaying, injecting messages, etc. For example, attacker δ 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 δ 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 δ 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_k bits, user A is the initiator, and user B is the responder.



A

$(d_A, E(F_p), G, H(\,), K(\,), n, P_A, P_B, Z_A, Z_B)$

Randomly selects $a \in [1, n - 1]$

Computes $M_A=[a]G=(x_A, y_A)$

Computes $\overline{x_A} =2^? +(x_A \& (2^? -1))$

Computes $t_A=(d_A+ \overline{x_A} .a) \bmod n$

B

$(d_B, E(F_p), G, H(\,), K(\,), n, P_A, P_B, Z_A, Z_B)$

Randomly selects $b \in [1, n - 1]$

Computes $M_B=[a]G=(x_B, y_B)$

Computes $\overline{x_B} =2^? +(x_B \& (2^? -1))$

Computes $t_B=(d_B+ \overline{x_B} .b) \bmod n$

$\xrightarrow{M_A}$

Computes $\overline{x_A} =2^? +(x_A \& (2^? -1))$

Computes ß = $[h{\cdot}t_B]\ (P_A+[\overline{x_A}]M_A)=(x_ß,y_ß)$

Computes $K_B=K(x_ß||y_ß||Z_A||Z_B, L_k)$

Computes $W_B=H(0x02||y_ß||H(x_ß||Z_A||Z_B||x_A||y_A||x_B||y_B))$

$\xleftarrow{M_B, W_B}$

Computes $\overline{x_B} =2^? +(x_B \& (2^? -1))$

Computes elliptic curve point a = $[h{\cdot}t_A]\ (P_B+[\overline{x_B}]M_B)=(x_a,y_a)$

Computes $K_A=K(x_a||y_a||Z_A||Z_B, L_k)$

Computes $W_1=H(0x02||y_a||H(x_a||Z_A||Z_B||x_A||y_A||x_B||y_B))$

Verifies W_1 ?=W_B

Computes $W_A=H(0x03||y_a||H(x_a||Z_A||Z_B||x_A||y_A||x_B||y_B))$

$\xrightarrow{M_A}$

Computes $W_2=H(0x03||y_ß||H(x_ß||Z_A||Z_B||x_A||y_A||x_B||y_B))$

Verifies W_A?= W_2,

Key established $K_A=K_B$

Key established $K_B=K_A$

FIGURE 2. SM2 key agreement protocol.

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 $\overline{x_B} = 2^\omega + (x_B \& (2^\omega - 1))$, where $\omega = \lceil (\lceil \log_2(n) \rceil /2) \rceil - 1$;

B4: computes $t_B = (d_B + \overline{x_B} \cdot b) \bmod n$;

B5: verifies that R_A satisfies the elliptic curve equation. If it is not satisfied, the key agreement fails. Unless the domain element x_1 is taken out of R_A, the data type of x_1 is converted to an integer according to Section II.B.7), which computes $\overline{x_A} = 2^\omega + (x_A \& (2^\omega - 1))$, where $\omega = \lceil (\lceil \log_2(n) \rceil /2) \rceil - 1$;

B6: computes elliptic curve point $\beta = [h \cdot t_B](P_A + [\overline{x_A}]M_A) = (x_\beta, y_\beta)$, If β is an infinity point, then user B fails in the agreement, unless the data type of x_β, y_β is converted to a bit string according to Section II.B.5) and Section II.B.4);

B7: computes $K_B = K(x_\beta||y_\beta||Z_A||Z_B, L_k)$;

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 $W_B = H(0x02||y_\beta||H(x_\beta||Z_A||Z_B||x_A||y_A||x_B||y_B))$

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 $\overline{x_A} = 2^\omega + (x_A \& (2^\omega - 1))$, where $\omega = \lceil (\lceil \log_2(n) \rceil /2) \rceil - 1$;

A5: computes $t_A = (d_A + \overline{x_A} \cdot a) \bmod n$;

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 $\overline{x_B} = 2^{\omega} + (x_B \& (2^{\omega} - 1))$, where $\omega = \lceil (\lceil \log_2(n) \rceil /2) \rceil - 1$;

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

A8: computes $K_A = K(x_\alpha || y_\alpha || Z_A || Z_B, L_k)$;

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), computes $W_1 = H(0x03 || y_\alpha || H(x_\alpha || Z_A || Z_B || x_A || y_A || x_B || y_B))$, and verifies $W_1 \overset{?}{=} W_B$, if it returns false, user B fails in the agreement;

A10: (optional for key confirmation) computes $W_A = H(0x03 || y_\alpha || H(x_\alpha || Z_A || Z_B || x_A || y_A || x_B || y_B))$, and sends W_A to client B;

User B:

B10: (optional for key confirmation) computes $W_2 = H(0x03 || y_\beta || Hash(x_\beta || Z_A || Z_B || x_A || y_A || x_B || y_B))$, verifies whether $W_2 \overset{?}{=} W_A$, and if it returns false, user A fails in the agreement.

At this point, the entire SM2 key exchange protocol 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 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.

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_k 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

User A:

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

User B:

B2: computes $D_{d_B}(E_{P_B}(r_A || Se || ID_A)$ to obtain r_A, Se, ID_A, and computes $H(r_A || Se || ID_A)$

B3: verifies $D_{P_A}(E_{d_A}(H(r_A || Se || ID_A))) \overset{?}{=} H(r_A || Se || ID_A)$, unsuccessful verification leads to authentication failure between user A and B and session termination.

B4: computes Replay $= E_{P_A}(r_B || Se || ID_B) || E_{d_B}(H(r_B || Se || ID_B))$, sends Replay to user A

User A:

A3: computes $D_{d_A}(E_{P_A}(r_B || Se || ID_B))$ to obtain r_B, Se, ID_B, and computes $H(r_B || Se || ID_B)$

A4: verifies $D_{P_B}(E_{d_B}(H(r_B || Se || ID_B))) \overset{?}{=} H(r_B || Se || ID_B)$, unsuccessful verification leads to authentication failure between user A and B, session termination.

2) KEY AGREEMENT

After the authentication phase, both user A and B should implement the following key agreement.

User A:

A5: selects P_1 as elliptic curve point $P_1 = (x_{P_1}, y_{P_1})$; P_1 is not an infinity point and satisfies the elliptic curve equation according to the definition given in Section II.A.1)

User B:

B5: selects P_2 as elliptic curve point $P_2 = (x_{P_2}, y_{P_2})$, P_2 is not an infinity point and the satisfies elliptic curve equation according to the definition given in Section II.A.1)

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 $\overline{x_{P_2}} = 2^{\omega} + (x_{P_2} \& (2^{\omega} - 1))$, where $\omega = \lceil (\lceil \log_2(n) \rceil /2) \rceil - 1$;

B7: computes $t_B = (d_B + \overline{x_{P_2}} \cdot r_B) \bmod n$;

B8: takes out the domain element x_{P_1} from P_1, converts the data type of x_{P_1} to an integer according to the function given in Section II.B.7), computes $\overline{x_{P_1}} = 2^{\omega} + (x_{P_1} \& (2^{\omega} - 1))$, where $\omega = \lceil (\lceil \log_2(n) \rceil /2) \rceil - 1$;

B9: computes elliptic curve point $V = [h \cdot t_B](P_1 + [\overline{x_{P_1}}]P_1) = (x_V, y_V)$; 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 the method given in Section II.B.5) and Section II.B.4);

B10: computes $K_B = K(x_V || y_V || Z_A || Z_B || r_A || r_B, L_k)$;

User A:

A6: takes out the domain element x_{P_1} from P_1, converts the data type of x_{P_1} to an integer according to the method given in the Section 2.B.7), computes $\overline{x_{P_1}} = 2^{\omega} + (x_{P_1} \& (2^{\omega} - 1))$, where $\omega = \lceil (\lceil \log_2(n) \rceil /2) \rceil - 1$;

A5: computes $t_A = (d_A + \overline{x_{P_1}} \cdot r_A) \bmod n$;

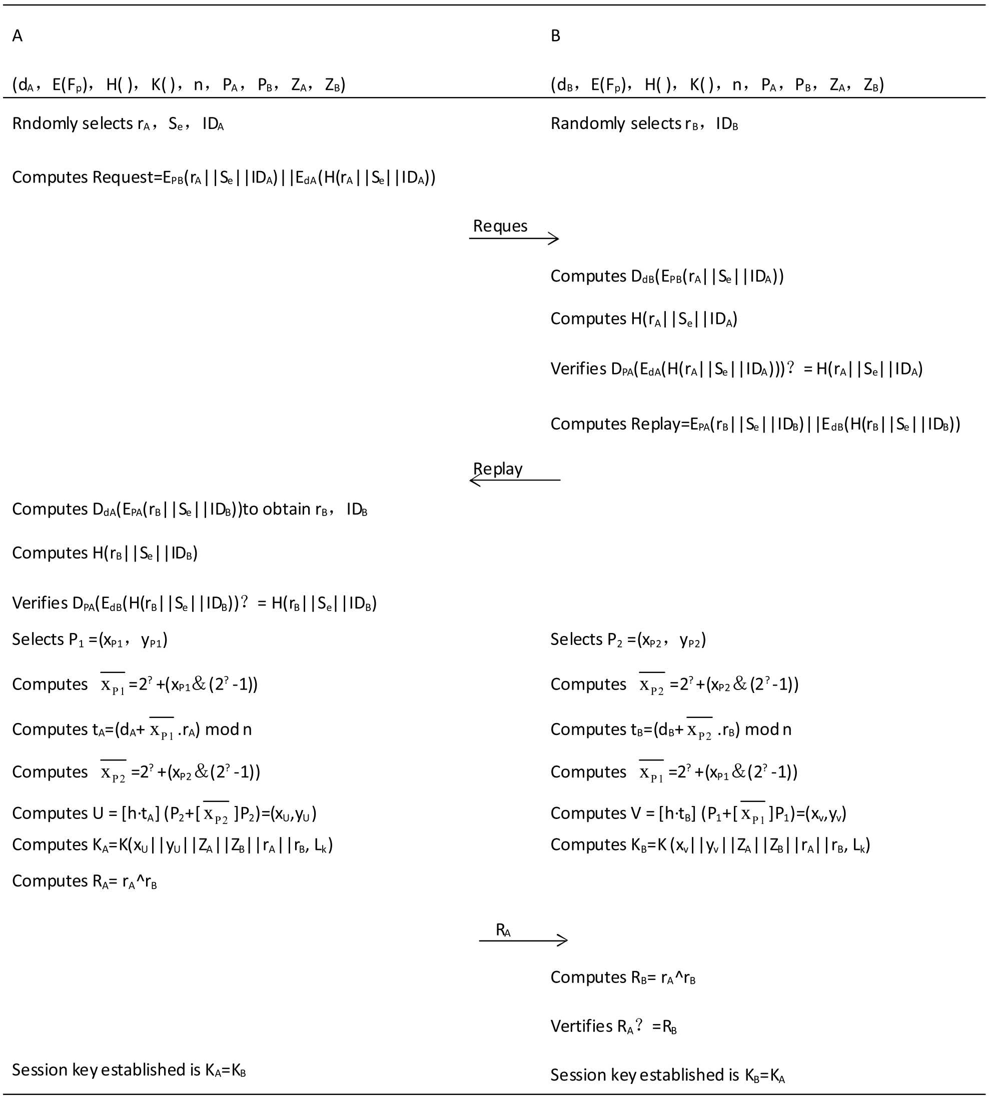

A
B
$(d_A, E(F_p), H(\,), K(\,), n, P_A, P_B, Z_A, Z_B)$
$(d_B, E(F_p), H(\,), K(\,), n, P_A, P_B, Z_A, Z_B)$
Rndomly selects r_A, S_e, ID_A
Randomly selects r_B, ID_B
Computes Request=$E_{PB}(r_A||S_e||ID_A)||E_{dA}(H(r_A||S_e||ID_A))$
Reques
Computes $D_{dB}(E_{PB}(r_A||S_e||ID_A))$
Computes $H(r_A||S_e||ID_A)$
Verifies $D_{PA}(E_{dA}(H(r_A||S_e||ID_A)))? = H(r_A||S_e||ID_A)$
Computes Replay=$E_{PA}(r_B||S_e||ID_B)||E_{dB}(H(r_B||S_e||ID_B))$
Replay
Computes $D_{dA}(E_{PA}(r_B||S_e||ID_B))$to obtain r_B, ID_B
Computes $H(r_B||S_e||ID_B)$
Verifies $D_{PA}(E_{dB}(H(r_B||S_e||ID_B))? = H(r_B||S_e||ID_B)$
Selects $P_1 = (x_{P1}, y_{P1})$
Selects $P_2 = (x_{P2}, y_{P2})$
Computes $\overline{x_{P1}} = 2^? + (x_{P1} \& (2^? - 1))$
Computes $\overline{x_{P2}} = 2^? + (x_{P2} \& (2^? - 1))$
Computes $t_A = (d_A + \overline{x_{P1}} . r_A) \bmod n$
Computes $t_B = (d_B + \overline{x_{P2}} . r_B) \bmod n$
Computes $\overline{x_{P2}} = 2^? + (x_{P2} \& (2^? - 1))$
Computes $\overline{x_{P1}} = 2^? + (x_{P1} \& (2^? - 1))$
Computes $U = [h \cdot t_A]\,(P_2 + [\overline{x_{P2}}]P_2) = (x_U, y_U)$
Computes $V = [h \cdot t_B]\,(P_1 + [\overline{x_{P1}}]P_1) = (x_v, y_v)$
Computes $K_A = K(x_U||y_U||Z_A||Z_B||r_A||r_B, L_k)$
Computes $K_B = K(x_v||y_v||Z_A||Z_B||r_A||r_B, L_k)$
Computes $R_A = r_A \wedge r_B$
R_A
Computes $R_B = r_A \wedge r_B$
Vertifies $R_A ? = R_B$
Session key established is $K_A = K_B$
Session key established is $K_B = K_A$

FIGURE 3. Proposed scheme.

A6: 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 $\overline{x_{P_2}} = 2^{\omega} + (x_{P_2} \& (2^{\omega} - 1))$, where $\omega = \lceil (\lceil \log_2(n) \rceil / 2) \rceil - 1$;

A7: computes elliptic curve point $U = [h \cdot t_A](P_2 + [\overline{x_{P_2}}]P_2) = (x_U, y_U)$, If U is an infinity point, user B fails in the agreement, unless the data type of x_U, y_U is converted to a bit string according to the method given in Section II.B.5) and Section II B.4);

A8: computes $K_A = K(x_U||y_U||Z_A||Z_B||r_A||r_B, L_k)$;

A9: computes $R_A = r_A \wedge r_B$ and sends R_A to B

User B:

B11: computes $R_B = r_A \wedge r_B$ verifies $R_A \underline{?} R_B$. Unsuccessful verification leads to agreement failure and session termination between user A and B.

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 $D_{P_A}(E_{D_A}(H(r_A||Se||ID_A)))\overset{?}{=} H(D_{d_B}(E_{P_B}(r_A||Se||ID_A)))$. 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 $D_{P_B}(E_{D_B}(H(r_B||Se||ID_B)))\overset{?}{=} H(D_{d_A}(E_{P_A}(r_B||Se||ID_B)))$. 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 δ replays a former message from user A, user B can identify and check the recentness/freshness of random number r_A by verifying $D_{P_A}(E_{d_A}(H(r_A||Se||ID_A))\overset{?}{=} H(r_A||Se||ID_A)$. Therefore, if δ 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 [23]. 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 $D_{P_A}(E_{D_A}(H(r_A||Se||ID_A)))\overset{?}{=} H(D_{d_B}(E_{P_B}(r_A||Se||ID_A)))$. 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 δ cannot pass the authentication process.

D. MESSAGE INJECTION ATTAK

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 δ cannot obtain the shared session secret key. It is not possible to inject malicious information, since δ cannot correctly decrypt the network packets without the shared session secret key. If δ 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 [24].

E. MAN-IN-THE-MIDDLE ATTACK

The adversary δ 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 δ 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, δ cannot make a correct reply to user A. Thus, the proposed scheme protects against MITM attacks.

TABLE 2. Comparison of security characteristics.

Scheme	Proposed	[1]	[19]	[25]	[13]
Key agreement	Yes	Yes	Yes	Yes	Yes
Mutual authentication	Yes	No	Yes	Yes	Yes
Replay attack	Yes	Yes	Yes	Yes	Yes
Impersonation attack	Yes	No	Yes	Yes	Yes
Information injection	Yes	Yes	No	No	Yes
Shared key update	Yes	Yes	Yes	Yes	Yes
MIMT	Yes	Yes	Yes	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes	Yes

F. EAVESDROPPING FOR A LONG TIME

Suppose that the attacker δ 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's refreshing and updating in a short-time. As a result, once the shared session key becomes effective, δ 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 2, and shows that the proposed scheme has more security characteristics.

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

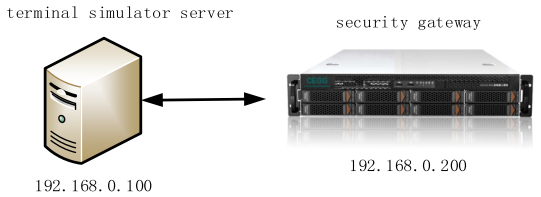


FIGURE 4. Network topology diagram of the experimental evaluation.

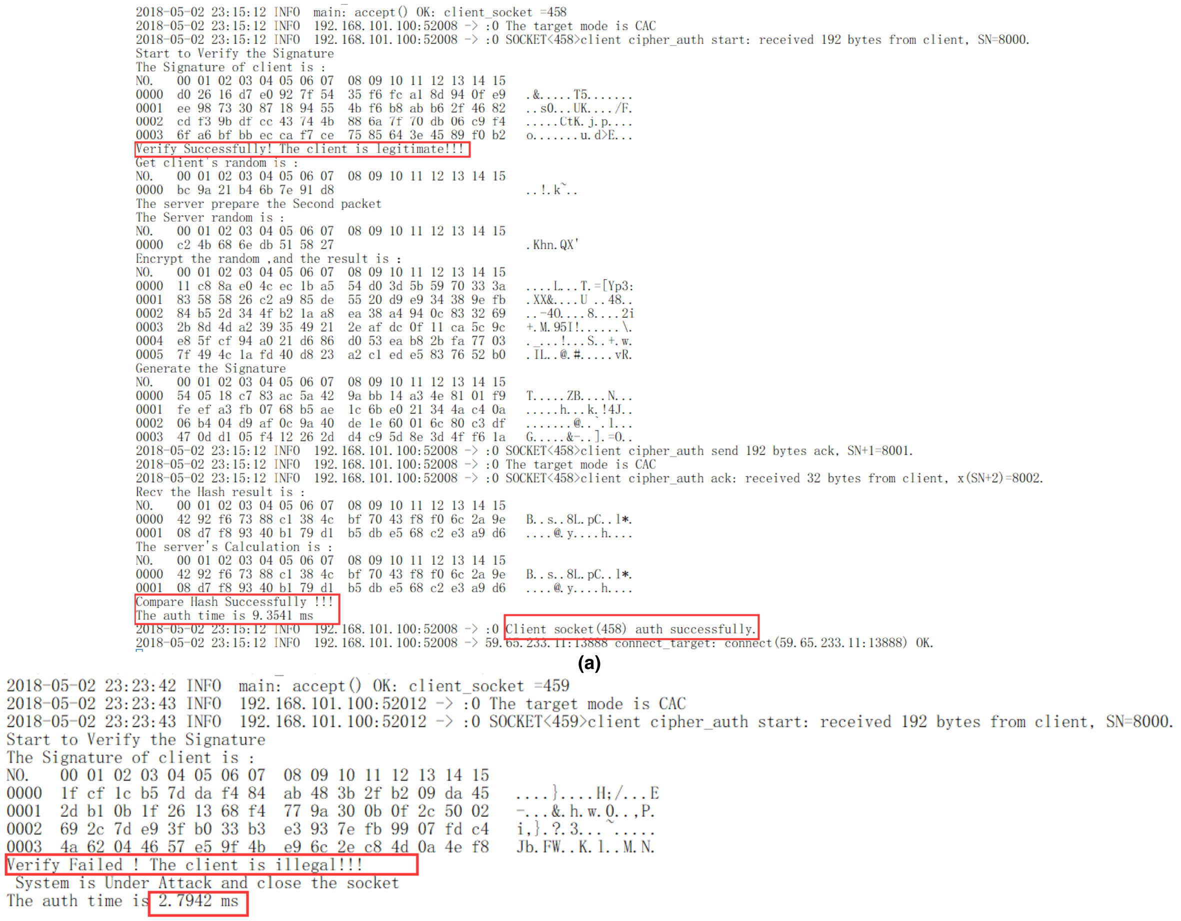
separately. The related primitive operations notations are listed in Table 3. The execution time of the primitive operations is presented in [26]. Furthermore, the communication overhead in our scheme is evaluated by selecting a 16-bit-long ID, an 8-bit random number r, an 8-bit session identifier *Se*, a 64-bit public keys and a 160-bit ECC operation size. Table 4 shows that our scheme outperforms the other similar schemes both in computational complexity and communication overhead terms.

TABLE 3. Notations.

Notations	Description
T_{H}	Execution time for operation of the hash function
T_{DS}	Execution time for operation of the digital signature
T_{VS}	Execution time for operation of the verification signature
T_{AD}	Execution time for operation of the asymmetric key decryption
T_{AE}	Execution time for operation of the asymmetric key encryption
T_{ESM}	Execution time for operation of the ECC scalar multiplication
T_{ME}	Execution time for operation of the modular exponentiation
T_{PA}	Execution time for operation of the point addition
T_{SE}	Execution time for operation of the symmetric key encryption
T_{SD}	Execution time for operation of the symmetric key decryption
T_{HMAC}	Execution time for operation of the hash-based message authentication code(HMAC)operation

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



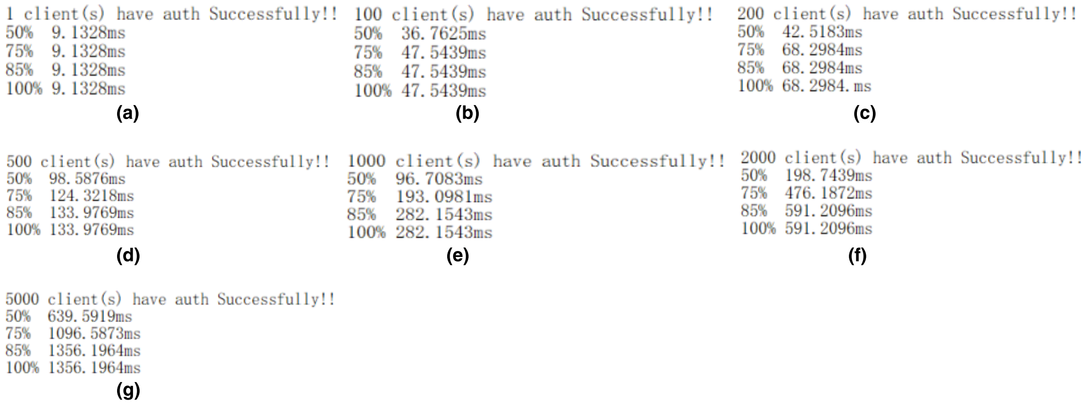
```
2018-05-02 23:15:12 INFO  main: accept() OK: client_socket =458
2018-05-02 23:15:12 INFO  192.168.101.100:52008 -> :0 The target mode is CAC
2018-05-02 23:15:12 INFO  192.168.101.100:52008 -> :0 SOCKET<458>client cipher_auth start: received 192 bytes from client, SN=8000.
Start to Verify the Signature
The Signature of client is :
NO.    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15
0000   d0 26 16 d7 e0 92 7f 54  35 f6 fc a1 8d 94 0f e9    .&.....T5.......
0001   ee 98 73 30 87 18 94 55  4b f6 b8 ab b6 2f 46 82    ..s0...UK..../F.
0002   cd f3 9b df cc 43 74 4b  88 6a 7f 70 db 06 c9 f4    .....CtK.j.p....
0003   6f a6 bf bb ec ca f7 ce  75 85 64 3e 45 89 f0 b2    o.......u.d>E...
Verify Successfully! The client is legitimate!!!
Get client's random is :
NO.    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15
0000   bc 9a 21 b4 6b 7e 91 d8                             ..!.k~..
The server prepare the Second packet
The Server random is :
NO.    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15
0000   c2 4b 68 6e db 51 58 27                             .Khn.QX'
Encrypt the random ,and the result is :
NO.    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15
0000   11 c8 8a e0 4c ec 1b a5  54 d0 3d 5b 59 70 33 3a    ....L...T.=[Yp3:
0001   83 58 58 26 c2 a9 85 de  55 20 d9 e9 34 38 9e fb    .XX&....U ..48..
0002   84 b5 2d 34 4f b2 1a a8  ea 38 a4 94 0c 83 32 69    ..-40....8....2i
0003   2b 8d 4d a2 39 35 49 21  2e af dc 0f 11 ca 5c 9c    +.M.95I!......\.
0004   e8 5f cf 94 a0 21 d6 86  d0 53 ea b8 2b fa 77 03    ._...!...S..+.w.
0005   7f 49 4c 1a fd 40 d8 23  a2 c1 ed e5 83 76 52 b0    .IL..@.#.....vR.
Generate the Signature
NO.    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15
0000   54 05 18 c7 83 ac 5a 42  9a bb 14 a3 4e 81 01 f9    T.....ZB....N...
0001   fe ef a3 fb 07 68 b5 ae  1c 6b e0 21 34 4a c4 0a    .....h...k.!4J..
0002   06 b4 04 d9 af 0c 9a 40  de 1e 60 01 6c 80 c3 df    .......@..`.l...
0003   47 0d d1 05 f4 12 26 2d  d4 c9 5d 8e 3d 4f f6 1a    G.....&-..].=O..
2018-05-02 23:15:12 INFO  192.168.101.100:52008 -> :0 SOCKET<458>client cipher_auth send 192 bytes ack, SN+1=8001.
2018-05-02 23:15:12 INFO  192.168.101.100:52008 -> :0 The target mode is CAC
2018-05-02 23:15:12 INFO  192.168.101.100:52008 -> :0 SOCKET<458>client cipher_auth ack: received 32 bytes from client, x(SN+2)=8002.
Recv the Hash result is :
NO.    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15
0000   42 92 f6 73 88 c1 38 4c  bf 70 43 f8 f0 6c 2a 9e    B..s..8L.pC..l*.
0001   08 d7 f8 93 40 b1 79 d1  b5 db e5 68 c2 e3 a9 d6    ....@.y....h....
The server's Calculation is :
NO.    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15
0000   42 92 f6 73 88 c1 38 4c  bf 70 43 f8 f0 6c 2a 9e    B..s..8L.pC..l*.
0001   08 d7 f8 93 40 b1 79 d1  b5 db e5 68 c2 e3 a9 d6    ....@.y....h....
Compare Hash Successfully !!!
The auth time is 9.3541 ms
2018-05-02 23:15:12 INFO  192.168.101.100:52008 -> :0 Client socket(458) auth successfully.
2018-05-02 23:15:12 INFO  192.168.101.100:52008 -> 59.65.233.11:13888 connect_target: connect(59.65.233.11:13888) OK.
```

(a)

```
2018-05-02 23:23:42 INFO  main: accept() OK: client_socket =459
2018-05-02 23:23:43 INFO  192.168.101.100:52012 -> :0 The target mode is CAC
2018-05-02 23:23:43 INFO  192.168.101.100:52012 -> :0 SOCKET<459>client cipher_auth start: received 192 bytes from client, SN=8000.
Start to Verify the Signature
The Signature of client is :
NO.    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15
0000   1f cf 1c b5 7d da f4 84  ab 48 3b 2f b2 09 da 45    ....}....H;/...E
0001   2d b1 0b 1f 26 13 68 f4  77 9a 30 0b 0f 2c 50 02    -...&.h.w.0..,P.
0002   69 2c 7d e9 3f b0 33 b3  e3 93 7e fb 99 07 fd c4    i,}.?.3...~.....
0003   4a 62 04 46 57 e5 9f 4b  e9 6c 2e c8 4d 0a 4e f8    Jb.FW..K.l..M.N.
Verify Failed ! The client is illegal!!!
 System is Under Attack and close the socket
The auth time is 2.7942 ms
```

(b)

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.



```
1 client(s) have auth Successfully!!
50%  9.1328ms
75%  9.1328ms
85%  9.1328ms
100% 9.1328ms
```

(a)

```
100 client(s) have auth Successfully!!
50%  36.7625ms
75%  47.5439ms
85%  47.5439ms
100% 47.5439ms
```

(b)

```
200 client(s) have auth Successfully!!
50%  42.5183ms
75%  68.2984ms
85%  68.2984ms
100% 68.2984.ms
```

(c)

```
500 client(s) have auth Successfully!!
50%  98.5876ms
75%  124.3218ms
85%  133.9769ms
100% 133.9769ms
```

(d)

```
1000 client(s) have auth Successfully!!
50%  96.7083ms
75%  193.0981ms
85%  282.1543ms
100% 282.1543ms
```

(e)

```
2000 client(s) have auth Successfully!!
50%  198.7439ms
75%  476.1872ms
85%  591.2096ms
100% 591.2096ms
```

(f)

```
5000 client(s) have auth Successfully!!
50%  639.5919ms
75%  1096.5873ms
85%  1356.1964ms
100% 1356.1964ms
```

(g)

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 100 terminals. (c) The execution time of 200 terminals. (d) The execution time of 500 terminals. (e) The execution time of 1000 terminals. (f) The execution time of 2000 terminals. (g) The execution time of 5000 terminals.

TABLE 4. Performance evaluation.

Schemes	Computation cost	Communication cost
[13]	$14T_H$=26.5123ms	960 bits
[1]	$5T_H+2T_{SD}$=10.4968ms	640 bits
[25]	$T_{AE}+T_{AD}+2T_H+2T_{ESM}+2T_{SE}+2T_{SD}$=19.8658ms	608 bits
[19]	$5T_{SM}+5T_H+1T_{PA}$=11.1703ms	576 bits
Proposed	$2T_{SE}+2T_{DS}+2T_{SD}+2T_{VS}+2T_{ME}+2T_{ESM}$=9.1328ms	422 bits

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 CentOS6.6_ x64. Figure 4 shows the 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.

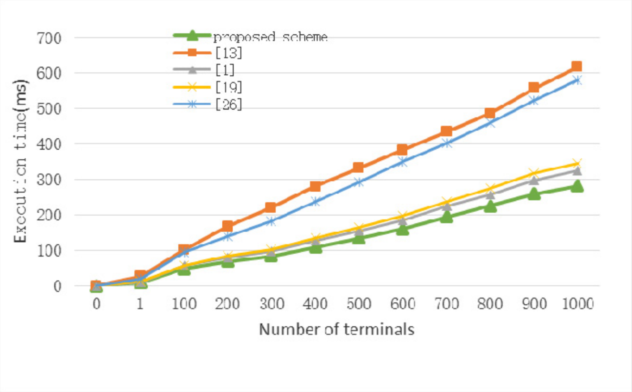


FIGURE 7. The execution time in related schemes.

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], and [25]. According to Figure 7, with the increasing number of STs, the execution time markedly increases in the scheme of [13], and [25] . 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], and [25].

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

[1] N. Saxena and S. Grijalva, "Dynamic secrets and secret keys based scheme for securing last mile smart grid wireless communication," *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1482–1491, Jun. 2017.

[2] W. Wang and Z. Lu, "Cyber security in the Smart Grid: Survey and challenges," *Computer Netw.*, vol. 57, no. 5, pp. 1344–1371, 2013.

[3] R. Khan, K. Mclaughlin, D. Laverty, and S. Sezer, "Design and implementation of security gateway for synchrophasor based real-time control and monitoring in smart grid," *IEEE Access*, vol. 5, pp. 11626–11644, 2017.

[4] X. Li, J. Niu, S. Kumari, F. Wu, and K.-K. R. Choo, "A robust biometrics based three-factor authentication scheme for global mobility networks in smart city," *Future Gener. Comput. Syst.*, vol. 83, pp. 607–618, Jun. 2018.

[5] Z. Ma, "CPSec DLP: Kernel-level content protection security system of data leakage prevention," *Chin. J. Electron.*, vol. 26, no. 4, pp. 827–836, 2017.

[6] Y. Jie, L. Yu, C. Li-Yun, and N. Wei, "A SM2 elliptic curve threshold signature scheme without a trusted center," *Ksii Trans. Internet Inf. Syst.*, vol. 10, no. 2, pp. 897–913, 2016.

[7] W. Zhaohui and Z. Zhenfeng, "Overview of SM2 elliptic curve public key cryptography," (in Chinese), *Inf. Secur. Res.*, vol. 2, no. 11, pp. 972–982, 2016.

[8] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2018.

[9] M. Bayat, M. Beheshti, and A. R. Aref, "A secure and efficient chaotic maps based authenticated key-exchange protocol for smart grid," *Wireless Pers. Commun.*, vol. 97, no. 2, pp. 2551–2579, 2017.

[10] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[11] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in Internet of Things environments," *J. Netw. Comput. Appl.*, vol. 103, pp. 194–204, Feb. 2018.

[12] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 998–1010, 4th Quart., 2012.

[13] L. Yan, Y. Chang, and S. Zhang, "A lightweight authentication and key agreement scheme for smart grid," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 2, 2017.

[14] Y. Chen, J.-F. Martínez, P. Castillejo, and L. López, "An anonymous authentication and key establish scheme for smart grid: FAuth," *Energies*, vol. 10, no. 9, p. 1354, 2017.

[15] O. R. M. Boudia, S. M. Senouci, and M. Feham "Elliptic curve-based secure multidimensional aggregation for smart grid communications," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7750–7757, Dec. 2017.

[16] D. Li, Z. Aung, J. R. Williams, and A. Sanchez, "Efficient and fault-diagnosable authentication architecture for AMI in smart grid," *Secur. Commun. Netw.*, vol. 8, no. 4, pp. 598–616, 2015.

[17] Q. Li and G. Cao, "Multicast authentication in the smart grid with one-time signature," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 686–696, Dec. 2011.

[18] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 675–685, Dec. 2011.

[19] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Gener. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.

[20] L. Zichen, "(Dean's Office, Beijing Inst. of Graphic Commun., Beijing, China); Liu Boya; Wang Peidong; Yang Yatao Source," (in Chinese) *Comput. Eng., v*, vol. 43, no. 6, pp. 97–100, 104, Jun. 2017.

[21] A. Yang, J. Nam, M. Kim, and K.-K. R. Choo, "Provably-secure (Chinese government) SM2 and simplified SM2 key exchange protocols," *Sci. World J.*, vol. 2014, Sep. 2014, Art. no. 825984.

[22] C. I. Fan, S.-Y. Huang, W. Artan, "Design and implementation of privacy preserving billing protocol for smart grid," *J. Supercomput.*, vol. 66, no. 2, pp. 841–862, 2013.

[23] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K.-K. R. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Comput. Netw.*, vol. 129, pp. 429–443, Dec. 2017.

[24] D. Chattaraj, M. Sarma, and A. K. Das, "A new two-server authentication and key agreement protocol for accessing secure cloud services," *Comput. Netw.*, vol. 131, pp. 144–164, Feb. 2018.

[25] L. Zhang, S. Tang, and H. Luo, "Elliptic curve cryptography-based authentication with identity protection for smart grids," *Plos One*, vol. 11, no. 3, p. e0151253, 2016.

[26] H. H. Kilinc and T. Yanik, "A Survey of SIP Authentication and Key Agreement Schemes," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1005–1023, 2nd Quart., 2014.

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.

LINPING SU received a B.S. degree in computer science and technology from the Tianjin University, Tianjin, China, in 1989.

She is a 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.