Wang, Mingjun; Yan, Zheng; Song, B.; Atiquzzaman, M.

**AAKA-D2D: Anonymous Authentication and Key Agreement Protocol in D2D Communications**

*Published in:*
2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation

*DOI:*
10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00248

*Published: 01/01/2019*

*Document Version*
Peer reviewed version

*Please cite the original version:*
AAKA-D2D: Anonymous Authentication and Key Agreement Protocol in D2D Communications

Mingjun Wang*, Zheng Yan†, Bin Song*, Mohammed Atiquzzaman§

*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, Shaanxi, China
†State Key Lab of ISN, School of Cyber Engineering, Xidian University, Xi'an, Shaanxi, China
‡Department of Communications and Networking, Aalto University, Espoo, Finland
§School of Computer Science, University of Oklahoma, Norman, USA

Emails: {mjwang, zyan}@xidian.edu.cn, bsong@mail.xidian.edu.cn, atiq@ou.edu

I. INTRODUCTION

Current demands on wireless and mobile communications motivate exploring new technology to improve network performance. Meanwhile, the appearance of new commercial services such as location-based services and content sharing services encourage us to design new paradigm to meet user demands. Device-to-Device (D2D) communication is proposed as a promising technology for communications between mobile devices geographical proximity, which is expected to play a key role in 5G system. D2D communication has shown great potential in improving communication capability, reducing communication delay and power dissipation, as well as fostering multifarious new applications and services. However, new application scenarios, use cases and unique system architecture of D2D communication not only provide benefits, but also lead to many security issues [1]. In conventional cellular networks, such as 3G or LTE networks [2], the communication model is centralized, which means User Equipment (UE) communicates with other UEs through the relay by Core Network (CN). The traffics are protected using Evolved Packet System Authentication and Key Agreement (EPS AKA) protocol [3] under the control of CN. Two UEs could authenticate with CN and negotiate session keys separately to protect subsequent communication sessions. However, in D2D communications, the direct D2D links between multiple UEs are novelty thus no existing scheme is available to protect the data confidentiality and integrity of D2D traffic against eavesdropping and modification by attackers. Meanwhile, authentications between UEs directly are more vulnerable to impersonation attacks. Thus, authentication and key agreement become indispensable for establishing secure D2D communications.

Many security schemes on authentication and key agreement for D2D communication have been proposed [4]–[14] recently. In these works, some schemes [4], [5] are based on security infrastructure of existing LTE networks. They adopt Key Deviation Function (KDF) and secret information stored in USIM card to achieve UE authentication and derive D2D session key. However, these schemes suffer from a strong assumption that CN is a fully trustworthy third party. However, in many application scenarios, CN is curious about communication contents between UEs for commercial purposes, e.g., personalized advertisement. Other works [6]–[14] use heavy cryptographic methods, e.g., Diffie-Hellman Key Exchange (DHKE) [15], asymmetric encryption or digital signature algorithms, to achieve user authentication and D2D session key agreement. Among these works, [8]–[10] only address CN-abSENT D2D communication scenarios, which are similar to mobile adhoc network. Works [11]–[14] explore methods to secure D2D communications with joint control of CN and UE. Yan et al. [6] proposed a flexible method to address secure D2D communications with or without the support of CN. However, the utilization of cryptography makes them suffering from high computation and communication overhead.

Privacy protection is another issue in D2D communications. In many D2D-based interaction scenarios, users expect their personal information, e.g., identifiers and communication content, are protected against leakage. However, most of work [4], [5], [8]–[14] don’t consider this significant issue.

In order to build a security and privacy-preserving D2D communication, in this paper, we propose AAKA-D2D, a novel anonymous authentication and key agreement protocol that assures user mutual authentication, secures D2D communications and preserves user privacy simultaneously. In AAKA-D2D, a UE first subscribe to D2D communication
system to get corresponding pseudonyms and its key pairs. Then, it requests establishing direct D2D connections with other UE under the control of CN. UE leverages Identity-based Signature (IBS) to authenticate the legality of its communication partner without learning the real identity but the pseudonym. If authentication succeeds, UEs are able to generate D2D session key by exchanging session key hints. Finally, UEs could communicate securely via direct D2D connection. In case of disputation, CN that acts as a session manager, can open an individual’s real identity of a disputable signature. This feature encourages good behaviors of UEs and enhances security in the D2D communication system. The contributions of this paper can be summarized as follows.

- We propose AAKA-D2D, an efficient authentication and key agreement protocol for D2D communication under the joint control of CN and UE.
- We realize user identity anonymous by merging pseudonyms management and identity-based signature to protect user identity privacy during mutual authentication.
- We implement AAKA-D2D and verify its effectiveness and efficiency on computation and communication costs.

The rest of the paper is organized as follows. Section II reviews related work. Section III gives preliminaries, including Identity-Based Signature, Bilinear Pairing and the notations used in the paper. Section IV describes AAKA-D2D, followed by security analysis in Section V and performance evaluation in Section V. Finally, Section VI draws conclusions.

II. RELATED WORK

Recently, the security issues in D2D communications have captured attention of many researchers. Wang et al. [1], [16] and Haus et al. [17] conducted extensive reviews on existing related work about authentication and key agreement in D2D communications.

3GPP published a Technical Specification [18] to analyze the security issues in Proximity Services (ProSe), the terminology of D2D communication in 3GPP specification. The specification provides system security requirements for different application scenarios and also includes some security mechanisms to achieve these requirements. But the security mechanisms listed are elementary and most them are designed for a specific security requirement. Comprehensive security scheme is missing.

A number of authentication and key agreement solutions have been proposed [4], [5], [8]–[14]. Alam et al. [5] proposed a key distribution scheme for D2D communications in the LTE-A system by using XOR operation to mix the keys of two pieces of D2D UEs in order to avoid the risky session key transmission between CN and UE. Wang et al. [4] proposed a series of key agreement and authentication protocols that support user roaming and inter-operator communications. However, both methods base on a strong security assumption that CN is fully trusted.

Some researches [8]–[10] focused on securing D2D communications when CN is absent. Sheng et al. [8] proposed a secret key establishment protocol for D2D communications based on DHKE. Goratti et al. [9] proposed a security communication protocol to establish direct links among devices by broadcasting beacon to nearby devices. Kwon et al. [10] proposed two protocols for D2D secure key establishment and authentication based on Bluetooth Pairing by using Ciphertext Policy Attribute-Based Encryption (CP-ABE) [19]. However, these methods are only appropriate for scenarios when CN is absent. Wang et al. [7] proposed two protocols to handle D2D group communication security. Zhang et al. [11], [13] proposed two secure data sharing and transmission protocols for D2D communications in LTE-A with joint control of CN and UE. However, these two protocols only support CN-present scenarios. Moreover, Zhang’s protocol was designed for data sharing, but unsuitable for securing D2D communications in a generic way.

Some schemes [6], [12], [14] have been proposed to ensure secure D2D communications with or without the support of CN. Hsu et al. [12], [14] proposed two protocols to provide accountable and anonymous D2D communication establishment. However, both suffer from heavy computation and communication cost, and cannot support user privacy. Recently, Yan et al. [6] proposed a trust-based secure D2D communication scheme to secure D2D communications with or without CN’s appearance. Moreover, it also uses trust levels instead of real identity to protect user privacy. However, this scheme relies on an external trust management system, which pares down the applicability of this scheme.

In order to solve the privacy and computational open issues, we attempt to design an authentication and key agreement protocol for two-user D2D communications with privacy preservation in this paper.

<table>
<thead>
<tr>
<th>TABLE I DEFINITION OF NOTATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notations</strong></td>
</tr>
<tr>
<td>CN</td>
</tr>
<tr>
<td>$UE_n$</td>
</tr>
<tr>
<td>$RID_n$</td>
</tr>
<tr>
<td>$PID_n$</td>
</tr>
<tr>
<td>$SID_i$</td>
</tr>
<tr>
<td>$PK_n$</td>
</tr>
<tr>
<td>$SK_n$</td>
</tr>
<tr>
<td>$K_{pub}$</td>
</tr>
<tr>
<td>$MSK_i$</td>
</tr>
<tr>
<td>$SSK_i$</td>
</tr>
<tr>
<td>$Encl(-)$</td>
</tr>
<tr>
<td>$Decl(-)$</td>
</tr>
</tbody>
</table>

III. PRELIMINARIES

A. Bilinear Pairing

Let $G_1$ and $G_2$ be two cyclic multiplicative groups with the same prime order $q$. Discrete Logarithm Problem (DLP) is assumed to be hard in both $G_1$ and $G_2$. Let $g_1$ and $g_2$ be two generators of $G_1$ and $G_2$ respectively. Let us have a bilinear map $e : G_1 \times G_1 \rightarrow G_2$, with the following properties:

- **Bilinear:** For all $R, S \in G_1$ and $a, b \in \mathbb{Z}_q^*$, $e(R^a, S^b) = e(R, S)^{ab}$;
Secure and Anonymous D2D communication

- Nondegenerate: There exist \( R, S \in G_1 \) such that \( e(R, S) \neq 1_{G_2} \);
- Computable: There is an efficient algorithm to compute \( e(R, S) \) for any \( R, S \in G_1 \).

For example, we can construct bilinear map \( e \) by the modified Weil or Tate pairings on elliptic curves [20].

B. Identity-Based Signature

Identity-Based Cryptosystem (IBC) allows the public key of an entity to be derived from its public identity information such as name, email address, etc. Shamir [21] first proposed a concept of identity-based cryptography and constructed an Identity-Based Signature (IBS) scheme using RSA function. In our system, IBS scheme is used to achieve the mutual identity authentication among UEs and also message authentication. Compared to the conventional PKI, IBS infrastructure avoids the use of certificates for public key verification and the exchange of public keys (and associated certificates). Therefore it greatly improve the efficiency of computation and communication.

For easy presentation, the notations used in this paper are listed in Table I.

IV. THE PROPOSED PROTOCOL

In this section, we first describe the system model to which AAKA-D2D is applied and define security assumptions for our protocol design. Then, we present AAKA-D2D in detail, consisting of four phases: System Initialization, User Registration, D2D Discovery, Secure and Anonymous D2D Session Establishment.

A. System and Security Model

The system model is given in Fig. 1. In the system, there are two kinds of entities: 1) Core Network (CN) and 2) D2D User Equipment (UE). Herein, both UEs are located within the wireless network coverage area. Each UE has established a secure connection with CN via existing security solutions, i.e. EPS AKA [3] in LTE network. UE can discover other UEs in the vicinity and exchange messages on insecure links, on which data confidentiality and integrity cannot be guaranteed. CN is assumed as a trusted party that is responsible for setting up the system, generating key pairs for UE, managing D2D communication sessions and tracing the real identity of UE. CN is supposed to have sufficient storage and powerful computation capability to perform the role of manager to support various D2D services. However, for various reasons that could offer benefits to CN, we assume the CN to be curious about the contents exchanged in D2D communication sessions. More specifically, a D2D UE would like to establish a secure D2D communication with another UE in the vicinity for the purpose of D2D services, such as content sharing, and gaming. The UEs authenticate with each other and negotiate a common session key, which should only be known by D2D session users in order to protect D2D communication confidentiality against curious CN and attackers. Moreover, the real identities of UE are unrevealed to other UEs (includes the communication partner UE) and attackers.

B. System Initialization

In this phase, CN generates system parameters. The following steps are executed by CN as shown in Fig. 2:

On input of security parameter \( 1^n \), CN generates a tuple \( (G_1, G_2, e, g, q) \) as defined in Section Preliminaries. Then it chooses two cryptographic hash functions \( H_1 : \{0, 1\}^n \rightarrow G_1 \) and \( H_2 : G_1 \times \{0, 1\}^* \times G_2 \rightarrow Z_q^* \). It picks a random number \( s \in Z_q^* \) as system master key, computes \( K_{pub} = g^s \) as system public key and chooses one secure symmetric encryption algorithm \( Enc(\cdot) \) (e.g., Advanced Encryption Standard (AES)). Finally, CN publishes the system parameters \( params = \{G_1, G_2, e, H_1, H_2, g, q, K_{pub}, Enc(\cdot)\} \) and keeps \( s \) a secret.

C. User Registration

\( UE_i \) with real identifier \( RID_i \) firstly registers to CN for D2D service. The following steps should be performed in turn:

- \( UE_i \) sends a registration request including its real identifier \( RID_i \) to CN. Once receiving the request, CN generates a pseudonym \( PID_i \) for \( UE_i \). The pseudonym plays the same
role as real identity in authentication and secure communications with privacy-preserving support. We define the form of pseudonym as below:

$$PID_i = (Pseudonym, ExpiryTime)$$  \hspace{1cm} (1)$$

where ExpiryTime denotes the validity period of PID_i. CN computes the public/private key pair associating with PID_i for UE_i, where public key PK_i = H_1(PID_i) and private key SK_i = PK_i^s. CN send PID_i, the corresponding key pair and the system parameters to UE_i via the existing secure channel. In addition, CN locally maintains a table to manage the related information of UE, i.e., the real identity RID_i, the pseudonyms PID_i and the corresponding key pair PK_i/ SK_i, which are shown in Table II.

**D. D2D Discovery**

As shown in Fig. 3, UE_1 discovers UE_2 with pseudonym PID_2 through a D2D discovery process. More details about the D2D discovery process can be found in [22]. After the discovery, UE_1 and UE_2 start to establish a secure D2D communication channel.

**E. Secure and Anonymous D2D Session Establishment**

In this phase, UE_1 and UE_2 perform anonymous mutual authentication and D2D session key agreement under the joint control of CN and themselves.

UE_1 generates a random number nonce_1 and sends a D2D session establishment request message M_req to CN, consisting of the pseudonyms of UE_1 and UE_2 and random number nonce_1: $M_{req} = (PID_1, PID_2, nonce_1)$.

When CN receives M_req, it checks ExpiryTime for the time validity of PID_1 and PID_2. CN rejects the request if any pseudonym is out of date. Otherwise, CN processes as follows:

1) Creates a new D2D session with identifier SID_i and adds this item into a session management table, shown in Table II.

![Diagram](image-url)
2) Randomly selects a master session key \( MSK_i = v \in \mathbb{Z}_q^* \) for \( SID_i \), where \( MSK_i \) is used for secure key agreement between \( UE_1 \) and \( UE_2 \).

3) Encrypts \( MSK_i, nonce_1 \), received pseudonym of \( UE_2 \) and the session identifier \( SID_i \) using \( Enc(\cdot) \) with \( K_1 \). Herein, \( K_1 \) is a pre-load shared key between CN and \( UE_1 \). Then CN replies to \( UE_1 \)'s request by sending \( M_{rsp} = Enc(K_1, (MSK_i, nonce_1, PID_2, SID_i)) \) and send \( M_{rsp} = Enc(K_1, (MSK_i, PID_1, SID_i)) \) to \( UE_2 \) using key \( K_2 \), where \( K_2 \) is a pre-load shared key between CN and \( UE_2 \).

Having received the response, \( UE_1 \) first decrypts message \( M_{rsp} \), using \( K_1 \). Then, it randomly selects \( r_1 \in \mathbb{Z}_q^* \) and computes \( hint_1 = g^{r_1} \) as as its key hint. Moreover, it picks a random number \( r \in \mathbb{Z}_q^* \) and computes a signature \( \sigma_i \) using its key pair \( PK_i/\text{SK}_i \) as follows:

\[
\sigma_i = (R, S),
\]

where \( R = PK_1^{r_1}, h = H_2(hint_1, T_{s1}, R) \) and \( S = SK_1^{r+h}, T_{s1} \) is timestamp. After computing the signature, \( UE_1 \) encrypts tuple \( (PID_1, hint_1, T_{s1}, \sigma_1) \) using 2D session master key \( MSK_i \) and sends authentication message \( M_{aut1} = Enc_{MSK_i}(PID_1, hint_1, T_{s1}, \sigma_1) \) to \( UE_2 \).

Upon receiving the authentication message \( M_{aut1} \) from \( UE_1 \), \( UE_2 \) decrypts the message using \( MSK_i \). Then it checks whether the timestamp \( T_{s2} \) is valid and the pseudonym \( PID_1 \) is the same as the one received from CN. If both checks pass, \( UE_2 \) verifies the signature as below:

1) Compute \( \overline{PK}_1 = H_1(PID_1); \overline{h} = H_2(hint_1, T_{s1}, R) \).

2) Verify \( e(g, S)^h e(K_{pub}, \overline{PK}_1^\overline{h} \cdot R) \).

Below we show why verification succeeds if all parameters are created as explained:

\[
LHS = e(g, S) = e(g, SK_1^{(r+h)}) = e(g, PK_1^{(r+h)}) = e(g, PK_1^{(r+h)}) = e(g, PK_1^{(r+h)})
\]

\[
RHS = e(K_{pub}, PK_1^{r+h} \cdot R) = e(g^s, PK_1^{(r+h)} \cdot R) = e(g^s, PK_1^{(h+r)}) = e(g^s, PK_1^{(h+r)})
\]

If the verification succeeds, \( UE_2 \) generates authentication message \( M_{aut2} \), similarly as what \( UE_1 \) does and sends it to \( UE_1 \) for authentication. Furthermore, \( UE_2 \) sends CN ACK message.

Upon receiving \( UE_2 \)'s authentication message, \( UE_1 \) verifies the timestamp, the pseudonym and the signature. If all verifications pass, \( UE_2 \) sends ACK message to CN. Meanwhile, both \( UE_1 \) and \( UE_2 \) get a sub-session key for 2D communications by computing \( SSK_i = (g^{r_1})^{r_2} = (g^{r_2})^{r_1} = g^{r_1 + r_2} \). With \( SSK_i \), \( UE_1 \) and \( UE_2 \) could communicate securely via D2D link.

V. Security Analysis

In this section, we theoretically analyze the security of AAKA-D2D against three security objectives: mutual authentication, secure session key generation, and privacy preservation.

A. Mutual Authentication

In the proposed protocol, in order to guarantee the validity of identities of the communication parties, mutual authentications should be performed between UE and CN, and also between two UEs. Since our protocol inherits the security architecture of EPS by introducing security features for D2D communications into it, the mutual authentication between UE and CN can be achieved based on the EPS AKA protocol [3]. We adopt identity-based signature to achieve mutual authentication between UEs. Each UE computes its signature \( \sigma_1 = (R, S) \) for time-stamped key hint by using its public key, private key and the random \( r \). Attacker has no knowledge of the private key and the random number chosen by UE, and thus cannot forge signature. The time-stamp prevents from simply replaying a recorded valid signature. Only the legitimate UE with the correct private key and the secret random number can compute a valid signature.

B. Secure Session Key Generation (Secrecy of Session Key)

To protect the communications over the direct D2D link between two UEs, a distinct session key should be negotiated for each 2D session with the cooperation of two UEs and CN. The session key is generated by carrying out the DHKE algorithm between two pieces of UE. \( UE_1 \) and \( UE_2 \) use their random secret \( r_1 \) and \( r_2 \) to generate their session key hints; then to generate the session key. Thus, only legitimate UE keeping secrets can generate the session key. Moreover, CN provides master session key \( MSK_i \) for each UE to secure communications between UEs at the session key generation phase. The UE that has no knowledge of \( MSK_i \) cannot get the session key generation materials in messages \( M_{aut1} \) and \( M_{aut2} \). On the other hand, CN cannot get the random secrets of UE to generate the session key because of the intractability of the computational Diffie–Hellman problem. Thus, AAKA-D2D preserves the D2D communication privacy from honest-but-curious CN and malicious attackers. At last, a session ID is used in the session key generation in order to guarantee the uniqueness of the session key for a specific D2D session.

C. Privacy Preservation

For AAKA-D2D, we analyze two aspects of the privacy preservation: communication content privacy and identity privacy. For communication contents, in the current security solutions in LTE, communications between two UEs must be routed through the core network, therefore, communication contents between UE can be protected against other UEs but they are open to CN. It means the privacy of communication contents between UE are under risk if the core network is compromised. But in AAKA-D2D, both the attackers and CN are unable to get the plaintext, even though CN participates in key generation process. For identity privacy, pseudonyms are used instead of real identities in the D2D discovery phase, the session establishment phase and, later on, in the D2D session communication. Only CN is able to retrieve the real identity from a pseudonym of a UE. UE has no ability to get the real identity from the pseudonym. For different D2D sessions, UE
could request different pseudonyms from CN and use them for privacy preservation. The link between different pseudonyms of one UE can only be revealed by CN.

VI. PERFORMANCE EVALUATION

In this section, we first analyze the performance of AAKA-D2D in terms of computation complexity and communication cost. Then, we evaluate the performance based on protocol implementation. We also compare the performance of AAKA-D2D with two existing authentication and key agreement D2D communication protocols [11], [14] to show the efficiency of our protocol.

A. Efficiency Analysis

We analyze the efficiency of the protocol in terms of the computation complexity and communication overhead.

1) Computation complexity: AAKA-D2D involves two types of system entities, i.e., CN and UE. We analyze the computation cost of each type and compare the computation complexity with SeDS [11] and CN-GD2C [14]. In [11], the Service Provider (SP) and the evolved NodeB (eNB) play the same role as CN and in [14], eNB, ProSe Function and HSS/AuC play the same role as CN in our protocol. Thus, we analyze SeDS by combining SP and eNB together and CN-GD2C by combining eNB, ProSe Function and HSS/AuC together.

Table III shows the computation complexity analysis of AAKA-D2D based on each step and the comparison with SeDS and CN-GD2C. We can see that the computation complexity of AAKA-D2D is lower than SeDS on all operations both at CN and UE. Compared with CN-GD2C, AAKA-D2D has lighter computation complexity on all operations at CN but has two more pairing operations at UE.

2) Communication overhead: We analyze the communication overhead in the system from two aspects: data transmitted between CN and UE and that between UEs. In AAKA-D2D, the communication overhead between CN and UE in user registration phase only occurs once when user joins the system, so we do not take into account this overhead. In SeDS and CN-GD2C, we also treat related entities in core network as CN as we does above. Table IV shows the comparison of communication overhead. We can find from the table that the communication overhead between UE and CN in our protocol is less than about 200 Bytes compared with SeDS and CN-GD2C, but the communication overhead between UEs is heavier than the others two protocols. However, the total communication overhead of our protocol is only half of SeDS and CN-GD2C, which reduces about 150 Bytes.

B. Simulation and Evaluation

In this subsection, we illustrate the performance testing result of AAKA-D2D based on simulation. Herein, we applied pairing-based cryptography library (PBC) [23] for algebraic operations and OpenSSL [24] for cryptography operations and communication transmission. The simulation used a 160-bit elliptic curve group based on the super-singular curve $y^2 = x^3 + x$ over a base 512-bit finite field. On our testing machine, the pairing operation can be computed in approximately 3.8 milliseconds (ms) (without preprocessing), and the exponentiation computations in $G_1$ and $G_2$ take about 3.4 ms and 0.5 ms, respectively. Random element selection is also a significant operation, which takes about 4.5 ms for $G_1$ and 2.2 ms for $G_2$.

Table V shows the operation time comparisons. As we can see from the table, the most time-consuming operation for UE is the session key generation. It involves some heavy operations, i.e., pairing and exponentiation operations. Compared with SeDS and CN-GD2C, the computation overheads of UE and CN in our protocol are lighter in general than these in SeDS and CN-GD2C. In our protocol, CN works in System Setup and User Registration steps rather than all three processing steps as happens in SeDS and CN-GD2C, which decreases computation overhead of CN. We can also see that the total performance of computation cost of AAKA-D2D are better with 23% decrease than SeDS and 19% than CN-GD2C.

### Table III: Computation Overhead Comparisons

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CN System Setup</td>
<td>1 * Rand + 1 * Exp</td>
<td>1 * Rand + 1 * Exp</td>
<td>1 * Hash</td>
</tr>
<tr>
<td>Establish</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Session</td>
<td>1 * Enc(-) + 2 * Dec(-) + 2 * Pair</td>
<td>1 * Enc(-) + 2 * Dec(-) + 4 * Pair</td>
<td>+ 2 * Enc(-) + 3 * Dec(-)</td>
</tr>
<tr>
<td>Establish</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table IV: Communication Overhead Comparisons

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Between UE and CN (eNB)</td>
<td>33</td>
<td>302</td>
<td>250</td>
</tr>
<tr>
<td>Between UEs</td>
<td>126</td>
<td>48</td>
<td>86</td>
</tr>
<tr>
<td>Total</td>
<td>159</td>
<td>350</td>
<td>336</td>
</tr>
</tbody>
</table>
TABLE V  
OPERATION TIME COMPARISONS

<table>
<thead>
<tr>
<th>Entity</th>
<th>Operation Time (ms)</th>
<th>System Setup</th>
<th>User Regist</th>
<th>Session Key Gen</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AAKA-D2D</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td>12.1</td>
<td>6.0</td>
<td>-</td>
<td>-</td>
<td>75.9</td>
</tr>
<tr>
<td>UE1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>29.0</td>
<td>36.9</td>
</tr>
<tr>
<td>UE2</td>
<td>-</td>
<td>-</td>
<td>28.8</td>
<td>-</td>
<td>57.6</td>
</tr>
<tr>
<td><strong>SeDS [11]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td>16.6</td>
<td>10.5</td>
<td>-</td>
<td>11.5</td>
<td>48.6</td>
</tr>
<tr>
<td>UE1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>31.3</td>
<td>31.3</td>
</tr>
<tr>
<td>UE2</td>
<td>-</td>
<td>-</td>
<td>28.4</td>
<td>-</td>
<td>56.8</td>
</tr>
<tr>
<td><strong>CN-GD2C [14]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td>12.3</td>
<td>6.0</td>
<td>-</td>
<td>7.8</td>
<td>36.1</td>
</tr>
<tr>
<td>UE1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>34.5</td>
<td>34.5</td>
</tr>
<tr>
<td>UE2</td>
<td>-</td>
<td>-</td>
<td>32.7</td>
<td>-</td>
<td>65.4</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

In this paper, we propose AAKA-D2D, a novel anonymous authentication and key agreement protocol that achieves user anonymous authentication, secures D2D communications, and preserves user privacy simultaneously. Two UEs subscribing to D2D communication service could discover each other in close proximity, establish a secure D2D direct communication link by adopting the proposed protocol. They use pseudonyms and secret key pairs obtained from CN to perform privacy-preserving mutual authentication and take advantage of DHKE to negotiate a session key for subsequent D2D direct communications. Analysis on security proves AAKA-D2D meets the security requirements. Performance analysis and experimental results show that the proposed protocol is efficient with regard to both computation and communication.

ACKNOWLEDGMENT

This work is sponsored by the NSFC (grants 61672410, 61802293 and U1536202), Academy of Finland (grants 308087 and 314203), the National Key Research and Development Program of China (grant 2016YFB0800704), National Postdoctoral Program for Innovative Talents (grant BX20180238), the Project funded by China Postdoctoral Science Foundation (grant 2018M633461), the Key Lab of Information Network Security, Ministry of Public Security under grant No.C18614, the Fundamental Research Funds for the Central Universities (grant JB191504), and the 111 project (grant B16037).

REFERENCES

[23] JPBC, “The Java Pairing-Based Cryptography Library (JPBC).”