Wang, Mingjun; Yan, Zheng

Privacy-preserving authentication and key agreement protocols for D2D group communications

Published in:
IEEE Transactions on Industrial Informatics

DOI:
10.1109/TII.2017.2778090

Published: 01/08/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Wang, M., & Yan, Z. (2018). Privacy-preserving authentication and key agreement protocols for D2D group communications. *IEEE Transactions on Industrial Informatics*, *14*(8), 3637-3647. Article 8122069. https://doi.org/10.1109/TII.2017.2778090

Privacy-Preserving Authentication and Key Agreement Protocols for D2D Group Communications

Mingjun Wang, and Zheng Yan, *Senior Member, IEEE*

***Abstract*—Device-to-Device (D2D) communications play a key role in the next generation mobile communication networks and wireless systems (5G) and the Internet-of-Things (IoT) ecosystem. D2D group communications are significant for group based services. In spite of its benefits, new application scenarios and new system architecture expose the D2D group communications to unique security threats. Although there are numerous studies on security and privacy in two-user D2D communications, a lack of solutions on secure and privacy-preserving D2D group communications would restrict their wide usage. In this paper, we propose two Privacy-Preserving Authentication and Key Agreement protocols (PPAKA-HAMC and PPAKA-IBS) to guarantee secure and anonymous D2D group communications. In our protocols, a group of D2D users mutually authenticate with each other without leaking their identity information while negotiate a common D2D group session key for secure communications in a D2D session. Formal security analysis and comprehensive performance evaluation show security and effectivity of our protocols.**

***Index Terms*—Anonymous Authentication, Device-to-Device Communications, Key Agreement, Privacy Preservation**

I. Introduction

Device-to-Device (D2D) communications were proposed as one of the promising technologies for communications in proximity. It is supposed to play as a key role in the next generation mobile communication networks and wireless systems (5G) [1, 2] and the Internet-of-Things (IoT) ecosystem [3, 4]. D2D communications have shown great potential for reducing communication delay, improving communication capability, as well as fostering multifarious new applications and services. D2D group communications is one of significant use cases to provide group based services, e.g., group gaming and group chatting.

While there are numerous benefits, new application scenarios and specific system architecture expose D2D communications into unique security and privacy threats [5]. User identity authentication and key agreement is one of the basic but significant security issues for establishing a secure communication channel between D2D devices. Some authentication and key agreement schemes [6-11] have been proposed for securing D2D communication sessions. But, they only addressed secure D2D communications between two users in the coverage of a Service Network (SN). D2D group communications have more complex security requirements compared with the two-user scenario, such as key agreement and management. So far, only few schemes have been proposed for secure group D2D communications. Recently, Hsu et al. [12, 13] proposed two protocols to provide secure D2D communications with group information anonymity. However, these protocols can only address secure communications between two users. The group anonymity is designed for protecting group information, not for group communication session establishment. They suffer from high performance overhead and cannot resist internal attacks raised by malicious users in the group.

Moreover, users in a group-based service show high concern on their privacy. It is obviously not wise to disclose user privacy, especially identity information to any strangers, even other users in a same group. In particular, group communication data should not be disclosed to any outsiders, including SN. Private information leakage may harm the safety of user properties. However, the current literature still lacks solutions on secure and privacy-preserving D2D group communications.

In this paper, we propose two Privacy-Preserving Authentication and Key Agreement protocols (PPAKAs) to establish secure and anonymous D2D group communications. In the proposed protocols, a group of D2D users, with the support of SN, mutually authenticate with each other without leaking their real identities, while at the same time they negotiate a common D2D group session key for secure communications in the group session. The contributions of our paper can be summarized as follows:

- We first propose a lightweight, provably secure against external attack and privacy-preserving authentication and key agreement protocol, named PPAKA-HMAC. It combines group key agreement with Hash-based Message Authentication Code (HMAC) and pseudonym management for secure and anonymous D2D group communications.
- We further propose an improved protocol, named PPAKA-IBS, by adopting Identity-Based Signature (IBS) instead of HMAC. It achieves authentication and key agreement to establish a secure D2D group communication session. Compared with PPAKA-HMAC, this protocol can resist internal attacks raised by malicious users.
- We apply pseudonym instead of real identity in both protocols in the establishment of a secure D2D group

M.J. Wang is with the State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, 710071, China (e-mail: wangmingjun1987 @hotmail.com).

Z. Yan is with the State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, 710071, China and the Department of Communications and Networking, Aalto University, Espoo, 02150, Finland (e-mail: zyan@xidian.edu.cn).

session to achieve privacy preservation. Only SN and user itself can map the pseudonym to its corresponding real identity. Different pseudonyms are generated by SN for pieces of User Equipment (UE) in different communication sessions.

- We conduct formal security analysis on both protocols to show their security features and provide extensive performance analysis and test to demonstrate our protocols' effectivity.

The remainder of the paper is organized as follows. Section II reviews related work. Section III describes the system model and design objectives. Section IV presents two PPAKA protocols, followed by their performance evaluation in Section V. Finally, a conclusion is drawn in the last section.

II. Related Work

Authentication and key management in D2D communications have captured attention of researchers and practitioners recently. Many AKA schemes have been proposed [6-13]. Wang et al. [6] proposed a series of key agreement and authentication protocols that support user roaming and inter-operator communications. Sheng et al. [7] proposed a protocol to generate a shared secret key between two D2D devices for D2D communications based on Diffie-Hellman Key Exchange (DHKE) [14]. Goratti et al. [8] proposed a security communication protocol to establish direct links among D2D devices by broadcasting beacon to nearby devices in order to authenticate and set up a communication session. Kwon et al. [9] proposed two protocols for D2D secure key establishment and authentication based on Bluetooth Pairing by applying Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [15]. Zhang et al. [10, 11] proposed two secure data sharing and transmission protocols for D2D communications in LTE-A. However, these schemes can only realize two-user D2D secure communications. They cannot ensure secure D2D group communications. Recently, Hsu et al. [12, 13] proposed two protocols to realize secure and group anonymous D2D communications. But these two protocols address the secure D2D communication establishment with group information anonymity for only two users. They failed to solve secure group session establishment.

Some key agreement protocols proposed in Wireless Body Area Networks (WBANs) [16-19], Vehicle Ad-hoc Networks (VANETs) [20-22] and Sensor Networks (SNs) [23-25] are heuristic. However, these protocols in WBANs, VANETs and SNs address different scenarios from D2D group communications. The lack of secure and privacy-preserving solutions for D2D group communication would restrict the range of applications of D2D communications.

III. System Model and Design Objectives

A. System Model

Fig. 1 shows a system model of D2D group communications. In the system, there are two kinds of entities: Service Network (SN) and D2D User Equipment (UE). Herein, UE is located within the wireless network coverage of SN and each UE has secure connection with SN via existing infrastructure. UE can discover other pieces of UE nearby and communicate with them via insecure connection. SN is responsible for user identity and key management, and D2D communication management. It generates and manages pseudonyms for UE, computes key pairs for UE and helps UE to establish D2D communication sessions. In our study, we assume that SN is honest but curious about the contents of D2D communications. It strictly follows protocol design to perform user and session management.

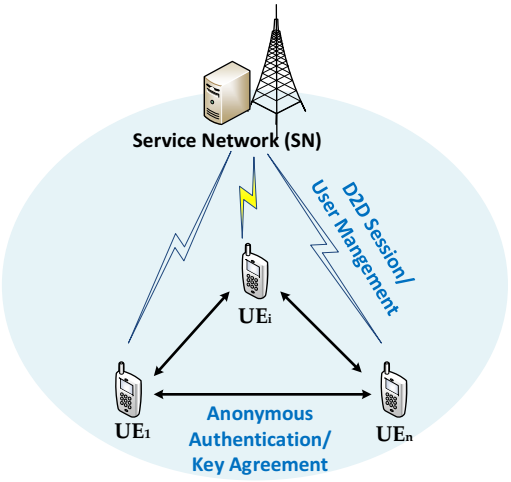

Fig. 1 A System Model

Our protocols are proposed based on the following D2D group communications scenario. Concretely, UEs in proximity want to establish a secure D2D communication group for the purpose of D2D group services. UEs should authenticate with each other and then negotiate a common session key, which should only be known by the group users in order to protect subsequent group communications. SN and non-group users have no knowledge of the session key and any group communication data. Furthermore, in the group communications, a user's real identity should be prevented from leaking to other UEs in the same group and even users outside the group.

B. Design Objectives

In order to countermeasure and mitigate potential threats, several security and privacy features should be satisfied.

1) Authentication

Authentication is required for both a message and a sender. A message receiver should be able to make sure that the message it receives is the original message and is not tampered. Meanwhile, it should be able to check if the message sender is a legitimate user.

2) Identity Privacy Preservation

In the system, UE participates in the D2D group communications using its pseudonym. SN and other UEs cannot get the real identity from the pseudonym.

3) Group Communication Security

The communications of one D2D group session should be secured. Only the member UE in the group is able to access the communication data. Any UE outside the group, even SN is

unable to access the D2D group communication data and eavesdrop the communication in the group.

4) Group Backward Secrecy

For supporting dynamic group management, backward secrecy should be ensured so that new joining users cannot know previous group communication contents.

5) Group Forward Secrecy

Forward secrecy should be also ensured, so that the group user who has left the group will have no way to know future group communication contents.

IV. The Proposed Protocols

In this section, we proposed two protocols to address the privacy-preserving authentication and key agreement issue in D2D group communications. The first protocol, named PPAKA–HMAC, achieves user identity anonymity, mutual authentication and session key agreement by combining group key agreement with HMAC and pseudonym management. It can defend against attacks raised by attackers outside of the D2D group. The second protocol, named PPAKA–IBS, improves PPAKA–HMAC by using IBS instead of HMAC. It can successfully resist internal attacks raised by malicious group users.

A. PPAKA–HMAC

In this section, we describe the first protocol PPAKA–HMAC. The detailed processes are illustrated in Fig. 2, Fig. 3 and Fig. 4, marked with yellow and green color.

1) System Setup

The following steps are executed by SN to generate system parameters:

On input a security parameter 1^k, SN selects an appropriate system prime q and a multiplicative group G of order $q-1$ with generator g. It also chooses a cryptographic hash functions: $H_1: G \rightarrow \mathbb{Z}_q^*$.

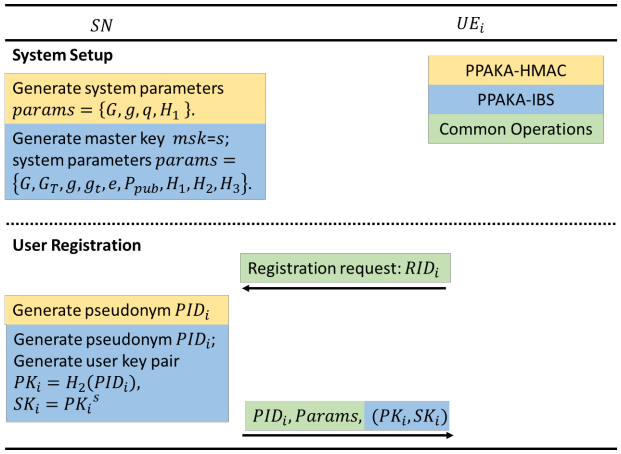

Fig. 2 System Setup and User Registration Phase of PPAKA Protocols

2) User Registration

User UE_i with real identifier RID_i should firstly perform user registration with SN in order to get its pseudonym. The following steps should be performed in turns:

i. UE_i sends a registration request including its real identifier RID_i to SN.

ii. Once receiving the request and checking the eligibility of UE_i, SN generates a pseudonym PID_i for UE_i. The pseudonym plays the same role as the real identity in authentication and secure communications, but can protect identify privacy. We define the form of pseudonym in our system as below:

$$PID_i \stackrel{\text{def}}{=} (Pseud, ExpiryTime),$$

where $ExpriryTime$ denotes the valid period of PID_i,

iii. SN sends PID_i and system parameters to UE_i via a secure channel.

In addition, SN locally maintains a table to manage the related information of UE_i, i.e., the real identity RID_i and the pseudonyms PID_i.

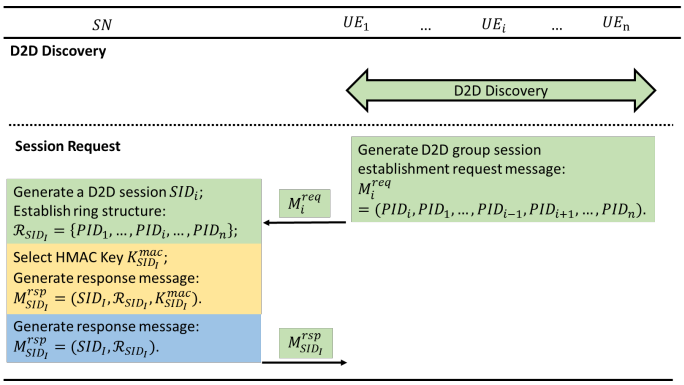

Fig. 3 Discovery and D2D Session Request Phase of PPAKA Protocols

3) D2D Discovery

For not losing generality, as shown in Fig. 3, we assume that n D2D users $UE_1, UE_2, \dots, UE_n$ with pseudonyms $PID_1, PID_2, \dots, PID_n$ discover each other through a D2D discovery process. For more details about a D2D discovery process, please refer to [26]. All UEs that want to establish a secure D2D communication group should anonymously authenticate each other and generate a secure group session key, which is described as below.

4) Session Request

In order to establish a secure D2D communication group, UE_i $(i = 1, \dots, n)$ sends a request message M_i^{req} about D2D group session establishment to SN, where the message is consist of the pseudonyms of UE_i and the other users discovered in the Discovery phase, which is showed as:

$$M_i^{req} = (PID_i, PID_1, PID_2, \dots, PID_{i-1}, PID_{i+1}, \dots, PID_n).$$

Herein, the pseudonyms of the other users in M_i^{req} are disordered. When SN receives the first request from UE_i, it firstly checks the $ExpriryTime$ of PID_i in M_i^{req}. SN rejects the D2D establishment request if PID_i is out of date. Otherwise, SN does the following:

i. SN starts a countdown to wait for the requests from other $n-1$ users included in M_i^{req}. If not all requests are received before the countdown hits zero, SN halts this session and sends reject messages to all users whose requests have been received. The reject message contains the pseudonyms of the users whose requests have not been received by SN. Otherwise, SN creates a new D2D

communication group session with identifier SID_I, and orders all participating users into a ring structure using their pseudonyms. Suppose the ring of n users is $\mathcal{R}_{SID_I} = (PID_1, \dots, PID_i, \dots, PID_n)$, where PID_{i-1} and PID_{i+1} are respectively left and right neighbors of PID_i for $1 \le i \le n$, $PID_0 = PID_n$ and $PID_1 = PID_{n+1}$.

ii. *SN selects a random* $K_{SID_I}^{mac}$ *as HMAC key,* $K_{SID_I}^{mac} \in G$.

iii. Then, SN replies to UE requests by sending a response message $M_{SID_I}^{rsp}$ to all group users, where $M_{SID_I}^{rsp}$ consists of group session information, i.e., the group session identifier SID_I, the ring structure with participating users' pseudonyms $\mathcal{R}_{SID_I} = (PID_1, \dots, PID_n)$ and the shared HMAC key $K_{SID_I}^{mac}$ as follows:

$$M_{SID_I}^{rsp} = \{SID_I, \mathcal{R}_{SID_I}, K_{SID_I}^{mac}\}.$$

iv. SN stores group session information in a session management table locally.

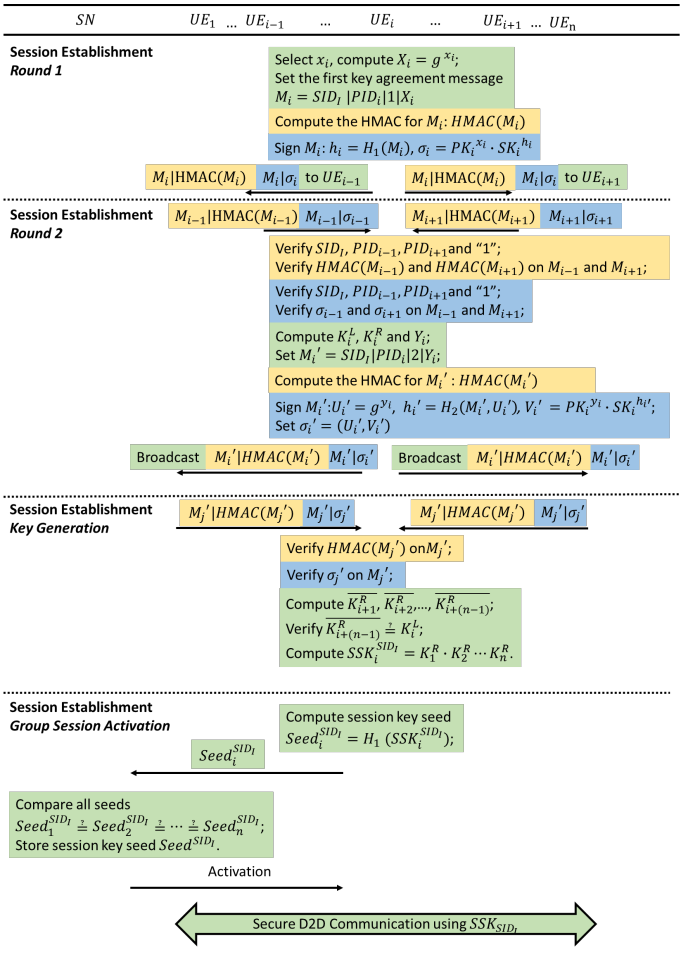

Fig. 4 Session Establishment of PPAKA Protocols

5) Session Establishment

A two-round anonymous authentication and key agreement protocol is proposed in Fig. 4. It is elaborated by combining a secure group key agreement and HMAC in order to realize privacy preserving user authentication and group key generation.

***Round 1*:** In this round, the users involved in the group compute their first group session key hints, which are used for the session key generation. In order to realize the authentication for user identity and message, the user attaches a HAMC code to each message it sends using HMAC key $K_{SID_I}^{mac}$. The following steps should be performed in turns: Upon receiving the response message $M_{SID_I}^{rsp}$ from SN, $UE_i (i = 1, \dots, n)$ picks a random number $x_i \in \mathbb{Z}_q^*$ and computes its first key hint $X_i = g^{x_i}$. Then, it sets the first key agreement message $M_i = SID_I|PID_i|1|X_i$, where parameter "1" indicates the sequence of message from UE_i, and X_i is the first key hint. In order to support identity and message authentication, UE_i uses $K_{SID_I}^{mac}$ to compute the message authentication code HMAC of M_i, which is denoted as $HMAC(M_i)$. Finally, UE_i sends $M_i|HMAC(M_i)$ to UE_{i-1} and UE_{i+1}, where $UE_0 = UE_n$, $UE_{n+1} = UE_1$.

Round 2: Upon receiving $M_{i-1}|HMAC(M_{i-1})$ from UE_{i-1} and $M_{i+1}|HMAC(M_{i+1})$ from UE_{i+1}, UE_i first checks if the session identities in M_{i-1} and M_{i+1} are same as the one it keeps, if the pseudonyms in M_{i-1} and M_{i+1} are within the valid period and if the sequence parameter are both "1". If so, UE_i uses its $K_{SID_I}^{mac}$ to verify $HMAC(M_{i-1})$ and $HMAC(M_{i+1})$. Since $K_{SID_I}^{mac}$ is only known by the UE in session group SID_I and SN, only UEs in session SID_I can verify $HMAC(M_{i-1})$ and $HMAC(M_{i+1})$.

After a series of verifications, UE_i computes a left key $K_i^L = X_{i-1}^{x_i}$, a right key $K_i^R = X_{i+1}^{x_i}$ and its second key hint $Y_i = \frac{K_i^R}{K_i^L}$. Then it sets the second key agreement message $M_i' = SID_I|PID_i|2|Y_i$, computes the message authentication code $HMAC(M_i')$ of M_i' using $K_{SID_I}^{mac}$, and finally broadcasts $M_i'|HMAC(M_i')$ to all the other group users UE_j, where $j = 1,2, \dots, n, j \neq i$.

Key Generation: Upon receiving all second key agreement messages $M_j'|HMAC(M_j')$ from UE_j ($j = 1,2, \dots, n, j \neq i$), UE_i firstly verifies all message authentication code using its $K_{SID_I}^{mac}$. It accepts UE_j and M_j' if all verifications hold.

Finally, UE_i computes $\overline{K_{i+1}^R}, \overline{K_{i+2}^R}, \dots, \overline{K_{i+(n-1)}^R}$ using its own right key K_i^R as follows:

$$\begin{cases} \overline{K_{i+1}^R} = Y_{i+1} \cdot K_i^R \\ \overline{K_{i+2}^R} = Y_{i+2} \cdot \overline{K_{i+1}^R} \\ \cdots \\ \overline{K_{i+(n-1)}^R} = Y_{i+(n-1)} \cdot \overline{K_{i+(n-2)}^R} \end{cases}$$

Then UE_i verifies if $\overline{K_{i+(n-1)}^R}$ is the same as that of its own left key K_i^L:

$$\overline{K_{i+(n-1)}^R} \stackrel{?}{=} K_i^L$$

If the verification fails, UE_i aborts; otherwise, UE_i computes the session key $SSK_i^{SID_I}$:

$$SSK_i^{SID_I} = K_1^R \cdot K_2^R \cdots K_n^R$$

It is obvious that all honest users compute the same key $SSK^{SID_I} = g^{x_1x_2 + x_2x_3 + \cdots + x_nx_1}$.

6) Group Session Activation

Furthermore, in order to guarantee the session key shared among all group users are same and active the group session, UE_i hashes its self-generated session key as key seed

$Seed_i^{SID_I} = H_1(SSK_i^{SID_I})$ and sends $Seed_i^{SID_I}$ to SN for verification.

Upon receiving all session key seeds from the group users, SN compares if all seeds are equal. If so, SN stores $Seed^{SID_I}$ in a session management table and actives this group session SID_I by sending all group users an activation message. After receiving the activation message, all group users can securely communicate with each other in this session using $SSK_i^{SID_I}$.

B. Security Analysis on PPAKA-HMAC

In this subsection, we discuss security issues of the proposed PPAKA-HMAC protocol according to our security objectives.

1) Secure Session Key Establishment

After successful protocol execution, a session key will be shared by all group members securely.

Proof: Our protocols are secure against both passive and active adversaries under Decisional Diffie-Hellman (DDH) assumption. Herein, we assume that the adversaries are external attackers outside of the group. They are capable of eavesdropping (passive adversary), intercepting and modifying messages (active adversary). Our proof follows security proof of authenticated key agreement protocol in [27]. We take advantage of HMAC instead of digital signature used in [27] . We assume that HMAC is secure, then prove the claim below:

Claim 1. Let Forge be the event that all HMACs are forged by adversary $\mathcal{A}$, then $Prob[Forge] \leq Adv_{HMAC}(t')$.

Proof of Claim: We assume that $\mathcal{A}$ is a polynomial time adversary of PPAKA-HMAC. It forges HMACs by querying Send (V, i, M) , where $M = SID_I|PID|l|X_i|HMAC_i$, $V\left(K_{SID_I}^{mac}, SID_I|PID|l\middle|X_i, HMAC_i\right) = 1$. V is the verification algorithm of HMAC. Using $\mathcal{A}$, we construct an algorithm $\mathcal{F}$ to forges HMACs as follows: The forger $\mathcal{F}$ simulates all oracle queries of $\mathcal{A}$ by executing protocol PPAKA-HMAC and obtaining the necessary information related to HMAC signature. If $\mathcal{A}$ ever ouputs a new valid message HMAC pairing with respect to HMAC key $K_{SID_I}^{mac}$, $\mathcal{F}$ outputs this pair as its forgery. The success probability of $\mathcal{F}$ is

$$Prob[Forge] \leq Adv_{HMAC}(t').$$

Thus, we can get the result of our proof following the proof of Theorem 4.2 in [27].

2) Anonymous User Authentication

Proof: In PPAKA-HMAC, UE achieves anonymous authentication by verifying $HMAC(M_j)$ from their neighbors and $HMAC(M_j')$ from other UEs in the group. The UE compares the received $HMAC(M_j)$ and $HMAC(M_j')$ to the values computed by itself using its HMAC key $K_{SID_I}^{mac}$ got from SN. $K_{SID_I}^{mac}$ is a secret generated by SN for each group session and is distributed to all group session members. Since the connections between SN and UE are assumed secure, only the UE in the group has $K_{SID_I}^{mac}$ and can calculate $HMAC(M_j)$ and $HMAC(M_j')$ correctly. If the comparison between received ones and the calculated result is positive, the UE can authenticate that the received $HMAC(M_j)$ and $HMAC(M_j')$ are originated from an eligible UE in the group without revealing the real ID of it.

3) Identity Anonymity

Proof: Since each UE uses a pseudonym, which is generated and distributed by SN randomly, instead of a real identity in the group communications, only SN and UE itself can map the pseudonym with its real identity. The probability for an adversary to reveal the real identity behind the pseudonym is negligible. Thus, the identity privacy can be protected.

4) Discussion

PPAKA-HMAC achieves secure session key agreement, user anonymous authentication and group privacy preservation. It can resist external attacks who do not have the HMAC key $K_{SID_I}^{mac}$. However, if a user inside the group does not follow the protocol honestly and behaves maliciously, the protocol is unable to resist the attacks raised by these malicious users. In order to resist internal attacks, we design the second protocol in the next section. For other security properties, like Internal Attack Resistance, Group Forward Secrecy, and Group Forward Secrecy, we will prove them in Subsection E.

C. PPAKA-IBS

In this section, we propose the second protocol for secure group D2D communications with privacy-preservation. This protocol defends against internal attacks raised by malicious group users by taking advantage of IBS instead of HMAC.

The process of PPAKA-IBS is similar to PPAKA-HMAC except for some operations in each phase. For concise description, we present this protocol by emphasizing its difference from PPAKA-HMAC and omitting their common processes. The detailed procedures of PPAKA-IBS are illustrated in Fig. 2, Fig. 3 and Fig. 4, marked with blue and green color.

1) System Setup

SN executes following operations to generate system parameters: On input a security parameter 1^k, SN generates a tuple $\{G, G_T, q, g, g_t = e(g, g)\}$ as defined in [28]. Then SN picks a random number $s \in \mathbb{Z}_q^*$ as a system master key and computes $K_{pub} = g^s$ as a system public key. It also chooses three cryptographic hash functions: $H_1: G \to \mathbb{Z}_q^*$, $H_2: \{0,1\}^* \to G$ and $H_3: \{0,1\}^* \to \mathbb{Z}_q^*$. Finally, SN keeps s secretly and publishes the system parameters $parmas = \{G, G_T, q, g, g_t, e, K_{pub}, H_1, H_2, H_3\}$.

2) User Registration

In this phase, UE_i registers into SN with its real identifier RID_i just like the processes in PPAKA-HMAC except that UE_i gets an extra public\private key pair from SN. The detail of this extra operation is shown as follows:

After generating pseudonym PID_i for UE_i, SN computes the public/private key pair associating with PID_i for UE_i, where public key $PK_i = H_2(PID_i) \in G$ and private key $SK_i = {PK_i}^s \in G$. Then, SN sends PID_i, the corresponding key pair PK_i/SK_i and the system parameters parmas to UE_i via a secure channel and stores information of UE_i, which includes RID_i, PID_i and PK_i/SK_i.

3) D2D Discovery

The D2D Discovery phase is the same as the operations in PPAKA-HMAC.

4) Session Request

In this phase, UE_i sends a request message to SN for D2D group session establishment and gets session related information from SN. The process of this phase is similar to PPAKA-HMAC except that SN does not generate the HMAC key $K_{SID_I}^{mac}$ for all users in the group. So, the response message $M_{SID_I}^{rsp}$ in PPAKA-IBS only consists of the group session identifier SID_I and the ring structure of group users' pseudonyms $\mathcal{P}_{SID_I} = \{PID_1, \dots, PID_n\}$, which is described as follows:

$$M_{SID_I}^{rsp} = (SID_I, \mathcal{P}_{SID_I}).$$

Moreover, related information of group session, which does not contain any HMAC key, is stored in the session management table by SN locally.

5) Session Establishment

Just like PPAKA-HMAC, a two-round anonymous authentication and key agreement protocol is proposed for session establishment. However, IBS is used instead of HMAC to achieve user identity authentication in PPAKA-IBS.

In ***Round 1 phase***, UE_i generates its first key hint $X_i = g^{x_i}$ and first key agreement message $M_i = SID_I|PID_i|1|X_i$, which is the same as the operation in PPAKA-HMAC. Next, UE_i signs M_i instead of generating the HMAC of M_i before sending it to its two neighbors UE_{i-1} and UE_{i+1}. The signature signing is performed as below:

i. Hash M_i: $h_i = H_3(M_i)$;
ii. Compute the signature of M_i as $\sigma_i = PK_i{}^{x_i} \cdot SK_i{}^{h_i}$ using its key pair PK_i/SK_i and random number x_i.

Finally, UE_i sends $M_i|\sigma_i$ to UE_{i-1} and UE_{i+1}, where $UE_0 = UE_n$, $UE_{n+1} = UE_1$.

In ***Round 2 phase,*** signature verification is performed by UE_i to verify the correctness of signatures. But in PPAKA-HMAC, HMAC is used for authentication. The detailed processes are presented below:

Upon receiving $M_{i-1}|\sigma_{i-1}$ from UE_{i-1} and $M_{i+1}|\sigma_{i+1}$ from UE_{i+1}, UE_i first checks if the session identities in M_{i-1} and M_{i+1} are same as the one it keeps, if the pseudonyms in M_{i-1} and M_{i+1} are within the valid period and if the sequence parameter are both "1". If so, UE_i performs signature verifications as blow:

i. Compute UE_j's public key $PK_j = H_2(PID_j)$, where $j = i-1, i+1$;
ii. Compute $h_j = H_3(M_j)$;
iii. Verify $e(g, \prod_j \sigma_j) \stackrel{?}{=} \prod_j e\left(PK_j, X_j \cdot K_{pub}{}^{h_j}\right)$, accepts UE_j and M_j if the verification equation holds.

After a series of verifications, UE_i computes the left key $K_i^L = X_{i-1}^{x_i}$, the right key $K_i^R = X_{i+1}^{x_i}$ and its second key hint $Y_i = \frac{K_i^R}{K_i^L}$. Then, it sets the second key agreement message $M_i{}' = SID_I|PID_i|2|Y_i$ and signs $M_i{}'$ as below:

i. Pick a random value $y_i \in \mathbb{Z}_q^*$;
ii. Compute $U_i{}' = g^{y_i}$, $h_i{}' = H_3(M_i{}', U_i{}')$, and $V_i{}' = PK_i{}^{y_i} \cdot SK_i{}^{h_i{}'}$;
iii. Sign a signature on message $M_i{}'$ as $\sigma_i{}' = (U_i{}', V_i{}')$.

Finally, UE_i broadcasts $M_i{}'|\sigma_i{}'$ to all the other group users UE_j, where $j = 1,2, \dots, n, j \neq i$.

In ***Key Generation*** as shown in Fig. 4**,** upon receiving all second key agreement messages $M_i{}'|\sigma_i{}'$ from UE_j ($j = 1,2, \dots, n, j \neq i$), UE_i firstly verifies all signatures in a batch as below:

i. Compute UE_j's public key $PK_j = H_2(PID_j)$;
ii. Compute $h_i{}' = H_3(M_i{}', U_i{}')$;
iii. Verify $e(g, \prod_{j\neq i,1}^{n} V_i{}') \stackrel{?}{=} \prod_{j\neq i,1}^{n} e\left(PK_j, U_i{}' \cdot K_{pub}{}^{h_i{}'}\right)$ and accept UE_j and M_j if the verification equation holds.

After verification, UE_i computes the session key $SSK_i^{SID_I}$ the same as PPAKA-HMAC does.

6) Group Session Activation

The group session activation is the same as that of PPAKA-HMAC.

D. Key Update

Due to the dynamism of a D2D group session, the session key should be updated when the session status changes. We categorize the status change into four scenarios, in which the group session key should be updated:

Scenario 1: In this scenario, the session key is expired before the group session membership changes. There are no new users joining into the group and no group members leaving the group. In this case, SN selects a random $r'$and sends it to all group users via secure channels. Upon receiving r', the group members generate a new session key by hashing the current session key SSK^{SID_I} with r', i.e., $SSK_i^{SID_I{}'} = H_1(r', SSK_i^{SID_I})$. *Group Session Activation* can be applied to initiate the usage of the new session key. Each member sends the new seed $Seed_i^{SID_I{}'} = H_1(SSK_i^{SID_I{}'})$ of the new session key to SN to check and active the new group session.

Scenario 2: In this scenario, the group membership is changed before the session key is expired. One or a set of group members leave from the group but no new user joins. In this case, the leaving group members report SN about their leaving. SN removes them from the group and selects a new random r' and sends it to all group members remaining in the group via secure channels. Once receiving r', each member computes the new session key by hashing current session key with r', i.e., $SSK_i^{SID_I{}'} = H_1(r', SSK_i^{SID_I})$. After the new session key generation, *Group Session Activation* is performed to activate the new session.

Scenario 3: In this scenario, the group membership is changed before the session key is expired. Some new users join the group communications with no group member leaving. There are two sub-scenarios based on the number of new joint users.

i. If only one new user joins the group, it firstly reports SN about its join. SN informs group members the join of the new user and selects a random $r^{'}$ and sends it to group members. Upon receiving $r^{'}$, group members update session key by hashing the current session key $SSK_i^{SID_I}$ with

r', which is computed as $SK_i^{SID_{I'}} = H(r', SK_i^{SID_I})$. The new member communicates with one of original group members to build a temporal secure channel using DHKE. Then the new member gets the new session key $SK_i^{SID_{I'}}$ from the original group member through the secure channel.

ii. If there are more than one new users join the group, they firstly report SN about their join. SN informs group members the join of new users and selects a random r' and sends it to group members. These new users firstly establish a secure group by adopting our protocols under the control of SN. After the new group building up, one user in the new group communicates with one of original group members to build a temporal secure channel using DHKE. Then the new member gets the new session key $SK_i^{SID_{I'}}$ from the original group member through the secure channel. Thus, the new group can merge into the original group with a new common session key $SK_i^{SID_{I'}}$.

After the new session key generation and new users join, *Group Session Activation* is performed to activate the new session.

Scenario 4: In this scenario, the group membership is changed before the session key is expired. Some new users join the group communications with some old group members leaving. In this case, we apply the approach described in Scenario 2 first to set up a new group that keeps the rest old group members. Then, we apply the approach used in Scenario 3 to allow the new members to join the above new group.

E. Security Analysis on PPAKA-IBS

In this subsection, we discuss security issues of the proposed PPAKA-IBS protocol according to our security objectives.

1) Internal Attack Resistance

PPAKA-IBS can resist malicious users to launch impersonation attacks in the group. It is able to trace the user who did not follow the protocol and reveal the real identity linked to its pseudonym.

Proof: We prove the internal attack resistance of PPAKA-IBS by following the idea of [29] in a Universal Composability (UC) framework [30].

Claim 2: The construction of PPAKA-IBS achieves the security requirement of Authenticated Key Exchange Universal Composability (AKE-UC) compiler proposed in [29].

Proof of Claim:

In Initialization Phase of the compiler, each player generates long-term verification/signing keys. Similarly, in our protocol, D2D users get their public/private key pairs from SN in the user registration phase. In the session establishment phase of our protocol, UE_i generates key hints X_i, Y_i and session key SSK^{SID_I} using some public parameters, e.g., g, and private information, e.g., x_i. These operations satisfy the definition in AKE-UC compiler, i.e.,

$$ack_i = F_{sk_i}(v_0) \text{ and } sk_i' = F_{sk_i}(v_1),$$

where v_0 and v_1 are two public parameters, F is a collision-resistant Pseudo Random Function (PRF), sk_i is private information of UE_i. In our protocol, $v_0 = v_1 = g$, ack_i indicate key hints X_i, Y_i and sk_i' indicate the session key SSK^{SID_I}.

Then, a signature is signed in the compiler to generate a signature for authentication as follows:

$$\sigma_i \leftarrow Sign_{SK_i}(UE_i, SID_I, PID_i, ack_i),$$

where SK_i is the private key of UE_i. In PPAKA-IBS, UE_i signs M_i and M_i' using its private key and sends σ_i and σ_i' to other UEs for authentication. PPAKA-IBS achieves the security requirement of AKE-UC compiler.

2) Group Backward Secrecy

Since both of our two protocols support dynamic group management. Backward secrecy should be ensured that new joining users are unable to know previous session key.

Proof: In our protocols, if one user joins the group, a key update algorithm will be triggered to generate a new session key for the new group, as Scenario 3 described in Subsection D. Since the new session key is generated by hashing the previous session key and a random getting from SN, i.e., $SSK_i^{SID_{I'}} = H(r', SSK_i^{SID_I})$, only the users in the group can generate the new session key. If a new user joins the group, a secure channel is built up between it and a group member, then the new key is transmitted to the new user. The new user has no knowledge of previous session key and the random. The hardness of breaking the group backward secrecy of our protocols can be reduced to breaking hash function.

3) Group Forward Secrecy

Forward secrecy ensures that the group user who have left the group session will have no ability to get the further session key.

Proof: In our protocols, after a group user left, it is hard for it to infer the new session key from the previous session key owned by it. Since the new session key is generated by hashing the previous session key and a random getting from SN, i.e., $SSK_i^{SID_{I'}} = H(r', SSK_i^{SID_I})$, it's impossible for the leaving user and other attackers to get the random r' from SN. The random is only distributed by SN to the users that are still in the D2D group via security channels. Thus, the hardness of breaking the group forward secrecy of our protocols can be reduced to break the security channel between SN and users and the hardness of the hash function.

V. Performance Evaluation

In this section, we firstly analyze the performance of our protocols in terms of computation complexity and communication cost. Then, we simulate the proposed protocols, evaluate their performance and compare it with related works [10, 13] to demonstrate their effectivity.

A. Performance Analysis

We analyze the efficiency of our protocols in terms of the computation complexity and communication overhead, respectively.

1) Computation Complexity

TABLE. I COMPUTATION COMPLEXITY OF PPAKAS

	Phase	Protocol	Operations	Computation Complexity
SN	System Setup	HMAC	-	-
		IBS	1*Rand+1*Exp	$\mathcal{O}(1)$
	User Registration	HMAC	-	-
		IBS	N(1*Hash+1*Exp)	$\mathcal{O}(N)$
	Session Request	HMAC	1* Rand	$\mathcal{O}(1)$
		IBS	-	-
UE	Session Establishment --Round 1	HMAC	1*Rand+1*Exp+1*Hmac	$\mathcal{O}(1)$
		IBS	1*Rand + 3*Exp +1* Hash + 1* Mul	$\mathcal{O}(1)$
	Session Establishment --Round 2	HMAC	2*Exp+3*Hmac+1*Mul	$\mathcal{O}(1)$
		IBS	1* Rand + 5*Hash + 3 Pair + 7 Exp + 6 * Mul	$\mathcal{O}(1)$
	Session Establishment ---Key Generation	HMAC	(n-1) *Hmac + (2n-2)*Mul	$\mathcal{O}(n)$
		IBS	(2n-2) * Hash + (n-1) * Exp + (n-1)*Pair + (5n - 7) * Mul	$\mathcal{O}(n)$
	Group Session Activation	HMAC	1*Hash	$\mathcal{O}(1)$
		IBS	1*Hash	$\mathcal{O}(1)$
Total (SN+UE)		HMAC	2*Rand+3*Exp+(5+n)*Hmac+(2n-1)*Mul+1*Hash	$\mathcal{O}(\mathrm{n})$
		IBS	3*Rand+(n+10)*Exp+5n*Mul+(2n+6)*Hash+(n+2)*Pair	$\mathcal{O}(\mathrm{n})$
SeDS [10]			[n*(n-1)/2]*(3*Rand+5*Exp+2*Hash+4*Pair)	$\mathcal{O}(\mathrm{n}^2)$
NA-GD2C [13]			[n*(n-1)/2]*(8*Exp+14*Hash+2*Mul+3*Pair)	$\mathcal{O}(\mathrm{n}^2)$

Note: Rand, Exp, Hash, Hmac, Mul and Pair denote random selection, exponent operation, hash function operation, HMAC operation, multiplication and pair operation, respectively. N is the number of registered users in the system. n is the number of users participating in the D2D group.

TABLE. II COMMUNICATION OVERHEAD COMPARISONS

Protocol	Communication Overhead
PPAKA-HMAC	$97*n^2+189n$
PPAKA-IBS	$97*n^2+229n$
SeDS	$[n*(n-1)/2]*306=153*n^2-153n$
NA-GD2C	$[n*(n-1)/2]*845=422*n^2-422$

In the two proposed protocols, both involve two types of system entities: UE and SN. We analyze the computation cost of each, respectively. In both protocols, SN is in charge of the system setup, user registration and session request. But the computation overheads of system setup and user registration in PPAKA-HMAC and session request in PPAKA-IBS are constant and trivial, we ignore the computation cost of these operations in our analysis. TABLE I shows the concrete analysis on computation operations and complexity of both protocols. We can see from the table that the computation complexity of User Registration in protocol PPAKA-IBS is $\mathcal{O}(N)$, which means the computation overhead of SN increases with the number of registration users in the system. In User Registration, PPAKA-HMAC shows advanced computation performance since its computation complexity in this phase is $\mathcal{O}(1)$. In Session Establishment -- Key Generation phase, the computation complexity of both PPAKA-HMAC and PPAKA-IBS is $\mathcal{O}(n)$, which means the computation overhead of each UE for key generation increases with number of users join in the D2D group. We compared our protocols with SeDS [10] and NA-GD2C [13] that are used in a similar application scenario to our protocols, shown in TABLE I. Since both SeDS and NA-GD2C were proposed to address two-user D2D communications, we assume that if n D2D users want to communication with each other, $[\frac{n*(n-1)}{2}]$ two-user D2D communication sessions should be establishment. So, the computation complexity of SeDS and NA-GD2C are both $\mathcal{O}(\mathrm{n}^2)$, but both of our protocols' is $\mathcal{O}(\mathrm{n})$.

2) Communication Overhead

In order to analyze the communication cost introduced by the proposed protocols, we first define the size of each parameter transferred in the protocols. We set the size of both RID and PID as 16 bytes, SID as 8 bytes. For K_{pub}, $K_{SID_I}^{mac}$, PK_i/SK_i, X_i/Y_i, σ_i/σ_i' and SSK^{SID_I}, their sizes are all set as 20 bytes. We classify the communication overhead into two types, i.e., the communications between SN and UE and the communications among UEs. We analyze the communication overhead in each phase. In the User Registration phase, communication overhead in PPAKA-HMAC is 60 bytes, which is less than PPAKA-IBS. The reason is that in PPAKA-IBS, each UE gets its key pair from SN and each key is 20 Bytes. However, P_{pub}, which is 20 Bytes, is not needed in PPAKA-HMAC. Another difference occurs in the Session Request phase. UE in PPAKA-HMAC gets an extra HMAC key from SN, thus the communication overhead between SN and UE in PPAKA-HMAC is 20 Bytes more than PPAKA-IBS. We can observe that in both protocols, the communication overheads in all phases increase with the growth of the number of users in the group. Except for the communication overhead in session request and in Round 2, all other overheads are linearly related to the number of users. In the session request phase, for each UE, the overhead between SN and UE is (32n+28) in PPAKA-HMAC and (32n+8) in PPAKA-IBS. In Round 2, each UE sends its key agreement message and signatures to other n-1 pieces of UE. The total size of one message and signature is 65 bytes, thus the communication overhead of each UE is 65(n-1) bytes and the overhead of the whole system is 65n(n-1) bytes, which is linearly changed with n^2. We summarize the whole communication overhead of our protocols and compare them with SeDS and NA-GD2C in TABLE. II. As described previously, we multiply $[\frac{n*(n-1)}{2}]$ by the actual overhead of SeDS and NA-GD2C to simulate n users D2D communications. The result shows that our two protocols have lower communication overhead than existing two protocols. PPAKA-HMAC performs better than PPAKA-IBS with regard to communication overhead.

B. Simulation and Evaluation

Furthermore, we simulated the proposed protocols and tested their performance on a laptop. The laptop runs 32-bit CentOS Linux 6.0 with 2.5 GHz Inter Core I5 Quad-processor CPU and 2GB RAM [6]. Herein, we applied PBC [31] for algebraic operations and OpenSSL [32] for secure communication transmission.

We tested the operation time of five main steps in our protocols, which are System Setup, User Registration, Round 1, Round 2 and Session Key Generation. Since the user number of a D2D group affects the operation time of different steps, we set the group size (user number) from 5 to 100 with 5 as an increment. In these five main steps, the execution of System Setup, Round 1 and Round 2 are not affected by the number of group users. The computation cost for these three setups are constant, which are about 15.6 millisecond (ms), 12.6ms and 22.7ms in PPAKA-HMAC and 23.6ms, 15.8ms, 65.2ms in PPAKA-IBS. In these operations, the most expensive operation is signature verification in Round 2. In PPAKA-HMAC, the operation time of HMAC verification is about 8.0ms. In PPAKA-IBS, it takes about 43.1ms to verify two signatures in an aggregated way. If there is no batch verification applied, the operation time of one signature verification is about 24.4ms and about 48.8ms for two.

We also tested the performance of User Registration and Session Key Generation. As shown in Fig. 5, the black line denotes the change of operation time of user registration in PPAKA-IBS. It increases linearly with the number of users registered into the system. The performance of Session Key Generation can be analyzed from two parts: verification and key generation. The blue line in Fig. 5 shows the operation time of HMAC verification in PPAKA-HMAC. The red line and green line show the operation time of signature verification with or without aggregation in PPAKA-IBS. The results show that PPAKA-HMAC has better performance than PPAKA-IBS. Batch verification can reduce the cost in signature verification to some extent. The performance of key generation operation in Session Key Generation is lightweight comparing with signature verification. Its cost increases linearly from 0.14ms to 3.47ms when the number of users increases from 5 to 100. The reason is that only a few multiplication operations are performed in the key generation compared with the signature verification, which consists of many pairing operations.

We further compared the performance of our protocols with SeDS [10] and NA-GD2C [12] in Fig. 6. The results demonstrate that our designed protocols perform much more efficient than SeDS and NA-GD2C. According to our theoretical analysis, the operation time of our protocols increases linearly with the number of group members. However, the operation time of SeDS and NA-GD2C increases with complexity $O(n^2)$, which performs much worse than our two protocols. PPAKA-HMAC achieves the best computational efficiency among these four protocols because of the usage of HMAC. PPAKA-IBS provides optimal security but with a tradeoff on performance.

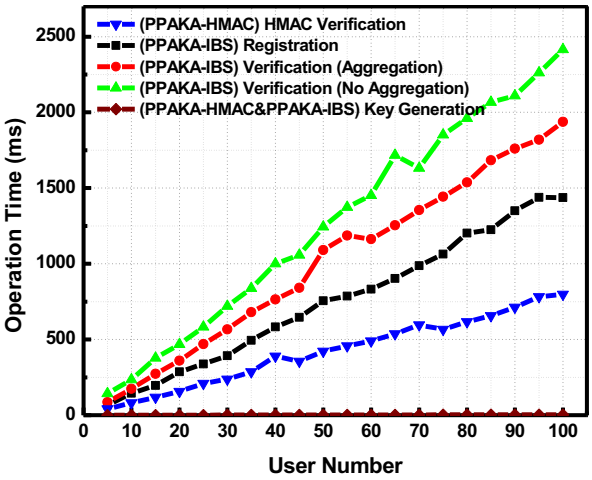

Fig. 5 Operation time of registration and verification in PPAKAs

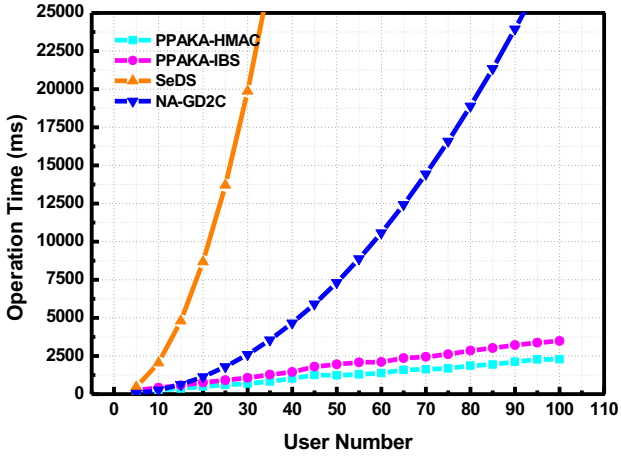

Fig. 6 Comparison of operation time of different protocols

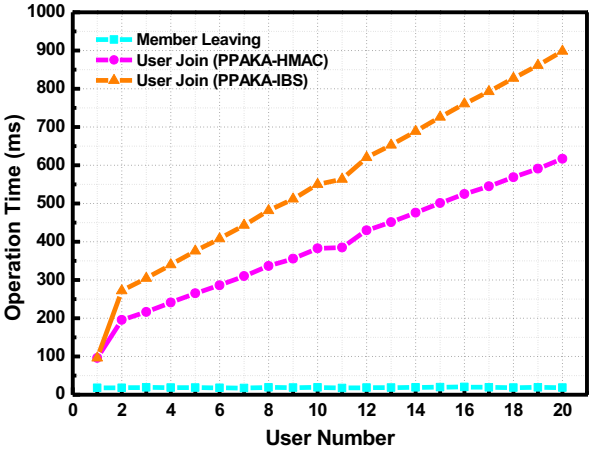

Fig. 7 Operation time of key update in different scenarios

We also evaluated the performance of key update of our protocols. In Scenario 1, key update is conducted due to old key expiration and the operation time of new key generation is constant about 17.8ms, which is not related to the number of group members. Fig. 7 shows the operation time of key update in Scenario 2 and Scenario 3. In Scenario 2, key update is caused by member leaving. The operation time of key update in this scenario is also constant without changing with the number of leaving members. This is because that no matter how many

members leave the group, the operation of key update for remaining members is running a hash function. In Scenario 3, key update is caused by user joining. Fig. 7 shows the change of operation time of key update with the number of joining users in both PPAKA-HMAC and PPAKA-IBS. We can see that the operation time of both protocols increases linearly with the number of joining users except one user joining case. The reason is that, in one user joining case, the joining user joins the group by only conducting DHKE with one existing member. No new group is built. When more than one users join the group, a new group should be built firstly, which makes the operation time increases linearly with the number of joining users. In Scenario 4, since the operations for old member leaving and new user joining are independent and operated in turn, the operation time of key update in this scenario is the addition of that in Scenario 2 and Scenario 3. Thus, it is not shown in Fig. 7.

With regard to the success rate of group session establishment, we think it is meaningful to test it in a real D2D communication system since it is impacted by many factors in reality (e.g., environmental factors, bandwidth, available spectrum, battery level and computation capacity of UE, etc.). Some important impact factors are not related to protocol design. This paper work focuses on secure D2D group communication protocol design. Security and operation performance evaluation is our focus.

VI. Conclusion

In this paper, we proposed two privacy-preserving authentication and key agreement protocols for group D2D communications. The first protocol PPAKA-HMAC helps a group of D2D users establish a secure D2D group session without leaking their identity privacy. It is secure against external malicious attackers with lightweight operations. The second protocol PPAKA-IBS can establish secure D2D group communications by providing better security than PPAKA-HMAC in terms of resisting internal attacks. Formal security analysis and extensive experimental test showed the security, efficiency and effectiveness of our protocols.

With regard to future work, we will further test the success rate of group session establishment in a 5G D2D communication test bed in order to evaluate the real applicability of our proposed protocols.

Acknowledgement

This work is sponsored by the NSFC (grants 61672410 and U1536202), the National Key Research and Development Program of China (grant 2016YFB0800704), the Project Supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016ZDJC-06), the 111 project (grants B08038 and B16037), and Academy of Finland (grant 308087).

References

[1] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Communications Magazine,* vol. 47, no. 12, pp. 42-49, 2009.

[2] P. Janis, C. H. Yu, , K. Doppler, C. Ribeiro, C.Wijting, K. Hugl, O. Tirkkonen, and V. Koivunen., "Device-to-device communication underlaying cellular communications systems," *International Journal of Communications, Network and System Sciences,* vol. 2, no. 3, p. 169-178, 2009.

[3] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal,* vol. 10, no. 3, pp. 1172-1182, 2016.

[4] L. Militano, G. Araniti, M. Condoluci, I. Farris, and A. Iera, "Device-to-device communications for 5g internet of things," *EAI Endorsed Transactions on Internet of Things,* vol. 1, no. 1, pp. 1-15, 2015.

[5] M. Wang and Z. Yan, "A Survey on Security in D2D Communications," *Mobile Networks and Applications,* vol. 22, no. 2, pp. 195–208, 2017.

[6] M. Wang, Z. Yan, and V. Niemi, "UAKA-D2D: Universal Authentication and Key Agreement Protocol in D2D Communications," *Mobile Networks and Applications,* vol.22, no. 3, pp. 510–525, 2017.

[7] W. Shen, W. Hong, X. Cao, B. Yin, D. M. Shila, and Y. Cheng, "Secure key establishment for device-to-device communications," in *Proc. 2014 IEEE Global Communications Conference (GLOBECOM 2014),* pp. 336-340, IEEE.

[8] L. Goratti, G. Steri, K. M. Gomez, and G. Baldini, "Connectivity and security in a D2D communication protocol for public safety applications," in *Proc. 11th International Symposium on Wireless Communications Systems (ISWCS 2014),* pp. 548-552, IEEE.

[9] H. Kwon, C. Hahn, D. Kim, K. Kang, and J. Hur, "Secure device-to-device authentication in mobile multi-hop networks," in *Proc. International Conference on Wireless Algorithms, Systems, and Applications*, 2014, pp. 267-278, Springer.

[10] A. Zhang, J. Chen, R. Q. Hu, and Y. Qian, "SeDS: Secure data sharing strategy for D2D communication in LTE-Advanced networks," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 4, pp. 2659-2672, 2016.

[11] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems," *IEEE Transactions on Information Forensics and Security,* vol. 12, no. 3, pp. 662-675, 2017.

[12] R.-H. Hsu and J. Lee, "Group anonymous D2D communication with end-to-end security in LTE-A," in *Proc. 2015 IEEE Conference on Communications and Network Security (CNS 2015),* pp. 451-459, IEEE.

[13] R.-H. Hsu, J. Lee, T. Q. Quek, and J.-C. Chen, "GRAAD: Group Anonymous and Accountable D2D Communication in Mobile Networks," *IEEE Transactions on Information Forensics and Security,* vol. PP, no. 99, pp. 1-1, 2017.

[14] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory,* vol. 22, no. 6, pp. 644-654, 1976.

[15] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *Proc. IEEE Symposium on Security and Privacy, (SP '07),* pp. 321-334, IEEE.

[16] H. Xiong and Z. Qin, "Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks," *IEEE Transactions on Information Forensics and Security,* vol. 10, no. 7, pp. 1442-1455, 2015.

[17] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Noninteractive pairwise key establishment for sensor networks," *IEEE Transactions on Information Forensics and Security,* vol. 5, no. 3, pp. 556-569, 2010.

[18] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wirelessbody area networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, no. 2, pp. 332-342, 2014.

[19] H. Xiong, "Cost-effective scalable and anonymous certificateless remote authentication protocol," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 12, pp. 2327-2339, 2014.

[20] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su, "An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications," *IEEE Transactions on Vehicular Technology,* vol. 59, no. 7, pp. 3589-3603, 2010.

[21] A. Wasef and X. Shen, "EMAP: Expedite message authentication protocol for vehicular ad hoc networks," *IEEE Transactions on Mobile Computing,* vol. 12, no. 1, pp. 78-89, 2013.

[22] J. Li, H. Lu, and M. Guizani, "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs," *IEEE Transactions on Parallel and Distributed Systems,* vol. 26, no. 4, pp. 938-948, 2015.

[23] A. Castiglione, P. D'Arco, A. De Santis, and R. Russo, "Secure group communication schemes for dynamic heterogeneous distributed computing," *Future Generation Computer Systems,* vol. 74, no. Supplement C, pp. 313-324, 2017.
[24] A. Castiglione, A. De Santis, B. Masucci, F. Palmieri, A. Castiglione, and X. Huang, "Cryptographic hierarchical access control for dynamic structures," *IEEE Transactions on Information Forensics and Security,* vol. 11, no. 10, pp. 2349-2364, 2016.
[25] A. Castiglione, A. De Santis, B. Masucci, F. Palmieri, X. Huang, and A. Castiglione, "Supporting dynamic updates in storage clouds with the Akl–Taylor scheme," *Information Sciences,* vol. 387, pp. 56-74, 2017.
[26] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on architecture enhancements to support Proximity-based Services (ProSe); Stage 2, 3GPP TS 23.3034, V15.0.0 (2017-06).
[27] R. Dutta and R. Barua, "Provably secure constant round contributory group key agreement in dynamic setting," *IEEE Transactions on Information Theory,* vol. 54, no. 5, pp. 2007-2025, 2008.
[28] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Advances in Cryptology (CRYPTO 2001)*, pp. 213-229, Springer.
[29] J. Katz and J. S. Shin, "Modeling insider attacks on group key-exchange protocols," in *Proc. the 12th ACM conference on Computer and communications Security (CCS 2005)*, pp. 180-189, ACM.
[30] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. the 42nd IEEE Symposium on Foundations of Computer Science*, 2001, pp. 136-145, IEEE.
[31] PBC: The pairing-based cryptography library. https://crypto.stanford.edu/pbc/.
[32] OpenSSL, "SSL/TLS Toolkit," *The document is available in http://www. openssl. org,* 2011.

Mingjun Wang received the BSc degree in communication and information systems from the Henan Normal University, Xinxiang, China, 2011. He is currently pursuing PhD degree in information security at the Xidian University, Xi'an, China. His research interests are in security, privacy and trust management in the next generation mobile communication networks and wireless systems, social networking and cloud computing.

Zheng Yan (M'06, SM'14) received the BEng degree in electrical engineering and the MEng degree in computer science and engineering from the Xi'an Jiaotong University, Xi'an, China in 1994 and 1997, respectively, the second MEng degree in information security from the National University of Singapore, Singapore in 2000, and the licentiate of science and the doctor of science in technology in electrical engineering from the Helsinki University of Technology, Helsinki, Finland in 2005 and 2007. She is currently a professor at the Xidian University, Xi'an, China and a visiting professor at the Aalto University, Espoo, Finland. She authored more than 170 peer-reviewed publications and solely authored two books. She is the inventor of over 60 patents and patent applications. Her research interests are in trust, security and privacy, social networking, cloud computing, networking systems, and data mining. Prof. Yan serves as an organization and program committee member for numerous international conferences and workshops. She is also an associate editor or a guest editor of many reputable journals, e.g., Information Sciences, Information Fusion, IEEE Systems Journal, IEEE Internet of Things Journal, IEEE Access, ACM TOMM, etc. She is a senior member of the IEEE.