Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.
Vol. **68**, *2*, pp. 218–223, Bucarest, 2023

AUTHENTICATED KEY AGREEMENT SCHEME BASED ON BLOCKCHAIN FOR AMI COMMUNICATION SECURITY

ZHENDONG LIU[1], LIANG MENG[1], QINGYUAN ZHAO[1], FEI LI[1], MANRUI SONG[1], YUZHI JIAN[2], HONGLIANG TIAN[2]

Keywords: Authenticated key agreement scheme; Consensus algorithm; Blockchain; Advanced measurement infrastructure security.

Security is the basis for the normal operation of advanced measurement infrastructure (AMI). As an important part of key management scheme, key establishment is indispensable in meeting AMI communication security requirements. Most proposed key management schemes rely on a trusted third party (TTP). Once there is a problem with TTP, the security of these schemes will be greatly reduced. Furthermore, the data concentrators (DCs) in traditional AMI architectures all manage smart meters (SMs) in their respective regions, and the lack of interaction between the DCs exposes a serious single point of failure. To alleviate these problems, we propose a blockchain-based authenticated key agreement scheme to secure the communication of AMI. In this scheme, the blockchain comprises DCs as network nodes that interact with the SMs. The proposed key agreement and distributed consensus protocol ensure the authenticity and validity of the communication content without relying on TTP. We analyse the resistance of the proposed protocol to multiple known attacks and evaluate its performance. The proposed protocol has higher security or better performance than other schemes.

1. INTRODUCTION

With the growing demand for interaction between power companies and users, automatic meter reading (AMR) technology has evolved into AMI. AMI is not a single technology but an infrastructure integrating multiple technology configurations. This AMI, which consists of SMs, DCs, measurement data Management systems (MDMS), and communication network between different levels of facilities, realizes two-way communication between users and power companies, thus bringing unlimited opportunities for both parties [1].

The use of wireless communication networks and the installation of exposed facilities determine the vulnerability of AMI in the face of physical and network attacks [2]. Like other network physical systems, AMI must also follow security primitives such as confidentiality, integrity, availability, and non-repudiation. To meet these security requirements, key encryption is usually used. Therefore, the security problem of the system can be transformed into a key management problem [3]. Key management usually includes key establishment (in this paper, key establishment and key agreement are not distinguished), key refresh, key distribution, key storage, etc., of which key establishment is the basic element [4]. Key establishment involves two or more entities establishing a session key. The key establishment method used in this paper is key agreement [5].

Diffie-Hellman protocol is the first key agreement protocol based on asymmetric encryption [6], and its security is based on the complexity of the Diffie-Hellman problem and discrete logarithm problem. At present, many key agreement protocols are based on the idea of Diffie-Hellman. Unfortunately, these protocols do not have an authentication function. In this regard, many researchers such as Menezes, Qu, and Vanstone try to add authentication and key confirmation functions to the Diffie-Hellman protocol, namely the MQV protocol [7]. This two-pass protocol provides mutual implicit key verification and has known key security, forward secrecy, key control, and other characteristics. Authenticated key agreement (AKA) is an enhanced key establishment method that can complete the verification of key materials while carrying out key establishment [8]. AKA can be realized by public-key infrastructure (PKI) [9] or identity-based encryption methods. Since PKI-based schemes have a large amount of certificate management overhead, the identity-based encryption method will be more suitable for AMI. In addition, most of the existing key establishment schemes are based on TTP. The existence of a single point of failure and trust crisis makes the communication security of AMI face great challenges.

We propose a blockchain-based authenticated key agreement scheme, which uses the DCs as the network node to construct the blockchain network and ensure the high reliability of the system while making full use of AMI component resources.

2. PROPOSED SCHEME

In this section, we give a detailed description of the proposed scheme. This scheme is mainly used for identity authentication between AMI components and subsequent secure communication, especially in the absence of TTP. The symbols used in this scheme and their meanings are shown in Table 1.

Table 1
Notations and their meanings.

Notation	Meaning
ID_{SM} , ID_{DC} , ID_{MDMS}	Unique ID of the corresponding device
P_{SM} , P_{DC} , P_{MDMS}	Public key of corresponding device
S_{SM} , S_{DC} , S_{MDMS}	Public key of corresponding device
L_{DC}	Leader node selected from DCs
$En(\cdot)_{P_{DC}}$, $En(\cdot)_K$, $En(\cdot)_{P_{pub}}$	Encrypt with P_D , K , P_{pub} , respectively

2.1 PROPOSED SCHEME

As shown in Fig. 1 (b), AMI key management architecture is mainly composed of SMs, DCs, MDMS, and communication networks (CNs) between components [10].

1) SM: a solid-state programmable measuring device that has the functions of power consumption measurement, bidirectional multi-rate measurement, and bidirectional data communication of multiple data transmission modes, *etc*. In addition, it enables demand-side management to be realized because timely information feedback has been proven to encourage consumers to reduce power consumption.

[1] State Grid Benxi Electric Power Supply Company, Benxi 117000, China. E-mails:1351129437@qq.com, 22961326@qq.com, 846855925@qq.com, 494779066@qq.com, 1351129437@qq.com
[2] Jilin Northeast Electric Power University Science and Technology Development Co., Ltd., Jilin 132000, China. E-mails: jian_yuzhi@163.com, hltian@foxmail.com

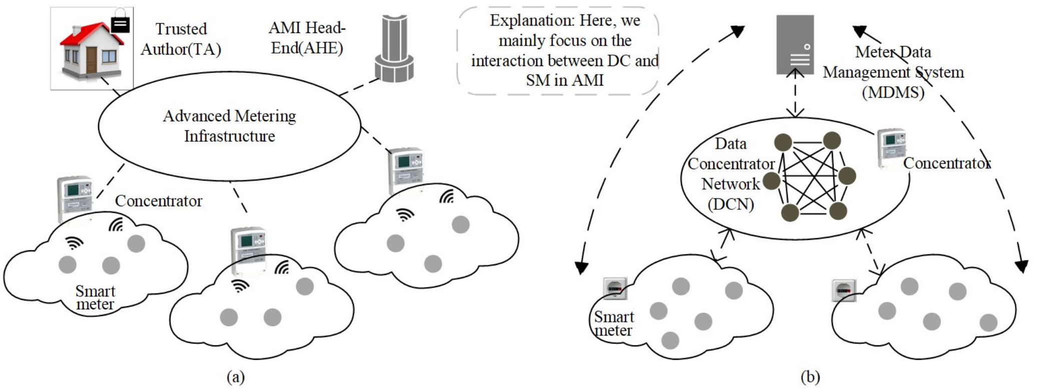
Trusted Author(TA)
AMI Head-End(AHE)
Advanced Metering Infrastructure
Concentrator
Smart meter
(a)
Explanation: Here, we mainly focus on the interaction between DC and SM in AMI
Meter Data Management System (MDMS)
Data Concentrator Network (DCN)
Concentrator
Smart meter
(b)

Fig. 1 – Different network architectures: a) traditional key management architecture; b) blockchain-based key management architecture.

2) DC: a device used as a client gateway. It realizes protocol conversion and communication between two heterogeneous networks, such as home LAN and WAN. In addition, it can also aggregate and forward measurement and management data.

3) MDMS: a database for long-term power data storage and management, which can interact with other management systems, including power outage management system, consumer management system for managing utility billing and user information, and distribution management system for providing power quality management and load forecasting based on measured data [11].

4) CN: it includes a home area network connecting smart meters, smart homes, and other control devices, a neighborhood network and wide area network supporting two-way communication between customers and power companies, and a P2P network for information sharing between network nodes.

2.2 NETWORK ASSUMPTIONS

1) In the key material generation phase, the communication channel between MDMS and SM, DC is private and secure.

2) In the phase of mutual authentication and key agreement, the communication links among AMI components for information exchange are public and risky.

3) Each device has a unique ID that can be identified, such as ID_{SM}.

4) Only legitimate device components can have the parameters published by the system.

2.3 KEY MANAGEMENT SCHEME

Elliptic curve bilinear pairing is usually used to improve the security of existing key management schemes. However, considering that bilinear pairing is always defined on hypersingular elliptic curve groups with large parameters, the pairing time is often much longer than RSA [12]. Therefore, in this section, we will make full use of the elliptic curve cryptosystem to ensure the security of the key while avoiding complex operations such as bilinear pairing to a large extent to alleviate the problem of resource constraints of components (such as SM). Next, based on the idea of Mohammadali et al. [3], we propose a new blockchain-based key agreement scheme as follows :

2.3.1 SYSTEM INITIALIZATION

The grid administrator selects k as the system parameter, and then MDMS will do the following operations:

1. Select a prime number q of k bit length, then construct $\{F_q, E(F_q), G_q, P\}$ where G_q is a group of points on the elliptic curve $E(F_q)$, F_q is the finite field of the elliptic curve, P is the generator or base point with prime order q.

2. Select the master key $x \in Z_q^*$ to generate the public key $P_{pub} = xP \in E(F_q)$ of the system.

3. Select two hash functions $H_1 : \{0,1\}^* \times G_q \to G_q$, $H_2 : \{0,1\}^* \times G_q \to Z_q^*$.

4. Publish the system parameter tuple $\{F_q, E(F_q), G_q, P, P_{pub}, H_1, H_2\}$ and maintain the confidentiality of the master key x. The published parameter tuples will be safely embedded into SM and DC through physical media.

2.3.2 GENERATION OF KEY MATERIALS

The generation of DC key materials will follow the following steps:

1. DC generates a random number $r_{DC} \in Z_q^*$ and calculates $R_{DC} = r_{DC}P$. Then send $\{R_{DC}, ID_{DC}\}$ to MDMS.

2. After MDMS obtains the information, it calculates $y_{DC} = H_1(ID_{DC}, R_{DC}).x$ and returns y_{DC} to DC.

The generation of SM key materials will follow the following steps:

1. SM first generates a random number $r_{SM} \in Z_q^*$ and calculates $R_{SM} = r_{SM}P$. Then send ID_{SM} to MDMS.

2. MDMS calculates $y_{SM} = H_2(ID_{SM}, y_{DC}).x$, and then returns y_{SM} to SM,.

3. SM calculates $S_{SM} = y_{SM} + r_{SM}$ and uses it as its public key.

MDMS generates an information tuple: $U = \{ID_{SM}, ID_{DC}, y_{DC}\}$, and stores the tuple on the blockchain for subsequent calls.

Eventually, every SM will have $\{S_{SM}, R_{SM}, y_{SM}, r_{SM}\}$, every DC will have $\{y_{DC}, r_{DC}\}$, and DCN will have multiple U.

The generation process of key materials can be completed when the equipment leaves the factory or during the initial installation, because this process is relatively independent of the subsequent session key agreement process. In this way, on the one hand, the network overhead when the device is online can be reduced. On the other hand, SM and DC can independently complete the agreement of session key when MDMS is not online. The generation process of key material is shown in Fig. 2.

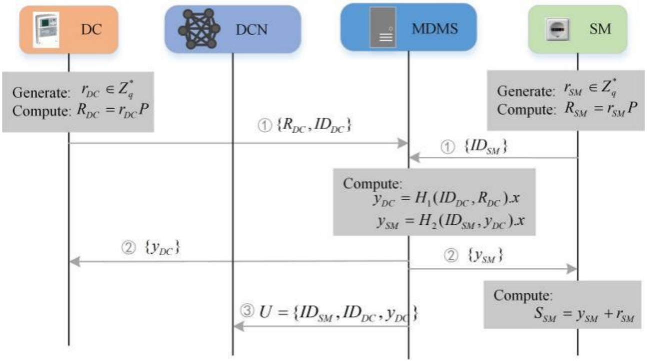
DC
DCN
MDMS
SM
Generate: $r_{DC} \in Z_q^*$
Compute: $R_{DC} = r_{DC}P$
Generate: $r_{SM} \in Z_q^*$
Compute: $R_{SM} = r_{SM}P$
① $\{R_{DC}, ID_{DC}\}$
① $\{ID_{SM}\}$
Compute:
$y_{DC} = H_1(ID_{DC}, R_{DC}).x$
$y_{SM} = H_2(ID_{SM}, y_{DC}).x$
② $\{y_{DC}\}$
② $\{y_{SM}\}$
Compute:
$S_{SM} = y_{SM} + r_{SM}$
③ $U = \{ID_{SM}, ID_{DC}, y_{DC}\}$

Fig. 2 – Generation of key materials.

2.3.3. AGREEMENT OF SESSION KEY

1. SM generates a random number $a \in Z_q^*$ and calculates $A = a + r_{SM}$. The difference from the traditional scheme is that the information tuple $\{A, ID_{SM}\}$ is sent to DCN instead of the corresponding DC.

2. After receiving the message sent from SM, DCN will perform the following steps:

2a) DCN uses algorithm 1 to select a DC from the nodes of the whole network as the leader node, L_{DC}.

2b) According to the received information tuple, the corresponding $\{A, U\}$ is sent to L_{DC}. Note that each ID_{SM} corresponds to one U.

2c) L_{DC} generates a random number $b \in Z_q^*$. Then calculate $T_M = AP, k_{DC \to SM} = (T_M + H_2(ID_{SM}, y_{DC})P_{pub}).b$, $T_{DC} = bP$ and $M_1 = H_1(0, k_{DC \to SM})$, and send $\{T_{DC}, ID_{DC}, M_1\}$ to SM.

3. SM calculates $k_{SM \to DC} = (S_{SM} + a)T_{DC}$ and $M_1^{'} = H_1(0, k_{SM \to DC})$. Then compare whether M_1 and $M_1^{'}$ are the same. If they are the same, L_{DC} authentication passes, and then set $K = H_1(ID_{SM} \| ID_{DC}, K_{SM \to DC})$ as the session key.

4. SM calculates $M_2 = H_1(1, k_{SM \to DC})$, and then returns M_2 to L_{DC}.

5. After receiving the message sent from SM, L_{DC} calculates $M_2^{'} = H_1(1, k_{DC \to SM})$. Then compare $M_2^{'}$ and M_2. if they are the same, SM authentication passes, and then set $K = H_1(ID_{SM} \| ID_{DC}, k_{DC \to SM})$ as the session key.

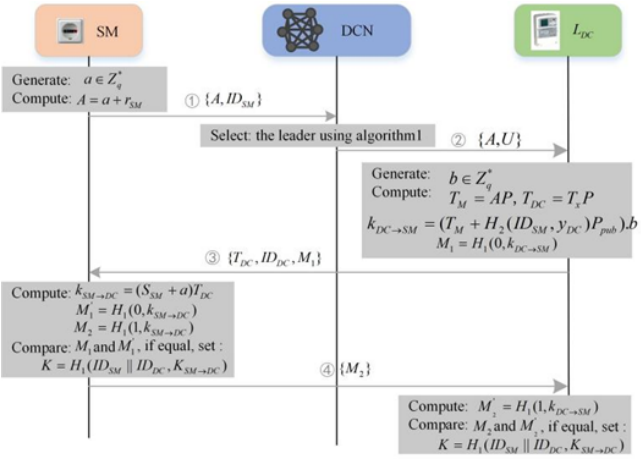
SM
DCN
L_{DC}
Generate: $a \in Z_q^*$
Compute: $A = a + r_{SM}$
① $\{A, ID_{SM}\}$
Select: the leader using algorithm1
② $\{A,U\}$
Generate: $b \in Z_q^*$
Compute: $T_M = AP, T_{DC} = T_x P$
$k_{DC \to SM} = (T_M + H_2(ID_{SM}, y_{DC})P_{pub}).b$
$M_1 = H_1(0, k_{DC \to SM})$
③ $\{T_{DC}, ID_{DC}, M_1\}$
Compute: $k_{SM \to DC} = (S_{SM} + a)T_{DC}$
$M_1^{'} = H_1(0, k_{SM \to DC})$
$M_2 = H_1(1, k_{SM \to DC})$
Compare: M_1 and $M_1^{'}$, if equal, set :
$K = H_1(ID_{SM} \| ID_{DC}, K_{SM \to DC})$
④ $\{M_2\}$
Compute: $M_2^{'} = H_1(1, k_{DC \to SM})$
Compare: M_2 and $M_2^{'}$, if equal, set :
$K = H_1(ID_{SM} \| ID_{DC}, K_{SM \to DC})$

Fig. 3 – Key agreement.

In the process of key agreement, if the authentication of both communication parties fails or the elected leader node fails, the above steps can be performed again. The key agreement steps are shown in Fig. 3.

Algorithm 1 Election of leader node (nodes can have three states: *Follower* , *Candidate* , *Leader*)

```
1  When $n > 3f + 1$, $DC_i \to Follower(i \in 1, 2, ..., n)$, where $f$ is the number of faulty nodes;
2  Set the tenure number to 0, that is, $TN_{DC_i} = 0(i \in 1, 2, ..., n)$;
3  Set the initial number of votes to 0, that is, $N_v = 0$;
4  Start timing, and represents as $Timer$;
5  Set a time threshold, that is, $T_{out}$;
6  **While** $Timer > T_{out}$ **do**
7    $Follower \to Candidate$ ;
8    $TN + 1$;
9    $Timer$ return to zero and restart the timing;
10   $N_v + 1$;
11   Send the voting request to other nodes and wait for the response;
12   **if** received response from other nodes **then**
13     Calculate the cumulative number of votes $N_v$;
14     **if** $N_v > n/2 + 1$, where $n$ is the number of nodes **then**
15       $Candidate \to Leader$ ;
16     **end if**
17   **else** (the leader node has been determined)
18     $Candidate \to Follower$ ;
19   **else**
20     Repeat steps 7-11 to start a new election;
21   **end if**
22 **end while**
```

2.3.4 SIGNATURE AND VERIFICATION

At this stage, SM will send metering data m to L_{DC} safely and frequently, and each communication authentication will be permanently recorded in the form of transactions on the tamper proof ledger.

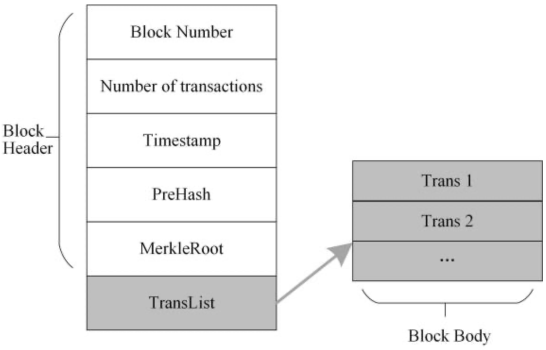

Fig. 4 – Block structure.

1. SM calculates the measurement data ciphertext, *i.e.*, $En(m)_K$, and then sends it to L_{DC}.

2. L_{DC} uses the session key K to obtain the measurement data m.

3. L_{DC} generates a signature, expressed as: $Sig_{DC} = En(ID_{SM}, ID_{DC})_{P_{pub}}$; Then generate a transaction, which can be expressed as: $Trans = \{En(m)_{P_{pub}}, U, T_s, Sig_{DC}\}$ where T_s is the timestamp generated by the transaction.

4. L_{DC} package transactions and generate blocks. The block structure is shown in Fig. 4. Then broadcast the block to other nodes of the whole network, and reach a consensus of the whole network through algorithm 2.

5. After receiving the new block, each node will verify the block parameters shown in Fig. 4 to ensure the authenticity and effectiveness of the transaction content.

6. Link the verified new block to the longest blockchain in the whole network to form the latest blockchain.

Algorithm 2 Consistency verification

```
Each $Follower$ receives a block from $Leader$:
  $B = \{PreHash, MerkleRoot, TimeStamp, Trans\}$, and verifies the block. The specific steps are as follows:
for each $Follower$ in DC do
    Extract $ID_{SM}$ and $ID_{DC}$ contained in $U$ in $Trans$, and calculate $Sig_{DC}^{'}$ according to $P_{pub}$ embedded locally;
    Extract all $Trans$ in Block $B$, and calculate $MerkleRoot'$ according to Merkle tree structure;
    According to the historical operation of the system, determine the minimum delay $T_{\min}$ and the maximum delay $T_{\max}$ of P2P network operation;
    if $Sig_{DC}^{'} = Sig_{DC}$, $MerkleRoot' = MerkleRoot$, $TimeStamp + T_{\min} \le T_{now} \le TimeStamp + T_{\max}$ then
        When the node completes the verification of the block and passes, it will reply to the $Leader$ node;
    end if
end for
$Leader$ initialize a parameter $V$ to count the number of replies received from $Follower$;
Each time $Leader$ receives a reply, $V = V + 1$;
if $V > 2f + 1$, Where $f$ is the number of failed nodes then
    $Leader$ sends committed to $Follower$;
    All the $Follower$ which receive the committed add the block to the blockchain.
end if
```

3. SECURITY ANALYSIS AND COMPARISON

In this section, we analysed the security of the proposed scheme and compared it with other schemes, as shown in Table 2.

Table 2

Resilience to known attacks.

	NIKE^{+} [3]	Sha [13]	SKM^{+} [11]	A-Mood [14]	LAKA [15]	Our
Resist replay attack	√	√	√	√	√	√
Resist Impersonation attack	√	√	√	√	√	√
Resist MITM attack	√	√	√	-	√	√
Resist desynchronization attack	√	-	√	√	-	√
Avoid single point of failure	-	-	-	-	-	√
Resist collusion attack	-	-	-	-	-	√

- Replay attack

Replay attack, also known as freshness attack, refers to that the attacker repeatedly sends a valid data that the receiver has received, so as to achieve the purpose of fraud. This type of attack seriously damages the correctness of authentication. In the proposed scheme, every interaction between SM and DC is a challenge for both sides. In the key agreement stage, the new random number a generated by SM will be used to generate session key. Similarly, DC also generates a new random number b with the same effect as a. By introducing fresh random numbers, any attempt to persuade the other party to accept the old information based on the old random numbers will fail.

- Impersonation attack

Impersonation attack is generally manifested as stealing and camouflaging valid identity documents to achieve illegal communication. In the proposed scheme, the communication parties verify each other through an asymmetric pre-shared key placed in their storage by MDMS, namely y_{SM} and y_{DC}. In the protocol, SM and DC calculate the common secret through $y_{SM} \times P$ and $H_2(ID_{SM}, \mathrm{y}_{\mathrm{DC}}) \times P_{pub}$ respectively. Even if there is a malicious DC or SM, it cannot successfully imitate another entity, because the secret generated by their calculation does not match the secret calculated by their claimed entity.

- MITM attack

MITM attack is an “indirect” intrusion attack. The attacker can read and modify the transmitted information without the knowledge of both sides of the communication. In the proposed scheme, SM and DC verify the authenticity of the received messages by judging whether $M_1 = M_1^{'}$ and $M_2 = M_2^{'}$ are established respectively, so as to prevent MITM attack.

- Desynchronization attack

In a desynchronization attack, an attacker can block the transmission of messages between the SM and the DC so that they permanently lose key synchronization and cannot communicate normally again. In the proposed scheme, the generation of session key does not depend on the previous session key, so the desynchronization attack is difficult to achieve. Even if the transmitted message is blocked, a new session key can be built at the cost of rerunning the protocol.

- Single point of failure and collusion attack

Traditional identity authentication and key agreement usually rely on TTP, so there is a risk of single point of failure and collusion attack. The single point of failure problem means that once the TTP on which the system depends is occupied or mechanical failure occurs, the whole system will be in a state of paralysis. A collision attack is multiple participants’ attempts to steal and tamper with trading information through private conspiracy.

As described in Section 2, the AMI components' key material generation phase and key agreement phase are relatively independent, and mutual authentication and key agreement completion between them do not need to rely on TTP. In addition, L_{DC} interacting with SM is randomly selected from DCN. Even if L_{DC} fails, the key agreement process can be restored only at the cost of rerunning the election algorithm once.

In the proposed scheme, only legitimate devices can obtain the parameters published by the system. Even if the insiders maliciously manipulate, the proposed consensus algorithm can also ensure the consistency of transaction content within the fault-tolerant ability to avoid collusion attacks to a certain extent.

4. PERFORMANCE EVALUATION AND COMPARISON

In this section, we analyse the performance of the proposed protocol from the aspects of computation and communication cost and compare it with the existing related protocols. Here, SM and DC will act as initiators and responders of key exchange protocol. To better compare with other protocols, we refer to the actual running time of relevant operations described in [16]. See Table 3 for details.

Table 3

The computation time of encryption operations.

Notation	Cryptographic Operation	Computation Time
T_{pm}	ECC point multiplication	3.4300 s
T_{bp}	bilinear pairing	6.2920 s
T_{hf}	hash function	0.0092 s
T_{HM}	Keyed-Hash MAC	0.0183 s
T_{rn}	random number	0.0070 ms
T_{se}	symmetric encryption	0.0017 s
T_{sd}	symmetric decryption	0.0016 s
T_{pe}	public-key encryption	2.0830 s
T_{pd}	public-key decryption	1.0620 s
T_{dig}	digital signature	2.8871 s
T_{vdig}	verifying a digital signature	3.6890 s
T_{el}	election of leader	10.5 ms ~ 520 ms

4.1 COMPUTATION COST

We only evaluate the computation cost generated in the key agreement stage. We do not involve the key initialization and material generation stages because these two stages can be completed before the equipment (AMI components) is officially put into operation. Therefore, the performance of the proposed scheme is mainly determined by the key agreement stage.

The above cost refers to the time required to perform encryption operations required for key exchange. In our scheme, SM needs to perform 1 point multiplication, 1 random number generation, and 3 hash operations, while DC needs to perform 4 point multiplication, 1 random number generation, and 4 hash operations. In addition, the DC also needs to participate in the leader node election. It is not difficult to find that the election process is consistent with the leader election used in the raft consensus algorithm. According to [16], the blockchain system needs to meet the following timing requirements before it can elect a relatively stable leader node:

$$broadcastTime << electionTimeout << MTBF \tag{1}$$

$broadcastTime$ indicates the average time required to broadcast a message and receive a response. $electionTimeout$ indicates the system's preset election timeout, specifically the time required for the node to change from follower to candidate. indicates the average time between failures of a blockchain node. The size of $broadcastTime$ and $MTBF$ depends on the underlying properties of the system, while $electionTimeout$ is set by us according to the above inequality. Generally, the $broadcastTime$ ranges from 0.5 ms to 20 ms, the $electionTimeout$ ranges from 10 ms to 500 ms, and the $MTBF$ is several months or more [17].

Table 4 compares the proposed scheme and the existing existing schemes. It is not difficult to find that bilinear pairing schemes, such as SKM+, have relatively high computing costs. According to the data shown in the table, Wu et al. [18] have the lowest calculation cost among these compared protocols, but this is not the case in practical applications. This is mainly because when we compare, SM (*i.e.*, U_I) and DC (*i.e.*, U_R) are calculated and measured under the same hardware facility by default. However, in practical application, there are great differences in the equipment used and the storage and computing power of the equipment. The time cost of DC is much lower than SM for the same operation. Therefore, in comparing these schemes, NIKE+ [3] and the proposed scheme have lower computing costs. Compared with the former, the proposed scheme removes the correlation between the key material generation and negotiation stages.

Further, it improves the feasibility of the implementation of the scheme. In addition, we introduced the operation of leader node election in the key agreement stage. Although this operation increases the scheme's calculation cost, improving the system's robustness is wise. Because in our scheme, the SMs will not only communicate with a specific DC as before, avoiding the problem that the SMS in a certain area cannot operate normally due to the failure of the DC.

Table 4

Comparison of computation costs.

Protocol	U_I	U_R	U_T	
NIKE$^+$[3]	$T_{pm}+T_{rn}+$ $3T_{hf}\approx 3.46s$	$4T_{pm}+T_{rn}+$ $4T_{hf}\approx 13.76s$	-	$5T_{pm}+2T_{rn}+$ $7T_{hf}\approx 17.22s$
Sha [13]	$5T_{hf}$ $\approx 0.05s$	$3361T_{hf}$ $\approx 30.92s$	$1766T_{hf}$ $\approx 16.25s$	$5132T_{hf}$ $\approx 47.22s$
SKM$^+$[11]	$3T_{pm}+T_{rn}$ $\approx 10.29s$	$2T_{bp}+T_{pm}+$ $T_{rn}\approx 16.01s$	-	$4T_{pm}+2T_{rn}+$ $2T_{bp}\approx 26.30s$
A-Mood [14]	$976T_{hf}$ $\approx 8.98s$	$1629T_{hf}$ $\approx 14.99s$	$662T_{hf}$ $\approx 6.09s$	$3267T_{hf}$ $\approx 30.06s$
LAKA [15]	$3T_{pm}+4T_{hf}+$ $2T_{HM}+2T_{se}$ $\approx 10.37s$	$3T_{pm}+5T_{hf}+$ $2T_{HM}+2T_{sd}$ $\approx 10.38s$	-	$6T_{pm}+9T_{hf}+$ $4T_{HM}+2T_{sd}+$ $2T_{se}\approx 20.75s$
Our	$T_{pm}+T_{rn}+3T_{hf}$ $\approx 3.46s$	$4T_{pm}+T_{rn}+$ $4T_{hf}+T_{el}\approx$ $13.77s$ ~$14.28s$	-	$5T_{pm}+2T_{rn}+$ $7T_{hf}+T_{el}\approx$ $17.23s$ ~$17.74s$

4.2 COMMUNICATION COST

This cost represents the number and size of messages transmitted during key agreements. Our proposed scheme requires 4 communications and 8 messages. The communication costs of all schemes are shown in Table 5. The communication bits in the table are based on various lengths of binary sequences [16], such as random number, 32 b; hash function, 160 b; MAC， 256 b; user identity, 160 b; symmetric encryption, 128 b; public key encryption, 160 b; digital signature, 160 b. Therefore, the number of communication bits required in our scheme is: 2|A , IDSM|+|TDC， IDDC， M1|+|M2|=1184 b. Among these schemes compared, the communication cost of the schemes proposed by Sha. and A-Mood. and LAKA protocol is much higher than NIKE+, SKM+, and the scheme proposed in this paper. Compared with NIKE+ protocol, although we have increased the number of communications and messages transmitted, we have achieved high system reliability at a small communication cost, as described in the previous section. Compared with SKM+, although the communication cost of this protocol is low, and it can also meet the requirements of AMI secure communication, its protocol efficiency is far lower than ours.

Table 5

Comparison of communication costs.

Protocol	Number of communications	Number of messages	Number of bits
NIKE$^+$[3]	3	6	992
Sha [13]	8	13	3328
SKM$^+$[11]	3	4	986
A-Mood [14]	4	16	4096
LAKA[15]	2	12	2368
Our	4	8	1184

5. **CONCLUSION**

To ensure the communication security of AMI, this paper proposes a new key establishment scheme mainly used for the interaction between SM and DC. We introduce blockchain technology to alleviate the problems of single-point failure and trust crises in traditional schemes. A blockchain network composed of DCs supports the secure transmission of messages between DCs. The proposed consensus algorithm can promote the whole network nodes to reach an agreement on the block content quickly and ensure the transmission message's authenticity and effectiveness. In addition, in the key agreement process, we flexibly use the leader election algorithm to solve the island phenomenon in the traditional scheme architecture to a certain extent.

Through the security analysis of the proposed scheme, we believe that the scheme can meet the requirements of AMI for communication security. Our scheme has better performance or higher security than other related protocols. Undeniably, introducing blockchain technology will cause high overhead (time cost, communication cost, *etc.*) to a certain extent. Therefore, next, we will shift the focus of our research work to how to improve consensus efficiency and transaction speed.

Received on 31 January 2023

REFERENCES

1. R.R. Mohassel, A. Fung, F. Mohammadi, et al., *A survey on advanced metering infrastructure*, International Journal of Electrical Power & Energy Systems, **63**, pp, 473–484 (2014).
2. H. Tian, Y. Jian, X. Ge, *Blockchain-based AMI framework for data security and privacy protection*, Sustainable Energy, Grids and Networks, **63**, pp. 1–9 (2022).
3. A. Mohammadali, M.S. Haghighi, M.H. Tadayon, et al. *A novel identity-based key establishment method for advanced metering infrastructure in smart grid*, IEEE Transactions on Smart Grid, **9**, *4*, pp. 2834–2842 (2016).
4. N. Liu, J. Chen, L. Zhu, et al., *A key management scheme for secure communications of advanced metering infrastructure in smart grid,* IEEE Transactions on Industrial Electronics, **60**, *10*, pp. 4746–4756 (2012).
5. L. Chen, C. Kudla, *Identity based authenticated key agreement protocols from pairings*, Proceedings of the 16th IEEE Computer Security Foundations Workshop, pp. 219–233 (2003).
6. W. Diffie, M.E. Hellman, *New directions in cryptography*, Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman, pp. 365–390 (2022).
7. D.R.L. Brown, *Some theoretical conditions for Menezes--Qu--Vanstone key agreement to provide implicit key authentication*, Cryptology ePrint Archive (2014).
8. N.L. Biggs, *Cryptography in theory and practice*, Codes: An Introduction to Information Communication and Cryptography, Springer, London, pp. 1–16 (2008).
9. T. Matsumoto, Y. Takashima, H. Imai, *On seeking smart public-key-distribution systems*, IEICE Transactions (1976-1990), **69**, *2*, pp. 99–106 (1986).
10. A. Ghosal, M. Conti, *Key management systems for smart grid advanced metering infrastructure: A survey*, IEEE Communications Surveys & Tutorials, **21**, *3*, pp. 2831–2848 (2019).
11. Z. Wan, G. Wang, Y. Yang, et al., *SKM: Scalable key management for advanced metering infrastructure in smart grids*, IEEE Transactions on Industrial Electronics, **61**, *12*, pp. 7055–7066 (2014).
12. A. Mohammadali, M.S. Haghighi, M.H. Tadayon, et al., *A novel identity-based key establishment method for advanced metering infrastructure in smart grid*, IEEE Transactions on Smart Grid, **9**, *4*, pp: 2834–2842 (2016).
13. K. Sha, N. Alatrash, Z. Wang, *A secure and efficient framework to read isolated smart grid devices*, IEEE Transactions on Smart Grid, **8**, *6*, pp. 2519–2531 (2016).
14. D. Abbasinezhad-Mood, M. Nikooghadam, *Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended Chebyshev chaotic maps*, IEEE Transactions on Industrial Informatics, **14**, *11*, pp. 4815–4828 (2018).
15. P. Kumar, A. Gurtov, M. Sain, et al., *Lightweight authentication and key agreement for smart metering in smart energy networks*, IEEE Transactions on Smart Grid, **10**, *4*, pp. 4349–4359 (2018).
16. A.S. Sani, D. Yuan, W. Bao, et al., *A universally composable key exchange protocol for advanced metering infrastructure in the energy Internet*, IEEE Transactions on Industrial Informatics, **17**, *1*, pp. 534–546 (2020).
17. D. Ongaro, J. Ousterhout, *In search of an understandable consensus algorithm*, USENIX Annual Technical Conference, Philadelphia, PA, USA (June 19-20, 2014).
18. D. Wu, C. Zhou, *Fault-tolerant and scalable key management for smart grid,* IEEE Transactions on Smart Grid, **2**, *2*, pp. 375–381 (2011).