This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2936278, IEEE Transactions on Industrial Informatics

Blockchain Based Anonymous Authentication with Key Management for Smart Grid Edge Computing Infrastructure

Jing Wang, Libing Wu, Kim-Kwang Raymond Choo, Debiao He

***Abstract*—Achieving low-latency and providing real-time services are two of several key challenges in conventional cloud-based smart grid systems, and hence there has been an increasing trend of moving to edge computing. While there have been a number of cryptographic protocols designed to facilitate secure communications in smart grid systems, existing protocols generally do not support conditional anonymity and flexible key management. Thus, in this paper, we introduce a blockchain based mutual authentication and key agreement protocol for edge computing based smart grid systems. Specifically, leveraging blockchain, the protocol can support efficient conditional anonymity and key management, without the need for other complex cryptographic primitives. The security analysis shows that the protocol achieves reasonable security assurance, and the comparative summary for security and efficiency also suggests that the potential of the proposed protocol in a smart grid deployment.**

***Index Terms*—Authentication, Blockchain, Conditional anonymity, Edge computing, Revocation, Smart grid**

I. Introduction

Smart grid system is one of several categories of Industrial Internet of Things (IIoT) can potentially improve the reliability, flexibility and the quality of energy delivery [1]. However, as the system scales (e.g., increasing number of customers), there may be challenges for example in decreasing latency and improving quality of service (QoS) [2]. Hence, there have been attempts to leverage edge computing to mitigate these challenges, for example using electric vehicle charging stations to act as edge computing devices and facilitate real-time decision making and consequently, improving provisioned QoS and eco-friendliness in latency-sensitive applications [3], [4].

As the proverbial saying goes, there is no perfectly secure system and the same can be said for smart grid systems. For example, characteristics inherent of the edge computing architecture such as heterogeneity, mobility, geo-distribution and location-awareness, can be exploited by attackers to carry out their nefarious activities. Thus, designing practical security solutions for edge computing based smart grid system is critical, particularly as smart grids are increasingly common in technologically advanced countries such as United States. Mutual authentication is an effective measure to ensure trust identity and secure communications by verifying the Internet-connected communicators' identity prior to further interactions, without sending sensitive information over an open channel [5].

Conventional public-key infrastructure based protocols are clearly not suitable for smart grid systems, due to the resource constrained nature of smart meters and other Internet of Things (IoT) devices in the grid. In addition, the reliance on a certificate authority (CA) to issue certificates periodically and for new devices incurs high communication overhead and asynchronous problem. Although existing identity based protocols can remove the certificate management problem, it must leak one's real identity to the other communicator for verification. However, an edge server generally is less secure than a centralized cloud and is more susceptible to attacks, and it is *not rational* to send a user's real identity to an edge server. For example, the potential risks of leaking identities in smart grid have been studied in [6].

To prevent the identity from being leaked to the verifier, we can use ring signature schemes as discussed in the existing literature [7], [8]. There are, however, three key disadvantages of deploying ring signature schemes in smart grid systems. First, it is difficult to trace the malicious user (e.g., when fabricated messages are detected). Second, the high computation and communication costs make it impractical for resource-constrained devices. Third, it cannot support flexible participation since a ring is always pre-set. Blind signature schemes suffer from the same shortcomings [6]. Although group signature schemes can provide traceability and dynamic participation, the computation and communication costs are still high for smart meters.

The work was supported by the National Key Research and Development Program of China (No. 2018YFC1604000), in part by the National Natural Science Foundation of China (Nos. 61772377, 61932016, 61972294), the Natural Science Foundation of Hubei Province of China (No. 2017CFA007), and the Science and Technology planning project of ShenZhen (No. JCYJ20170818112550194).

J. Wang is with the Shool of Computer Science, Wuhan University, Wuhan 430072, China
E-mail: cswjing@whu.edu.cn

L. Wu is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China and the Shenzhen Research Institute of Wuhan University, Shenzhen 518057,ÂăChina,
E-mail: whuwlb@126.com

K.-K.R. Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, TX 78249, USA
E-mail:raymond.choo@fulbrightmail.org

D. He is with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China and the Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518000, China

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2936278, IEEE Transactions on Industrial Informatics

In addition, to achieve conditional anonymity and facilitating dynamic participation, a mutual authentication protocol should provide *provable* key update and revocation particularly in the context of an edge computing architecture comprising resource constrained IoT devices. This allows one to update and revoke private keys before their expiry date in order to secure the network, for example when private key compromise incidents are detected and the need to exclude malicious meters is required. Certificate revocation list (CRL) and Online Certificate Status Protocol (OCSP) are two typical revocation tools [9]. However, the former incurs high communication costs and *asynchronous problem*, and the latter needs to be always be online to verify the certificates. In other words, "*how do we effectively and efficiently achieve properties such as conditional anonymity, allowing dynamic participation, key update and revocation in mutual authentication?*"

Seeking to answer the above question, we design a blockchain based anonymous authentication and key agreement protocol for edge computing based smart grid system in this paper. Specifically in such a system, only edge servers would need to join the blockchain system, but end users (e.g., smart meters) do not need to do so. This prevents the leakage of a user's identity to the edge server. Moreover, new end users joining and leaving will not influence other existing end users, due to the identity-based registration. The proposed protocol provides novel conditional traceability and revocability. Specifically, due to the use of smart contract in key management, only the registration authority can link a public key with the corresponding user's real identity. The smart contract recording key materials can help to prove key revocation; thus, avoiding the need to require a trusted center (and prevents single point of failure).

The reminder of this paper is outlined as follows. In Sections II and III, we review related literature and present the problem statements and relevant preliminaries, respectively. Then, our proposed protocol is presented shown in Section IV. In Section V, we present some theoretical analysis to demonstrate the proposed protocol meeting all necessary security requirements. A comparative summary of the performance and security of the proposed protocol and three related protocols (i.e., those of Amor et al. [10], Mahmood et al. [11] and Jia et al. [12]) is presented in Section VI. Finally, we conclude the paper in Section VII.

II. Review of Related Literature

A number of security solutions for smart grids and edge computing systems have proposed in recent years. For example, to establish secure communication sessions, Tsai and Lo proposed an anonymous key distribution protocol using identity-based signature and encryption [13]. He et al. also presented a new authentication and key agreement protocol [14], with reduced computation and communication costs compared to those of [13]. However, it was subsequently pointed out that the protocol is vulnerable to ephemeral secret key leakage and does not achieve smart meter credentials' privacy, and an improved authenticated key agreement protocol was presented [15].

More recently in 2018, Kumar et al. proposed a lightweight anonymous authentication and key agreement protocol [16]. In their approach, the identity anonymity is achieved using symmetric encryption; thus, the neighborhood area network gateway has to maintain many symmetric keys for various home area network gateways. Chaudhary et al. proposed a software defined network (SDN)-enabled security solution for secure communication in a smart grid environment, in order to support real-time data flow. By introducing a third-party authenticator (Kerberos), the peer entities can authenticate with each other. The secure data transmission is guaranteed by the attribute-based encryption mechanism [17]. Then, Chaudhary et al. proposed a lattice-based key exchange protocol using a third party auditor [18]. Compared with [17], this new protocol has improved security and is more effective.

Gope and Sikdar integrated physically uncloneable functions (PUFs) in their proposed authenticated key agreement protocol [19], and they also designed anther lightweight authentication industrial wireless sensor networks using PUF [20]. Wazid et al. proposed a three-factor user authentication protocol that supports dynamic addition, password and biometric update, and traditional anonymity for smart meters [21]. However, the protocol cannot provide flexible revocation to remove / exclude malicious or faulty smart meters. Mahmood et al. proposed an authentication protocol with high efficiency for smart grid[22], but their protocol does not provide anonymity since the real identity of smart meter is transmitted over the open channel in wireless network. Then, Mahmood et al. proposed another anonymous key agreement protocol for smart grid edge computing infrastructure [11]. However, we found that the protocol does not achieve mutual authentication because the smart meter does not verify the validity of utility control (i.e., the value of $Q_i = \frac{1}{b+R_i} S_j$ is not verified).

In 2019, Jia et al. proposed an efficient identity-based anonymous authentication protocol for mobile edge computing [12], and they formally proved the security of their protocol. However, the protocol also does not consider key management of communicating participants. Kahvazadeh et al. proposed a key management and authentication strategy for the edge-cloud computing model [23]. Instead of authenticating with the centralized cloud, all devices authenticate to an authorized control-area unit (CAU). In this model, the cloud and CAU are considered to be trusted and the devices do not validate their legitimacy. Moreover, the protocol does not achieve device anonymity and revocation; thus, limiting its utility in a smart grid system.

To provide conditional anonymity and dynamic participation, Zheng et al. proposed an auditing anonymity communication protocol using linkable group signature, and they also used blind signature, trapdoor indicative commitment and signature of knowledge in the protocol [24]. Li and Cheng proposed a privacy preserved mobile sensing scheme based on the region-based group signature [25]. A privacy protection protocol for trusted smart meters was proposed by Zhao et al., which uses attribute certificates and ring signature [26]. However, it is known that both group signature and ring signature schemes are time-consuming. Amore et al. proposed a privacy-preserving authentication protocol for an edge-fog

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2936278, IEEE Transactions on Industrial Informatics

computing architecture, using pseudonym based cryptography [10]. While the protocol achieves secure key agreement with identity anonymity against fog servers, it does not prevent traceability of outsider attacks. Moreover, if an end user moves away from his/her registration area, the authentication between an user and a fog server has to rely on an online RA, and each fog server has to maintain a verification list (called SV in [10]) to record the valid pseudonym identities. In other words, the communication costs of updating the list can be significantly high and the system may suffer from stolen verifier attacks.

Thus, designing secure and efficient anonymous authentication and key agreement protocols for edge computing based smart grid systems remains challenging.

III. Preliminaries

In this section, we will discuss the relevant background materials.

A. Blockchain

Blockchain is essentially a decentralized database, and a new application mode of distributed data storage, consensus mechanism, encryption algorithm and other technologies [27], [28]. Our proposed protocol is designed to be extremely flexible, allowing any blockchain system that supports both *transactions* and *smart contracts* to be deployed. However, to achieve scalability and efficiency in the smart grid system, one should preferably choose a lightweight consortium blockchain.

- *Transactions*: The transaction is one of the most important components of blockchain systems. Taking the first successful application (i.e., Bitcoin) as an example, its transactions are corresponding signatures of transactional information which mainly contain the sender's address, the receiver's address and the transferred amount. These signatures can be verified by all global nodes and then added to the blockchain. No one can modify the records unless it controls at least 51% of the nodes (known as the majority attack in the literature).
- *Smart contract*: It is first introduced by Nick Szabo in 1994 and defined as "a computerized transaction protocol that executes the terms of a contract" [29]. Smart contracts in blockchain are programmed in a Turing completed language (e.g., Solidity and Serpent) and permanently recorded on the blockchain. Each contract can be regarded as a database slot with a unique address that someone can publish a transaction to trigger its functions for managing database (e.g., storing, adding, deleting and updating of some data entries). Authorized nodes can query the smart contract by submitting some related parameters.

In the proposed protocol, the transactions are used to trigger the smart contract for registration, key update and revocation, and the smart contract helps to record the public keys for identity validity (i.e., supporting the functions of both registry and revocation list).

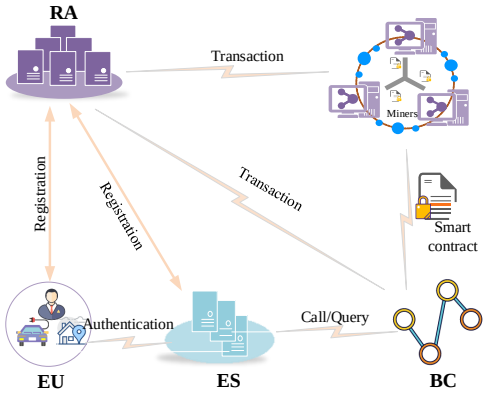

Fig. 1: System model

B. Network model

As shown in Fig.1, the smart grid network model used in this paper consists of registration authority (RA), end users (EUs), edge servers (ESs) and blockchain (BC).

- **RA**: RA is the electricity service provider, and is trusted by all participants in the smart grid system. RA is tasked to distributed key materials for all participants, and it is able to utilize the blockchain to record participants' key materials using smart contract for identity validation, key update and revocation.
- **EU**: The EU is usually a smart device in the smart grid system. For example, smart meters in smart homes report energy consumption and related information to an edge server, and each EU can connect with multiple edge servers, but generally chooses the nearest one.
- **ES**: The ES, acting as the aggregator or utility controller, has the required computation and storage resources. It is responsible to supply timely data analysis and service delivery. Each ES joins the BC network to prevent web spoofing attacks and guarantee the blockchain's normal work. The ES can also communicate with some remote cloud to carry out further data analysis or provide long-term storage.
- **BC**: The BC used in this paper is responsible for recording public key materials in the smart contract. Since it only acts as a trusted recorder for key issuing, update and revocation in this paper, almost all robust blockchain systems can be used in such a smart grid network model.

C. Network assumptions

The network assumptions in our work are as follows:

- The smart contract records are reliable and can be accessed at all times. This is because an attacker can hardly tamper with record issuing on the BC, and the BC is essentially a distributed ledger running all the time.
- The identities and public keys of ESs are known to EUs. In the smart grid system, ESs usually act as the relay nodes that provide timely service, so that it is not necessary to provide identity anonymity for ESs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2936278, IEEE Transactions on Industrial Informatics

- The key materials of ESs do not need to be updated or revoked frequently, unless they are suspected or determined to be corrupted. If it is corrupted, the RA will revoke the server and decline any subsequent service requests from the particular / affected EUs as well as shutting down existing connections.

D. Security requirements

A practical authentication protocol for smart grid edge computing infrastructure should satisfy the following essential security requirements[30], [31], [32], [33], [34], [35].

- *No online RA*: To minimize the communication overhead and resist a single point of failure, it should allow communicators to achieve mutual authentication without relying on an online RA.
- *Mutual authentication*: Only registered EU and ES are allowed in the smart grid system and to run the proposed protocol for validating the communicator's identity prior to message exchange.
- *Session key agreement*: A session key should be generated during the execution of proposed protocol for further secret message exchange. The session key is only shared between the communicators (i.e., EU and ES), and not even the RA can obtain any knowledge about the session key.
- *Identity anonymity*: It should guarantee the EU's identity privacy, such that no potential attacker can obtain EU's real identity information during authentication.
- *Conditional traceability*: To track the identity of malicious or misbehaving users, the protocol should guarantee that there is one, and only one, entity that can reveal the user's real identity.
- *Perfect forward secrecy*: To protect previously transmitted messages, the protocol should guarantee that any attacker is not capable of recovering prior session keys even if it obtains the private key of communicators.
- *Resilience against other attacks*: To further enhance the security, the protocol should provide resilience against other common attacks, such as stolen verifier attacks.

IV. Our proposed protocol

A. System setup

The system setup phase is executed by the RA at the beginning of system deployment, as described below:

1) **Basic initialization:** The RA picks a cyclic additive group $\mathbb{G}$ with generator P and prime order q on an elliptic curve $E(\mathbb{F}_p)$ over the finite field $\mathbb{F}_p$, and two secure one-way hash functions $h_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$, $h_2 : \{0,1\}^* \rightarrow \{0,1\}^{2\kappa}$, where $\kappa = log_2 q$ is the security parameter. Then, the RA selects a random number $s \leftarrow \mathbb{Z}_q^*$ as the master private key, and computes the corresponding master public key $P_{pub} = s \cdot P$.
2) **Blockchain initialization:** RA creates a genesis file including the configure parameters to establish a blockchain. Then, the RA selects several trusted partners and starts the blockchain following specific consensus mechanism (e.g., Practical Byzantine Fault Tolerance in Hyperledger Fabric). For simplicity, the RA can directly join an existing blockchain system (e.g., Ethereum and Hyperledger Fabric).

RA keeps s secretly, and publishes the public system parameters $param = (\mathbb{G}, P, q, h_1, h_2, P_{pub})$.

B. Smart contract deployment

In our proposed protocol, the smart contract is employed to manage the key materials table (KMST). Using the underpinning smart contract, the system can provide conditionally anonymous authentication for communicators, and support efficient revocation without the accompanying asynchronous problem. The related smart contracts are briefly shown in Algorithms 1 to 4.

Algorithm 1. KMST Initialization

```
contract KMST {
  address owner;
  % Define the structure of components in KMST.
  struct KMS {
    byte32 PID;
    uint256[2] R;
    uint256[4] Cipher;
    DataTime expiryTime;}
  KMS[] public KMST;
  % Constructor, automatically invoked when the smart contract is deployed.
  constructor KMST() {
    owner = msg.sender;
    len = 0;
    return 1; }
}
```

Algorithm 2. Update KMST

```
function updateKMST (oldPID, PID, R, Cipher, expiryTime) {
  % Invoked by the TA to update KMST.
  if owner ≠ msg.sender then
    return 0;
  else {
    % If there exists a tuple of <PID, · > in KMST, update it; otherwise, add a new tuple for it.
    if Exist(KMS[i].PID == oldPID) then {
      KMS[i].PID = PID;
      KMS[i].R = R;
      KMS[i].Cipher = Cipher;
      KMS[i].expiryTime = expiryTime;
      return 1; }
    else{
      len++;
      KMS[len].PID = PID;
      KMST[i].R = R;
      KMS[len].Cipher = Cipher;
      KMS[len].expiryTime = expiryTime;
    return 1; }
}
```

C. Registration

The registration phase is interactively executed by the RA and an end user EU_i (or edge server ES_j). Taking an end user EU_i as example, the steps of registration is as below. Assume

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2936278, IEEE Transactions on Industrial Informatics

Algorithm 3. Query KMST

```
function queryKMST (PID) {
  % Invoked by the ES to retrieve specific public key materials.
    if Exist(KMS[i].PID == PID) then
      return KMS;
    else;
      return 0;
}
```

Algorithm 4. Revoke KMST

```
function revokeKMST (PID) {
  % Invoked by the TA to revoke a vehicle.
  if owner ≠ msg.sender then
    return 0;
  else {
    % If there exists the tuple of <PID,· > in KMST, delete
it.
    if Exist(KMS[i].PID == PID) then {
      Release(KMS[i]);
      for ; i < len; i++
        KMS[i] = KMS[i+1];
      len−−;
      return 1; }
    else
      return 0; }
}
```

that the communication channel between RA and EU_i in this phase is a private and secure.

1) Upon EU_i sending its identity ID_i to the RA for registration request, the RA checks whether the user has previously registered, and if so, it will abort the registration request. Otherwise, it will compute EU_i's key materials:
   - The RA picks a random number $r_i \in \mathbb{Z}_q^*$, and computes $R_i = r_i \cdot P$, $x_i = r_i + s \cdot h_1(ID_i||R_i)$ and $PK_i = x_i \cdot P$.
   - The RA computes $PID_i = h_1(PK_i)$ and encrypts ID_i as $C_i = Enc_{P_{pub}}(ID_i)$ using a secure public encryption algorithm, and sets an expiry time ET_i for the user. Then, the RA uploads the entry (PID_i, C_i, R_i, ET_i) to the smart contract by invoking Algorithm 2, i.e., $\mathbf{updateKMST}(Null, PID_i, C_i, R_i, ET_i)$.
   - The RA sends the secret message (x_i, PK_i) to EU_i.
2) EU_i computes $PID_i = h_1(PK_i)$ and then invokes Algorithm 3 (i.e., $\mathbf{queryKMST}(PID_i)$) to obtain (PID_i, C_i, R_i, ET_i). EU_i then checks the validity of received key pair using the following steps.
   - EU_i checks the validity of $x_i \cdot P = R_i + h_1(ID_i||R_i) \cdot P_{pub} = PK_i$.
   - EU_i stores the secret key x_i securely if the above equation passes the verification. Otherwise, EU_i asks for the registration again.

The registration process for ES_j is similar to EU_i, and hence we omit the process here.

D. Authentication

The authentication phase is interactively run by the registered EU_i and ES_j. And in this phase, the BC helps to offer trust assistance for the identity validation.

1) $\mathbf{EU_i} \rightarrow \mathbf{ES_j} : M_1 = (A, pid_i, k, t_i)$
   - EU_i generates a random number $a \in \mathbb{Z}_q^*$, and computes $A = a \cdot P$, $pid_i = PK_i \oplus h_2(A||a \cdot PK_j)$.
   - EU_i computes $k = a + x_i \cdot h_1(PK_i||pid_i||A||t_i)$, where t_i is the current timestamp.
   - EU_i sends (A, pid_i, k, t_i) to ES_j.
2) $\mathbf{ES_j} \rightarrow \mathbf{EU_i} : M_2 = (B, w, t_j)$
   - ES_j first obtains EU_i's public key by computing $PK_i^{'} = pid_i \oplus h_2(A||x_j \cdot A)$ if $|t_i - t_i^{'}| \leq \Delta t$, where $t_i^{'}$ is the current timestamp and Δt is a predefined threshold value.
   - ES_j computes $h(PK_i^{'})$ and invokes the interface $\mathbf{queryKMST}(h(PK_i^{'}))$ for validity checking. If the result is "0", then the user is determined not to be valid. Otherwise, if the returned ET_i has not expired, it then verifies if $kP = A + h_1(PK_i^{'}||pid_i||A||t_i) \cdot PK_i^{'}$ holds.
   - ES_j selects a random number $b \in \mathbb{Z}_q^*$ and computes $B = b \cdot P$, $K_1 = x_j \cdot A + b \cdot PK_i, K_2 = b \cdot A$, if the above verification is successful. And then it computes $SK_{ji} = h_1(PK_i^{'}||ID_j||K_1||K_2)$ and $w = h_1(SK_{ji}||K_1||K_2||t_j)$.
   - ES_j sends (B, w, t_j) to EU_i.
3) $\mathbf{EU_i} : SK_{ij}$
   - EU_i computes $K_3 = a \cdot PK_j + x_i \cdot B$ and $K_4 = a \cdot B$ if $|t_j - t_j^{'}| \leq \Delta t$, where $t_j^{'}$ is the current timestamp.
   - EU_i computes $SK_{ij} = h_1(PK_i||ID_j||K_3||K_4)$, and sets it as the session key if $h_1(SK_{ij}||K_3||K_4||t_j) = w$ holds.

Correctness. From the above equations, we obtain:

$$\begin{aligned} K_1 &= x_j \cdot A + b \cdot PK_i = x_j \cdot aP + b \cdot x_i P \\ &= a \cdot (x_j P) + x_i \cdot (bP) = a \cdot PK_j + x_i \cdot B = K_3 \quad (1) \\ K_2 &= b \cdot A = b \cdot aP = a \cdot (bP) = a \cdot B = K_4 \end{aligned}$$

Thus, we obtain $SK_{ij} = SK_{ji}$.

E. Update and revocation

a) Update: There are also two cases that EU_i should update the key materials. In the first case, when it reaches the expiry time ET_i and the end user is still valid in the smart system, the RA will generate new private key and corresponding public key with the same steps as those described in the registration phase. Note that the ciphertext C_i is also updated to prevent traceability. In the second case, if EU_i's private key is compromised, EU_i has to ask for key update and then the RA helps to update the key materials. When there is a update event for EU_i, the RA invokes the smart contract interface $\mathbf{updateKMST}(oldPID_i, PID_i, R_i, C_i, ET_i)$.

b) Revocation: There are also two cases that EU_i's (or ES_j's) private key and public key will get revoked. In the first case, if the RA discovers suspicious behavior / activities from EU_i, the RA sends a revocation transaction, and deletes the entry $(h(PK_i), R_i, \sigma, txid_i, ET_i)$ from the smart contract RSC. In the second case, if EU_i wants to leave the system, it sends a revocation request to RA, and the RA will

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2936278, IEEE Transactions on Industrial Informatics

revoke EU_i's keys accordingly. When there is a revocation event for EU_i, the RA invokes the smart contract interface $\mathbf{revokeKMST}(PID_i)$.

V. Security Analysis and Comparison

Now, we discuss how the proposed protocol satisfies the security requirements presented previously in Section III-D.

- *No online RC*: It is clear that the proposed protocol does not need to call the RA to complete the authentication for communicators. In addition, there is no need for the RA to maintain a revocation list online for identity validation.
- *Mutual authentication*: 1) Assume that an attacker successfully forge a valid login message, then we have $kP = A + v \cdot R_{ut} + v \cdot h_{ID_{ut}} P_{pub}$, where $v = h_1(PK_{ut}||pid_{ut}||A||t_{ut})$. By invoking the forking lemma, the attacker intends to execute the aforementioned procedure once again with the same input randomness but receives different hash oracle answers. Another message $(A^*, pid^*_{ut}, k^*, t^*_{ut})$ is produced to pass $\mathcal{S}$'s verification, such that $k^*P = A^* + v^* \cdot R_{ut} + v^* \cdot h^*_{ID_{ut}} P_{pub}$. Thus, there will be a solution $(k - k^*)(h_{ID_{ut}} - h^*_{ID_{ut}})^{-1}$ to the instance (P, sP), which contradicts the ECDL assumption. In other words, no attacker can successfully forge a valid authentication message. 2) Assume that an attacker outputs a valid message (B, w, t_j) to pass the authentication of the EU, then there will be a solution to break the ECCDH assumption. This is because if w is a valid authenticator such that $w = h_1(SK_{ij}||K_3||K_4||t_j)$, the attacker has to compute $K^* = K_3 = aPK_j + x_iB$. Then, it can obtain $(a \cdot x_j)P = K^* - x_iB$ as a solution to solve ECCDH, which contradicts the ECCDH assumption. In other words, no attacker can successfully forge a valid authentication message.
- *Session key agreement*: To compute a correct session key, an attacker has to obtain the values of $K_3 = aPK_j + x_iB$ and $K_4 = aB$, even if the public key PK_i and identity ID_j are exposed to the attacker. While the private keys and random nonce a, b are not transmitted on the public channel, the attacker can obtain a valid session key only on the premise that it breaks the ECCDH assumption.
- *Identity anonymity*: In the proposed protocol, the EU_i uses its public key instead of its real identity to authenticate with the ES_j. We even blind the public key PK_i by $pid = PK_i \oplus h_2(A||aPK_j)$ to provide unlinkability.
- *Conditional traceability*: Since the proposed protocol provides identity anonymity and unlinkability for EU_i, any outside attacker cannot trace the behaviors of EU_i. While the trusted party RA can decrypt EU_i's identity related to its public key.
- *Perfect forward secrecy*: Suppose that the private keys of both EU_i and ES_j are compromised, and the messages (A, pid_i, k, t_i), (B, w, t_j) are intercepted by an attacker. To obtain a previous session key SK_{ij}, the attacker can easily obtain the other parts of the session key but cannot get aB (or bA) since a, b are randomly chosen and not transmitted on the public channel. In other words, the attacker is unable to compute the session key.

TABLE I: : A Comparative Summary for Security properties

Security Properties	Protocols			
	Ref[10]	Ref[11]	Ref[12]	Ours
No online RA	×	×	✓	✓
Mutual authentication	✓	×	✓	✓
Session key security	✓	×	✓	✓
Conditional anonymity	✓	×	×	✓
Un-traceability	×	✓	✓	✓
Efficient update	✓	×	×	✓
Efficient revocation	×	×	×	✓
Stolen verifier attack	×	✓	✓	✓
Man-in-the-middle attack	✓	×	✓	✓
Impersonation attack	✓	×	✓	✓

✓: satisfying the security property
×: not satisfying the security property

- *Resilience against other attacks*: There are a number of known attacks that the proposed protocol can resist.
  (1) Impersonation attack: In this protocol, if an attacker wants to impersonate an authenticated user, it must break the MA-security (being proven in Theorem 1, unless it can obtain the corresponding private key).
  (2) Stolen verifier attack: Since the smart contract acts as an encrypted verifier table always running on the BC, the proposed protocol is secure against the stolen verifier attack.
  (3) Replay attack: In the proposed protocol, we use both timestamp and randomness to achieve replay attack resilience.

We now compare our proposed protocol with three recent related protocols, namely those of Amor et al. [10], Mahmood et al. [11] and Jia et al. [12]. As shown in Table I, the protocols in [11] and [12] do not support conditional anonymity, meaning that they have to leak the end user's real identity to the edge server or control utility, and they do not support efficient key update and provable revocation. Although the protocol in [10] can achieve conditional anonymity, it cannot prevent malicious traceability and stolen verifier attack, and the authentication with a cross-domain edge server needs an online RA. It is clear that our proposed protocol can provide much better resilience against these attacks and supports the essential security properties.

VI. Performance Evaluation and Comparison

In this section, we evaluate the efficiency of our proposed protocol and those of Amor et al. [10], Mahmood et al. [11] and Jia et al. [12].

Since Jia et al. [12] have evaluated the performance of some cryptographic operations used in the related protocols on the Alibaba Cloud under Ubuntu system and on a Google Nexus One smart phone to simulate the power of edge server and end user respectively, we will directly use their parameters for the comparison. Table II summarizes their experiment results. The runtime of some lightweight operations and the performance evaluation of registration phase are omitted, since they have little influence on the whole system performance. A comparative summary of the computation costs and communication costs for the three protocols [11],[12],[10] and ours are depicted in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2936278, IEEE Transactions on Industrial Informatics

TABLE II: Experiment results shown in [12]

Notation	Description	Alibaba Cloud	Google Nexus
T_{bp}	Bilinear pairing	5.275 ms	48.660 ms
T_{Gm}	Scalar multiplication on $\mathbb{G}$	1.970 ms	19.919 ms
T_{Ga}	Addition on $\mathbb{G}$	0.012 ms	0.118 ms
T_h	General hash function	0.009 ms	0.089 ms
T_e	Modular exponentiation on $\mathbb{Z}_q$	0.339 ms	3.328 ms
$\|G\|$	Bit length of an element in $\mathbb{G}$	1024 bits	1024 bits
$\|G_T\|$	Bit length of an element in $\mathbb{G}_T$	1024 bits	1024 bits
$logq$	Bit length of an element in $\mathbb{Z}_q$	160 bits	160 bits
$\|T\|$	Bit length of a timestamp	32 bits	32 bits
$\|ID\|$	Bit length of an identity	256 bits	256 bits

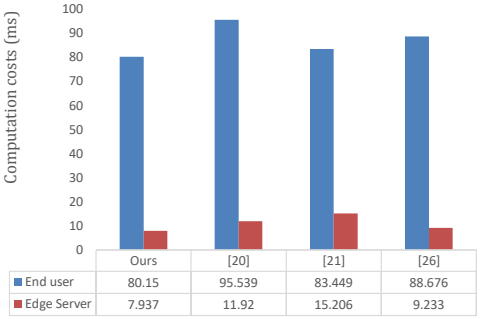

Fig. 2: Computation and communication costs for authentication: A comparative summary

Fig.2, and the specific operations and the length of transmitted messages are summarized in Table III.

To evaluate the operations of smart contract, we use the hyperledger-composer to build a permissioned test chain, where the version of composer is V0.20.7. The blockchain runs on the x86_64 GNU/Linux system with 1 core and 2GB RAM. With the Docker Engine, there are four permissioned nodes in the blockchain network, i.e., Fabric CA, Orderer, Committer and Endorser. The runtime of Algorithms 2 to 4 as shown in Table IV. Note that the running time is the sum total of the time for transaction issuing, verifying and synchronization.

From Fig 2, we know that both computation and communication costs of our proposed protocol are the least for the basic cryptographic operations. And in our protocol, it needs only one round to exchange messages for mutual authentication, but the protocol in [10] needs three rounds of message exchanges for authentication and the protocol in [11] needs two rounds of message exchange. Note that if an end user wants to

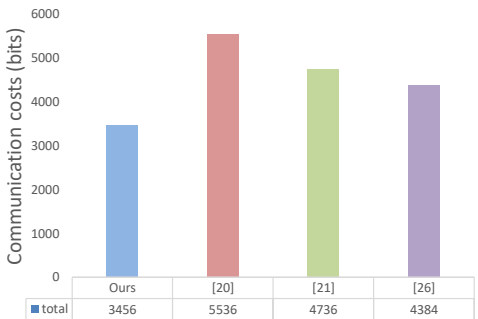

Fig. 3: Comparison of communication costs for authentication

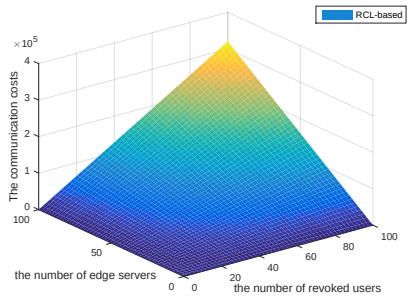

Fig. 4: Communication costs for CRL-based revocation(B)

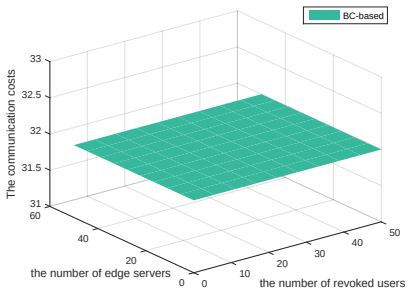

Fig. 5: Communication costs for BC-based revocation(B)

authenticate with an edge server away from its registration field, then the communication costs of protocol [10] will be much higher than those shown in Table III. Moreover, the cost of updating the key materials for end users is also higher than our proposed protocol, since the protocol in [10] has to update all verifier list sent to the corresponding edge servers.

When it comes to the revocation, the authentication time cost at the edge server's side may be as much as 234.233 ms since the blockchain query costs 0.225s. Although it seems to be a little higher, it is still practical for smart grid systems, because a real edge server should have much more computation resources than our simulation platform. Moreover, there is no additional computation costs at the end user's side while supporting conditional anonymity. Since the other protocols do not support dynamic revocation, we do not compare the computation performance on revocation here. Fig 4 shows the communication costs of revocation by utilizing the CRL. We can see that the costs will increase rapidly as the number of revoked users and the number of edge servers increase. However, it is obvious that the communication costs of blockchain-based revocation in our proposed protocol is a constant number (i.e., the length of PID), since it only needs to invoke the smart contract $\mathbf{revokeKMST}(PID)$ given a parameter PID as shown in Fig.5.

Therefore, taking both security and efficiency into account, our proposed protocol is more suitable for the edge computing based smart grid systems.

VII. Conclusion

The capability to achieve private and secure communication between end users and edge servers is crucial in edge computing based smart grid infrastructure. This paper presented a novel anonymous authentication and key agreement protocol with efficient key management. Compared with most existing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2936278, IEEE Transactions on Industrial Informatics

TABLE III: Computation and communication costs for authentication: A comparative summary

Protocols	Computation(End user)	Computation(Edge server)	Communication	Rounds
Ref[10]	$T_{bp} + 2T_{Gm} + 2T_h$	$T_{bp} + 2T_{Gm} + 2T_h$	$3\|G\| + 5logq + 2\|ID\|$	3
Ref[11]	$T_{bp} + 2T_{Gm} + T_{Ga} + 2Te + 3T_h$	$T_{bp} + 3T_{Gm} + T_{Ga} + 2Te + T_{inv} + 5T_h$	$\|G_T\| + 4\|G\| + logq + \|ID\|$	2
Ref[12]	$4T_{Gm} + T_e + 5T_h$	$2T_{bp} + 5T_{Gm} + 3T_{Ga} + 5T_h$	$4\|G\| + 2logq + 2\|T\| + \|ID\|$	1
Ours	$4T_{Gm} + T_{Ga} + 5T_h$	$4T_{Gm} + T_{Ga} + 6T_h$	$3\|G\| + 2logq + 2\|T\|$	1

TABLE IV: Time costs(in s) of the smart contract

Operations	**updateKMST**	**queryKMST**	**revokeKMST**
Max Time	2.681	0.312	2.590
Min Time	1.989	0.138	2.086
Average Time	2.335	0.225	2.338

protocols, including those of Amor et al. [10], Mahmood et al. [11] and Jia et al. [12], the proposed protocol not only provides basic security properties (i.e., mutual authentication, secure key agreement and replay attacks), but it also achieves other important security properties. The highlight of this protocol is that it offers efficient key update and revocation with reduced communication costs, and conditional identity anonymity with reduced computation costs. Findings of the performance evaluation also demonstrate that the proposed protocol is even more efficient than those of Amor et al. [10], Mahmood et al. [11] and Jia et al. [12], while achieving the relevant security properties.

REFERENCES

[1] L. Lyu, K. Nandakumar, B. I. P. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. Industrial Informatics*, vol. 14, no. 8, pp. 3733–3744, 2018. [Online]. Available: https://doi.org/10.1109/TII.2018.2803782

[2] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 44–51, 2018. [Online]. Available: https://doi.org/10.1109/MCOM.2018.1700622

[3] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of internet of things," *IEEE Trans. Cloud Computing*, vol. 6, no. 1, pp. 46–59, 2018. [Online]. Available: https://doi.org/10.1109/TCC.2015.2485206

[4] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, "Vehicular delay-tolerant networks for smart grid data management using mobile edge computing," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 60–66, 2016. [Online]. Available: https://doi.org/10.1109/MCOM.2016.7588230

[5] L. Wu, J. Wang, K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Trans. Information Forensics and Security*, vol. 14, no. 2, pp. 319–330, 2019. [Online]. Available: https://doi.org/10.1109/TIFS.2018.2850299

[6] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma, "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

[7] M. Badra and S. Zeadally, "Design and performance analysis of a virtual ring architecture for smart grid privacy," *IEEE Trans. Information Forensics and Security*, vol. 9, no. 2, pp. 321–329, 2014. [Online]. Available: https://doi.org/10.1109/TIFS.2013.2296441

[8] Y. Gong, Y. Cai, Y. Guo, and Y. Fang, "A privacy-preserving scheme for incentive-based demand response in the smart grid," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1304–1313, 2016. [Online]. Available: https://doi.org/10.1109/TSG.2015.2412091

[9] K. Rabieh, M. M. E. A. Mahmoud, K. Akkaya, and S. Tonyali, "Scalable certificate revocation schemes for smart grid AMI networks using bloom filters," *IEEE Trans. Dependable Sec. Comput.*, vol. 14, no. 4, pp. 420–432, 2017. [Online]. Available: https://doi.org/10.1109/TDSC.2015.2467385

[10] A. B. Amor, M. Abid, and A. Meddeb, "A privacy-preserving authentication scheme in an edge-fog environment," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2017, pp. 1225–1231.

[11] K. Mahmood, X. Li, S. A. Chaudhry, H. Naqvi, S. Kumari, A. K. Sangaiah, and J. J. P. C. Rodrigues, "Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure," *Future Generation Comp. Syst.*, vol. 88, pp. 491–500, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2018.06.004

[12] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, 2019.

[13] J. Tsai and N. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 906–914, 2016. [Online]. Available: https://doi.org/10.1109/TSG.2015.2440658

[14] D. He, H. Wang, M. K. Khan, and L. Wang, "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography," *IET Communications*, vol. 10, no. 14, pp. 1795–1802, 2016. [Online]. Available: https://doi.org/10.1049/iet-com.2016.0091

[15] 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, 2018. [Online]. Available: https://doi.org/10.1109/TSG.2016.2602282

[16] P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," *IEEE Transactions on Smart Grid*, 2018.

[17] R. Chaudhary, G. S. Aujla, S. Garg, N. Kumar, and J. J. P. C. Rodrigues, "Sdn-enabled multi-attribute-based secure communication for smart grid in iiot environment," *IEEE Trans. Industrial Informatics*, vol. 14, no. 6, pp. 2629–2640, 2018. [Online]. Available: https://doi.org/10.1109/TII.2018.2789442

[18] R. Chaudhary, G. S. Aujla, N. Kumar, A. K. Das, N. Saxena, and J. J. P. C. Rodrigues, "Lacsys: Lattice-based cryptosystem for secure communication in smart grid environment," in *2018 IEEE International Conference on Communications, ICC 2018, Kansas City, MO, USA, May 20-24, 2018*, 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICC.2018.8422406

[19] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," *IEEE Transactions on Smart Grid*, 2018.

[20] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, 2019.

[21] M. Wazid, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Secure three-factor user authentication scheme for renewable-energy-based smart grid environment," *IEEE Trans. Industrial Informatics*, vol. 13, no. 6, pp. 3144–3153, 2017. [Online]. Available: https://doi.org/10.1109/TII.2017.2732999

[22] 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 Generation Comp. Syst.*, vol. 81, pp. 557–565, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2017.05.002

[23] S. Kahvazadeh, X. Masip-Bruin, R. Diaz, E. Marín-Tordera, A. Jurnet, and J. Garcia, "Towards an efficient key management and authentication strategy for combined fog-to-cloud continuum systems," in *3rd Cloudification of the Internet of Things, CIoT 2018, Paris, France, July 2-4, 2018*, 2018, pp. 1–7. [Online]. Available: https://doi.org/10.1109/CIOT.2018.8627111

[24] H. Zheng, Q. Wu, B. Qin, L. Zhong, S. He, and J. Liu, "Linkable group signature for auditing anonymous communication," in *Australasian Conference on Information Security and Privacy*. Springer, 2018, pp. 304–321.

[25] L. Ma, X. Liu, Q. Pei, and Y. Xiang, "Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing," *IEEE Transactions on Services Computing*, 2018.

[26] J. Zhao, J. Liu, Z. Qin, and K. Ren, "Privacy protection scheme based on remote anonymous attestation for trusted smart meters," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3313–3320, 2018.
[27] C. Lin, D. He, X. Huang, K.-K. R. Choo, and A. V. Vasilakos, "Bsein: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *Journal of Network and Computer Applications*, vol. 116, no. 1, pp. 42–52, 2018.
[28] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, no. 1, pp. 45–58, 2019.
[29] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.
[30] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.
[31] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2015.
[32] D. He, Y. Zhang, D. Wang, and K. K. R. Choo, "Efficient and secure two-party signing protocol for the identity-based signature scheme in the ieee p1363 standard for public key cryptography," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–10, 2018, dOI: 10.1109/TDSC.2018.2857775.
[33] Q. Feng, D. He, S. Zeadally, N. Kumar, and K. Liang, "Ideal lattice-based anonymous authentication protocol for mobile devices," *IEEE Systems Journal*, pp. 1–11, 2018, dOI: 10.1109/JSYST.2018.2851295.
[34] T.-H. Lin, C.-C. Lee, and C.-H. Chang, "Wsn integrated authentication schemes based on internet of things," *Journal of Internet Technology*, vol. 19, no. 4, pp. 1043–1053, 2018.
[35] C. Meshram, C.-C. Lee, S. G. Meshram, and C.-T. Li, "An efficient id-based cryptographic transformation model for extended chaotic-map-based cryptosystem," *Soft Computing*, vol. 23, no. 16, pp. 6937–6946, 2019.

Jing Wang received the B.S. degrees in computer science from from Wuhan University, Wuhan, China, in 2016. She is currently pursing a Ph.D degree in the School of Computer Science, Wuhan University, China. Her main research interests include cryptography and information security, in particular, secure cloud storage and cryptographic protocols.

Libing Wu was born in 1972. He received the B.S. and M.S. degrees in computer science from Central China Normal University, Wuhan, China, in 1994 and 2001, respectively. He received his Ph.D. degree in computer science from Wuhan University in 2006. He is now a Professor in the School of Computer Science, Wuhan University, China. He is a senior member of IEEE and CCF. His areas of research interests include distributed computing, trusted software and wireless sensor networks.

Kim-Kwang Raymond Choo (SM15) received his Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio. He is the recipient of various awards including ESORICS 2015 Best Paper Award, Winning Team of the Germanys University of ErlangenNuremberg (FAU) Digital Forensics Research Challenge 2015, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer SocietyâĂŹs Wilkes Award in 2008. He is an Australian Computer Society fellow, and an IEEE senior member.

Debiao He received his Ph.D. degree in applied mathematics from School of Mathematics and Statistics, Wuhan University in 2009. He is currently a professor of Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.