2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)
September 6-7, 2017; Shiraz University - Shiraz, Iran

A Lightweight and Secure Data Sharing Protocol for D2D Communications

Atefeh Mohseni-Ejiyeh
Department of Computer Engineering
University of Isfahan, Isfahan, Iran
Email: atefeh_mohseni@eng.ui.ac.ir

Maede Ashouri-Talouki
Department of Computer Engineering
University of Isfahan, Isfahan, Iran
Email: m.ashouri@eng.ui.ac.ir

Mojtaba Mahdavi
Department of Computer Engineering
University of Isfahan, Isfahan, Iran
Email: m.mahdavi@eng.ui.ac.ir

***Abstract*—D2D communications empower operators to offer their services at the highest level of quality provided that issues concerning availability and security are addressed first. The explosive amount of mobile data traffic, on one hand, and the growing demand for available services on the other hand, motivate us to propose a secure, lightweight and available data sharing scheme for D2D communications. Data sharing, an increasingly popular service among mobile users, could play a noticeable role in offloading the traffic data from operators if handled by D2D communications. In this paper, we propose an efficient protocol for secure data sharing in D2D communication. In the proposed protocol, the major security challenges about data sharing like, data confidentiality, integrity, detecting message modification, and preventing the propagation of malformed data are considered. Additionally, not only unauthorized users are banned from using our service, but also by keeping records about the history of the authorized users actions, we are able to punish misbehaving users, if their malicious behavior exceeds a threshold. The evaluation of the proposed protocol proves that it is more lightweight than the previous works and supports the security requirements of data sharing scheme.**

***Index Terms*—D2D communications, 5G, mobility, traffic offloading, security,lightweight, data sharing**

I. Introduction

Given the anticipated growth of mobile traffic in cellular networks by the arrival of the fifth generation of mobile networks (5G) [1], the demand for traffic offloading approaches becomes an inevitable problem for mobile operators. Among the several approaches proposed to address this problem [2], device-to-device (D2D) communication appears to be a satisfactory solution [3].

D2D communication refers to direct communication between devices in a cellular network, established either under the control of operators or directly by the users[4]; the operator has zero involvement at the user plane side[1]. So connecting devices directly to each other -through D2D communication- is inherently exposed to certain security and privacy vulnerabilities [5]. With respect to the variety of services and applications that can utilize D2D communications and the importance of the security and availability for such services, we propose a secure data sharing scheme to connect adjacent users securely, to get their intended multimedia data.

Data sharing, as a popular application among users, especially for sharing multimedia [1], can have a striking effect on traffic offloading if D2D technologies are used in cellular networks [3]. However, despite the advantages D2D communication for services such as video live streaming [6] or data sharing, only a few efforts have been made to address the related security and availability issues.

We modified an authentication and key agreement scheme of the Evolved Packet System (EPS-AKA) [7] to authenticate users based on their unique identities in a cellular network. Then, the required secret keys are generated by using the shared secret session key which is obtained from a Key Derivation Function (KDF) [8]. Therefore, we can achieve our security goals like data confidentiality and integrity, resistance to message fabrication, replay and man-in-the-middle attacks with the ability to detect any malicious user behavior in the proposed protocol. The contributions of our proposed protocol are (1) all the security requirements of data sharing are considered while the users do not need to register in the system to certify based on their public keys[2]; (2) the computation costs on users are very lightweight; and (3) it is compatible with users mobility.

The rest of the paper is organized as follows. We will review the literature in Sec. II, the security preliminaries of the proposed protocol will be explained in Sec. III, the proposed secure data sharing protocol is presented in Sec. IV, its performance and security will be analyzed in Sec. V and finally we conclude the paper in Sec. VI.

II. Related Works

A large number of studies on D2D communications[4] have been reported, most of which represent the challenges of interference management, resource allocation or peer discovery[9]-[10]. Despite the importance of security issues in D2D communication, only few works have focused on the topic.[11]

A classification provided by Tehrani et al.[12] defines four types for D2D communication: (1) device relaying with operator-controlled link establishment (DR-OC); (2) direct D2D communication with operator-controlled link establishment (DC-OC); (3) device relaying with device-controlled link

[1]Generally the term user plane refers to the transmission of user's data packet in LTE

[2]To have a USIM Which already exists on their SIM card is enough to get their integrity and confidentiality keys used in our protocol

establishment (DR-DC); and (4) direct D2D communication with device-controlled link establishment (DC-DC). The related security works for each type will be discussed in the following.

The authors in [13] present a secure data transmission protocol for mobile health systems by exploiting D2D communications in a DR-DC scenario. Data confidentiality, integrity, mutual authentication and unforgeability are achieved by using a CLGSC[3] scheme which adaptively works in each signcryption, signature or encryption modes. However, [13] is an application-oriented scheme which is limited for mobile health application and the security requirements of data sharing application like resistance to free-ridding attacks are not considered.

In [14], a secure solution to connect users via multi-hop communications proposed for emergency services like public safety. In their scheme, D2D communication between users could be established either in a DR-OC scenario or DR-DC. Although the proposed scheme in [14] ensures confidentiality, integrity, and availability, it is better suited to multi-hop communication among ProSe[4] enabled devices.

Zhang et al.[15] proposed a secure data sharing scheme, for a DC-OC scenario, which guarantees availability, data confidentiality and integrity. They use a public key-based cryptographic algorithm to achieve user authentication and a message authentication code (MAC) for data origin; however, this imposes communication and computation overhead on the users. In addition, in their scheme, devices are considered stationary and must register with eNB to obtain their certificate. Therefore, as evident, the scheme is impractical in cases which users are mobile (which happens frequently in mobile networks).

In [16], an AKA protocol was proposed to establish a secure connection between D2D users using an EPS-AKA[7] called UAKA. For the first time, mobility scenarios are considered in LTE-A networks such as inter-operator and roaming. Although the authors claim resilience against cryptographic attacks such as replay and MitM[5], we have found that UAKA suffers from MitM attacks. According to the MAC protection over the key hint message during the D2D session key generation phase where a secret key $K_M = R_p \oplus R_k$ is used; A valid MAC key (K_M) is obtained by sniffing the secret R_p transmitted via an open channel between users and by capturing the values r_1 and $r_2 (R_k = r_1 \oplus r_2)$ where transmitted without any cryptographic protection. Therefore, the MitM attacker can violate the security of authentication and key agreement in UAKA. For more details about UAKA please refer to [16].

The EPS-AKA scheme (described in Sec III) is also modified here to authenticate users and manage their required confidentiality and integrity keys during the data sharing process. Although, the EPS-AKA is typically run per location update[7], in order to secure direct communications in this paper, it is executed per session establishment same as the approach adapted in [16]. It is worth mentioning that we consider the security analysis of EPS-AKA [17] so that the changes to the EPS-AKA can resist the security vulnerabilities revealed in [17].

In summary, our protocol is the first secure and lightweight data sharing protocol that offloads data traffic from the operators, while guaranteeing the confidentiality and integrity of the transmitted data between devices and is compatible with geographical mobility.

[3]CertificateLess Generalized SignCryption

[4]proximity services

[5]Man-in-the-Middle

III. PRELIMINARIES AND SECURITY ASSUMPTIONS

A. Network Architecture in LTE-A based D2D

The main LTE-Advanced network entities include UE, eNB, MME, S-GW, P-GW and HSS that participate in managing and securing the access of D2D users to the network. For the purpose of our service, we also consider a Service Provider (SP). Fig.1 illustrates their relationships.

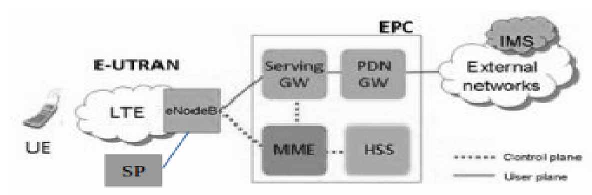

Fig. 1. The Architecture of LTE-A networks

UE: User Equipment which must be authenticated to gain access to his/her intended data via D2D communication. **MME**: Mobility Management Entity is the brain of the Evolved Packet Core (EPC) and is responsible for security procedures such as user authentication (with the help of HSS), idle management and mobility management among others. **HSS**: Home Subscriber Server has a database of subscriber identities and their private keys. To perform UE authentication and generate their authentication vector (AV), it connects to the Authentication Center (AuC). **eNB**: Evolved NodeB is a key element in E-UTRAN responsible for controlling radio resources and managing physical layer issues such as interference. Additionally, in the proposed protocol, it also contributes to authenticate users, controls the direct connection between them and also similar to [15], stores records of the users owned data and their behavior background in Table I. **GW**: S-GW is a gateway to E-UTRAN that serves the UEs by routing the incoming and outgoing IP packets. P-GW, on the other hand, is a terminal point of packet data interface to packet data network. Moreover, it is able to run the proximity service control function to detect adjacent users. **SP**: The Service Provider, in our model, is responsible to provide original data to users.

B. Authentication and Key Agreement (AKA) in LTE-A

To authenticate users, the universal circuit card (generally known as SIM card), runs the universal subscriber identity

TABLE I
RECORDED USER HISTORIES IN ENB

User's ID	Owned Data	Share frequency	Malicious behavior
ID_i	0	0	0
ID_j	P_i	0	0
ID_n	...	...	...

module (USIM) that has access to the user's permanent key (K), where is only known by USIM and AuC in the user's HSS. Therefore, HSS can generate user's AV, through the EPS-AKA algorithm [7] for each received authentication request. Generally, the EPS-AKA algorithm takes the user's permanent key (K), a random number $RAND$, a sequence number SQN and the user's serving network identity $SNID$ as input, then outputs the user's authentication vector containing $AUTN$, RES and K_{sh}. Both the user and HSS are able to use the EPS-AKA algorithm. K_{sh}, an important parameter of AV, will be used to generate user's integrity and confidentiality keys during future steps. For the purpose of our protocol, eNB and UE must go through an instance of the EPS-AKA procedure. Therefore, they can get access a pair of integrity and confidentiality keys for user's data and control signaling separately. Fig.2, depicts the above mentioned procedure. For more detail, we refer the interested readers to [7].

C. Security Assumptions

The considered security model of the proposed protocol is as follows:

Attacker Model: Attacker(s) could be either internal or external adversary(s) who participate in any malformed behaviors such as fabricating the messages, trying to repudiate their malicious behavior, prevent sharing their data with others (free ridding), deny service to other users or network element entities, either individually or by colluding with other entities.

Trust Model: The backbone entities of the network like eNB, MME and HSS are assumed to be honest enough to follow the protocol and not to be compromised by attackers. No trust relationship is assumed among the users.

Security Goals: To establish a secure communication between users in D2D communication, we propose a lightweight and secure data sharing protocol which connects adjacent users to each other in order to offload network-side traffic and also enhance the QoS.

IV. THE PROPOSED APPROACH

C. Proposed Protocol

1) System Setup: The trust authority chooses a symmetric encryption algorithm $Enc(*, k)$, an HMAC function $h(*, k)$ and a hash function of $H_1 : \{0, 1\}^* \rightarrow G$; It then publishes the system parameters(Enc, h, H_1, g, G, q) where g is a generator of G (a multiplicative cyclic group of prime order q). The data packets are indexed by their frame numbers (especially for large video files) denoted by P_i. To guarantee data originality, SP computes a signature σ_{sp} over data frames through $\sigma_{sp} = H_1(P_i \parallel M)^{x_0}$.

2) Secure Data Sharing: Users are assumed to be authenticated through an EPS-AKA before start using our service; therefore, a secret session key K_amse is shared between user and eNB. Furthermore, for those users who are subscribing to a different operator or roaming to a remote region, we utilize the secure UAKA protocol [16]. Our secure data sharing protocol for D2D communication, shown in Fig.3, consists of the following steps.

step 1: UE_i to get its intended data with the portion index (P_i), must be authenticated first. So it generates an $AV = RES, AUTH_i, RAND_i, K_{shi}$ by using the EPS-AKA algorithm. The service request is sent as $(RAND_i, AUTH_i, ID_i, P_i, T_s, h(*, K_{cpi,i}))$. $K_{cpi,i}$ represents the UE_i's integrity key in order to use at control plane side and together with $(K_{cpi,i}, K_{cpe,i})$ and $(K_{upi,i}, K_{upe,i})$ are derived from the generated secret key k_{shi}, through a key derivation function (KDF). First key pair stands for control plane and the latter for the user plane communication between UE_i and eNB.

Step2: To authenticate UE_i, eNB sends an authentication request to HSS with user's id and its chosen $RAND$ value. Following this, HSS returns user's $AV_i = XRES_i, AUTH_{hss}, RAND_i, K_{shi}$ back to the eNB.

Step 3: eNB first checks the $AUTH_{hss}$ value with the one received from UE_i, if it held, eNB sends the user's ID to GW to get the list of users that are close to UE_i. Then, in order to fairly choose a candidate from the received list, eNB refers to Table I and selects UE_j, which has the lowest share frequency.

Step 4: eNB chooses a random number $RAND_j$ and sends an authentication request for UE_j to HSS. After getting UE_j's session key (K_{shj}), it derives its keys via KDF(K_{shj}). eNB then stores the received $AV_j = XRES_j, AUTH_{hss}, RAND_j, K_{shj}$ of UE_j for further steps.

Step 5: eNB makes a response for UE_i as $(ID_i, ID_j, XRES, P_i, (k_{upi,j} \oplus k_{cpi,i}), h(*, k_{cpi,i}))$. $XRES$ stands for eNB authentication and the HMAC value $h(*, k_{cpi,i})$ ensures the message integrity. eNB to allow UE_i to verify the UE_j's message in next step, sends UE_j's integrity key ($k_{upi,j}$) which is XOR coded with UE_i's integrity key($k_{cpi,i}$).

Step 6: eNB sends a message of $(ID_j, ID_i, RAND_j, AUTH_j, P_i, h(*, k_{cpi,j}))$ to UE_j to notify it about the UE_i's request.

Step 7: By obtaining a data sharing request from eNB, UE_j first, uses the EPS-AKA algorithm to authenticate eNB (checks the received value of $AUTH_j$ with one generated itself), then, derives its keys from the shared session key (K_{shj}). UE_j encrypts the data M(related to the index P_i) through $M' = Enc(M, k_{upe,j})$, then sends the message $((ID_j, ID_i, P_i, M', \sigma_{sp}, T_s, h((ID_j, ID_i, P_i, M', \sigma_{sp}, T_s), k_{cpi,j})), h(*, k_{upi,j}))$ to UE_i. The outer HMAC value, $h(*, k_{upi,j})$, can be verified by UE_i and the inner one $(h((ID_j, ID_i, P_i, M', \sigma_{sp}, T_s), k_{cpi,j}))$ is generated to be verified by the eNB in case of any message modification reports to eNB by UE_i in the last step. T_s

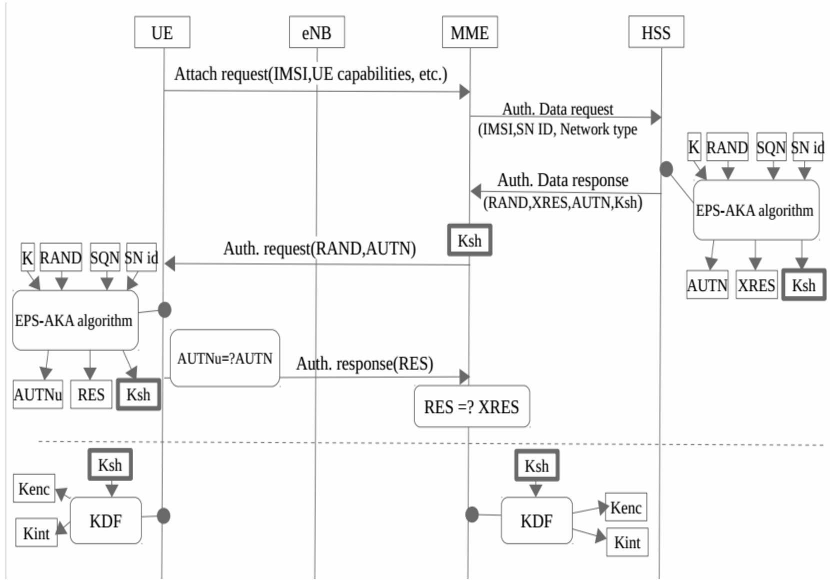

Fig. 2. An Overview of EPS-AKA algorithm [7]

applies to inform eNB about the time when the intended data was sent to UE_i(It will be checked in step 11).

Step 8: While receiving the UE_j's message, UE_i first, verifies the message integrity by using UE_j's integrity key $k_{upi,j}$ (received from eNB in step 5), which received from eNB in step 5 . After that, UE_i sends a key hint request to eNB to get the data encryption key through $((ID_i, ID_j, P_i, T_s), h(*, k_{cpi,i}))$

Step 9: When a key hint request delivered to the eNB, first, it checks the validity of the attached time stamp then checks the integrity of the message through $h(*, k_{cpi,i})$. If it verifies, instead of sending the data encryption key($k_{upe,j}$) directly to UE_i, it transmits ($k_{upe,j} \oplus k_{upe,i}$) to UE_i to prevent man in the middle attack and ensures the confidentiality of the encryption key as well. The format of the transmitted message to UE_i is as $(ID_i, ID_j, P_i, (k_{upe,j} \oplus k_{upe,i}), h(*, k_{cpi,i}))$.

Step 10: UE_i can access to the encryption key $k_{upe,j}$ by computing an exclusive or over the received key hint message form eNB, with its own confidentiality key ($k_{upe,i}$) and decrypts the M' (to get M''). Following this, it checks the validity of σ_{sp} through $\hat{e}(X_0, H_1(P_i \parallel M'')) \stackrel{?}{=} \hat{e}(\sigma_{sp}, g)$. If it is verified ($M'' = M$), sends no feedback to eNB, otherwise sends a beacon message through forwarding the received message in step 7 attached to HMAC value with the key $k_{cpi,i}$, in an allowable window time to eNB.

Step 11: If during the waiting time($T_s + \Delta T$)[15] , a beacon message received, eNB first decrypts the message M' then verifies SP's signature (σ_{sp}). If it is verified, the P_i will be added to the records of UE_i's data and the share frequency amount of UE_j will be incremented by one. Otherwise, it verifies the value of $h(*, k_{cpi,j})$ to realize who is the real sender of the fabricated data. If $h(*, k_{cpi,j})$ is verified, the malicious behavior amount of UE_j will be added, otherwise, eNB concludes that the UE_i itself maliciously pretends to receive a fabricated data, therefore its malicious behavior amount will be incremented by one by the eNB.[6]

TABLE II
THE NOTATIONS USED IN THE PROPOSED PROTOCOL

Notation	Description
$h(*, k)$	A secure HMAC function with the key k
$k_{upi,i}$	UE_i's integrity key for user plane connections
$k_{upc,i}$	UE_i's confidentiality key for user plane connections
$k_{cpi,i}$	UE_i's integrity key for control plane connections
$k_{cpc,i}$	UE_i's confidentiality key for control plane connections
P_i	The index of data
σ_{sp}	The service provider signature over an original message M
$Enc(*, k)$	A symmetric encryption algorithm with key k
T_s	Time stamp
X_0, x_0	SP's public and private keys

V. EVALUATION AND RESULTS

In this section, we show that our proposed protocol outperforms previous studies in terms of security and efficiency.

A. *Performance Evaluation of the Proposed Protocol*

The dominant computation costs of the proposed protocol is pairing execution for SP's signature verification; while the other computation overhead is purely symmetric. Each user is authenticated by an instance of EPS-AKA which consists of KDF and MAC functions [8].The most similar work to us is a secure data sharing protocol proposed by Zhang et

[6]Although in some cases it is possible that UE_j itself sends an incorrect version of HMAC over the message during step 7, but with regard to this fact that at this step UE_j has transmitted the verifiable data(M), and its resources like battery usage and etc. is used during the protocol, so there is no reason for UE_j to behave so.

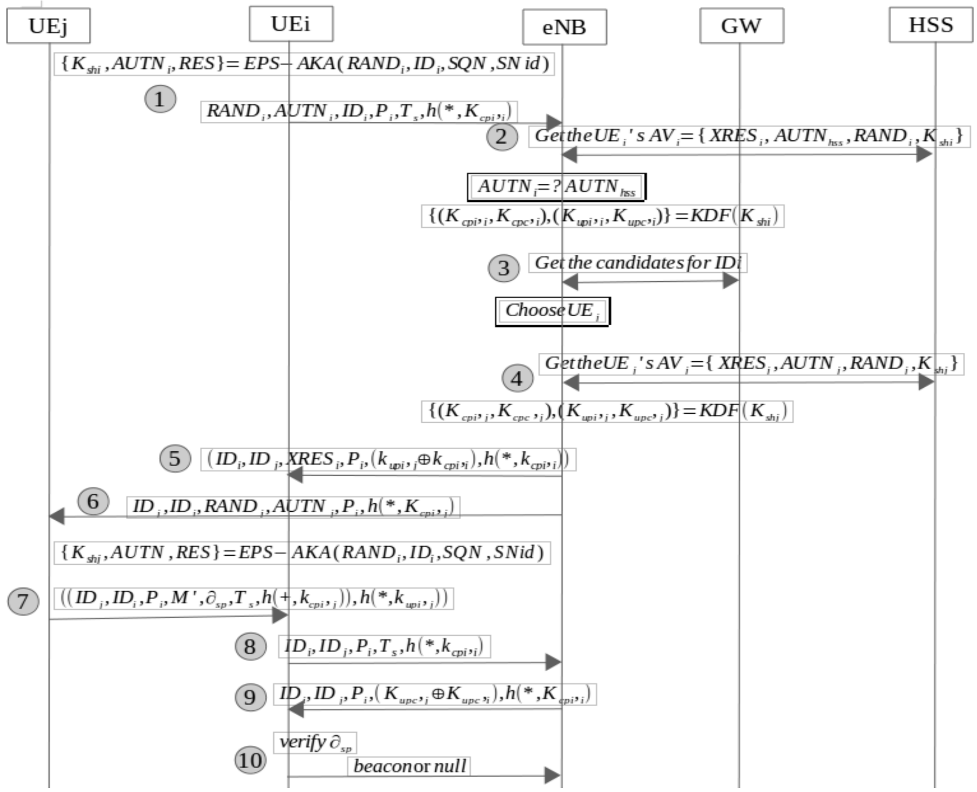

Fig. 3. The proposed protocol

al. named SeDS[15]. SeDS uses public key based digital signature for user authentication and a symmetric encryption algorithm for data confidentiality. The comparison of the overall computation cost between the proposed protocol and SeDS is tabulated in Table III.

Let time T_{mul} stands for the time of one multiplication execution in G. Therefore, according to [18], the computation cost of $T_{KDF} \approx T_{MAC} \approx 0.36Tmul$. The expensive computational costs are $T_{pair} \approx 22.5T_{mul}$ and $T_{exp} \approx 3T_{mul}$[15]. So as listed in Table III, the overall computation cost of HSS and eNB in our scheme is about $4.68T_{mul}$ while the computation cost of eNB in SeDS is about $7.8T_{mul}$. The computation cost of UE_i in our scheme is about $25.74T_{mul}$ whereas it is about $49.8T_{mul}$ in the SeDS. Finally, UE_j spends $2.52T_{mul}$ for computation cost while in the SeDS, it is about $29.22T_{mul}$. So it is blindingly obvious that the computation cost of the proposed protocol is more efficient than the SeDS.

B. Security Verification of the Proposed Protocol

To investigate the security of the proposed protocol, we will assess how our protocol meets the security requirements related to our research context and it resists to attacks.

1) Authentication and Key Agreement: We utilize an instance of the EPS-AKA algorithm to authenticate users in our protocol; thus, given the secrecy and uniqueness of the users permanent key (K), which is only known by the user and its subscriber, there is no way to generate a verifiable version of AV without having any information about the users private key (K). Therefore, only the user and its subscriber can get access to the derived integrity and confidentiality keys during our data sharing protocol.

2) Confidentiality and Integrity: Confidentiality must be guaranteed for the user's data both in the key agreement and data sharing phases. Based on EPS-AKA algorithm, the confidentiality of the cipher key agreement is guaranteed and by encrypting the data through $Enc(M, k_{upc,j})$ in step 7, the confidentiality of the data is ensured as well. In addition, the key hint response in step 9, is sent over the XOR-codded version(similar to [19] and [16]), which gives no information about the data sharing key to adversaries. To satisfy integrity demands in our protocol, all transmitted messages are attached to a verifiable HMAC value by using the user's integrity keys both on control and data signaling. However, the UE_j's integrity key for the user plane $(k_{upi,j})$ is shared with UE_i, in the proposed protocol, as a result of using the integrity key $k_{cpi,j}$ over the message transmitted in step 8 (which is only known by UE_j and eNB). any modification on the data by either adversaries or UE_i is revealed. The attack resistance of the proposed protocol is described bellow:

3) Man-in-The-Middle Attack: If an adversary, sniffs the transmitted message between user and eNB, he is not able to get access to K_{sh} or replace it with another valid one, because just $AUTN$ and RES values of AV are sent without protection and no key is transmitted directly. Note that the

TABLE III
THE COMPARISON OF COMPUTATIONAL OVERHEAD OF OUR PROTOCOL AND SEDS

entity	Proposed Protocol	SeDS
HSS	$4MAC + 2KDF$	0
eNB	$5MAC + 2KDF + 1XOR$	$5MAC + 2EXP + 1DEC$
UE_i	$7MAC + 2KDF + 1PAIR + 1DEC + 1XOR$	$5MAC + 2PAIR + 1EXP + 1DEC$
UE_j	$5MAC + 2KDF + 1ENC$	$2MAC + 2ENC + 2EXP + 1PAIR$

EXP: exponential computation ,PAIR: one pairing execution

values of $AUTN$ and RES can be used once for only one session and expire at the end of the protocol.

4) Replay Attack: In the proposed protocol, most of the messages are attached with a time stamp value. Therefore, repeated messages are dropped. In addition, the $RAND$ value used in EPS-AKA resists replay attacks in steps 1 and 6.

5) DoS Attack: The highest computation cost of eNB is during the users service request, in which, eNB is responsible to authenticate both sender and receiver (UE_i and UE_j) and run KDF to derive their keys. The presence of a random value $RAND$ in EPS-AKA in step 1, prevents attackers from resending service requests. In addition, the HMAC value attached to each message, prevents attackers from generating and sending a bunch of service request to eNB in a short time. Also, all the messages delivered to the user are attached with an HMAC that used a key derived from users secret permanent key K. So the adversary(s) cannot run a DoS attack nor make a verifiable HMAC value for even one message. However, HSS which is the bottleneck of the EPS-AKA protocol is still involved in the proposed protocol. To address this issue we will consider a group-based model in which only the cluster head connects to eNB. This is beyond the scope of this paper and will be covered in the future works of the authors.

6) Impersonation Attack: An adversary will not be authenticated by the eNB, unless it is able to generate verifiable AV parameters ($AUTN, RES,$) as a legal user can. With respect to the security of EPS-AKA, attacker(s) cannot carry out an impersonation attack. Additionally, the sniffed packets that are retransmitted will be dropped by eNB given the presence of random number.

7) Free ridding Attack: Similar to the approach proposed by Zhang et al.[15] for detecting selfish user behavior, our protocol punishes free ridders, by maintaining records which represent how many times they participated in a data sharing process. It is worth to note that the only way to increase a users share frequency amount is to transmit verifiable data, in which the signatures of the SP (σ_{sp}) and the HMAC value with the key $k_{cpi,j}$ transmitted in step 7, are verified.

VI. CONCLUSION

Here, we proposed a secure and lightweight data sharing protocol for D2D communication. To get rid of the user registration phase, we modify the EPS-AKA algorithm, to meet the demand of authentication and also to generate integrity and confidentiality keys that will be used during the data sharing process. In addition, by keeping a history of the users actions, we can revoke malicious users who are trying to share fabricated data with others. In a nutshell, the proposed protocol guarantees data confidentiality and integrity and resists message fabrication, man-in-the-middle, replay and DoS attacks with an acceptable performance by decreasing the computation cost of users compared with previous works.

REFERENCES

[1] Cisco,*Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update* , 2015-2016, Cisco, USA, Feb 2016.
[2] Adnan et al., *A survey on mobile data offloading: technical and business perspectives*,IEEE Wireless Communications, Vol 20, pp 104-112,2013.
[3] Sergey et al., *Cellular traffic offloading onto network-assisted device-to-device connections*, IEEE Communications Magazine,Vol 52, pp 20-31,2014.
[4] Asadi et al., *A survey on device-to-device communication in cellular networks*, IEEE Communications Surveys and Tutorials, Vol 16,1801-1819,2014.
[5] Wang, Mingjun, and Zheng Yan, *A survey on security in D2D communications*, Mobile Networks and Applications,1-14,2016.
[6] Naslcheraghi et al., *FD device-to-device communication for wireless video distribution*, IET Communications, Vol 11, pp 1074-1081, 2017.
[7] 3GPP, TR 33.401, v.14.2.0, *Security Architecture*, Release 14, 2017
[8] 3GPP, TS 33.105 version 14.0.0, *Cryptographic Algorithm Requirements*, Release 14, 2017
[9] Ekram et al., *Evolution toward 5G multi-tier cellular wireless networks: An interference management perspective*, IEEE Wireless Communications, Vol 21,118-127,2014.
[10] Choi, Kae Won, and Zhu Han, *Device-to-device discovery for proximity-based service in LTE-advanced system*, IEEE Journal on Selected Areas in Communications, Vol 33, pp 55-66, 2015
[11] Wang, Mingjun, and Zheng Yan. *A survey on security in D2D communications*, Mobile Networks and Applications,Vol 22, pp 195-208, 2017.
[12] Tehrani et al., *Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions*, IEEE Communications Magazine, Vol 52,pp 86-92, 2014.
[13] Zhang et al., *Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems*, IEEE Transactions on Information Forensics and Security, Vol 12,pp 662-675,2017.
[14] Schmittner et al., *SEMUD: Secure Multi-hop Device-to-Device Communication for 5G Public Safety Networks*
[15] Zhang, Aiqing, et al., *SeDS: Secure data sharing strategy for D2D communication in LTE-Advanced networks*, IEEE Transactions on Vehicular Technology,Vol 65, pp 2659-2672, 2016.
[16] Wang, Mingjun,et al., *UAKA-D2D: Universal Authentication and Key Agreement Protocol in D2D Communications*, Mobile Networks and Applications, pp 1-16, 2017
[17] Han, Chan-Kyu, and Hyoung-Kee Choi., *Security analysis of handover key management in 4G LTE/SAE networks*, IEEE Transactions on Mobile Computing, Vol 13,2014
[18] Chatterjee et al., *An Enhanced Access Control Scheme in Wireless Sensor Networks*, Adhoc and Sensor Wireless Networks, Vol 21, 2014.
[19] Alam, Muhammad, et al., *Secure device-to-device communication in LTE-A*, IEEE Communications Magazine, Vol 52, pp 66-73, 2014. 457-468.