American Journal of Electrical and Computer Engineering
2017; 1(1): 40-49
http://www.sciencepublishinggroup.com/j/ajece
doi: 10.11648/j.ajece.20170101.16

Secure Device to Device Communications for Next-Generation Networks Using Software-Defined Network

Firas Masoud, Mohammad Alchaita*, Mohammad Assora

Computer Science Department, Syria Higher Institutes for Applied Sciences and Technology, Damascus, Syria

Email address:
masoudferas@gmail.com (F. Masoud), malchaita@yahoo.com (M. Alchaita), m.assora@yahoo.co.uk (M. Assora)

*Corresponding author

To cite this article:
Firas Masoud, Mohammad Alchaita, Mohammad Assora. Secure Device to Device Communications for Next-Generation Networks Using Software-Defined Network. *American Journal of Electrical and Computer Engineering.* Vol. 1, No. 1, 2017, pp. 40-49.
doi: 10.11648/j.ajece.20170101.16

Received: March 11, 2017; **Accepted**: April 14, 2017; **Published**: May 26, 2017

Abstract: Mobile network security needs more attention to meet new emerging situations and applications that use modern technologies such as Device to Device (D2D) communications. One of these situations is getting the connection back securely between out of coverage (OoC) stations and the core network. This paper proposes a framework to reestablish this connection by using some of in-coverage stations, which are located at the edge of the injured area. The framework uses Software-defined Network (SDN) architecture. The local controllers (LCs), within SDN, plan the communications by selecting cluster heads (CHs) inside the injured area to begin D2D communications between these stations and the CHs, under the authority of the core network. In our framework, an effect of Free Riding Attack (FRA) can be mitigated. In addition, the privacy of user entity (UE) is achieved by decoupling the transmitted ID and the ID which is used in calculations. Furthermore, we accomplish secure connections between OoC UEs and the core network, with many security objectives such as data origin authentication, entity authentication and other security goals.

Keywords: D2D Communication, Free Riding Attack, MIKKY-SAKKE Algorithm, Radio Bearer

1. Introduction

The new directions of the next generation of mobile communications have many challenges, such as voluminous data, high data rate, large number of mobile stations in service, and unlimited demands with limited recourses. Another challenge is making mobile stations connected anywhere, anytime, and under any circumstances, with the new emerging applications that are used in both commercial and disaster situations [1]. These challenges drive us to reconsider the network architecture and the new emerging technologies. The Software-defined network (SDN) architecture can help to make the network functionalities programmable. By decoupling control and data planes, we can apply the network policies in elegant way. In addition, the network actions could be driven by events. On the other hand, the new technologies such as Device to Device D2D communications provide an effective infrastructure to enable new applications using direct communication between nearby user entities UEs. This technology has many desirable characteristics such as data offloading solution, enhancing resource utilization, increasing throughput, etc. But, the major problem that faces D2D communications is the lack of security objectives [5], which make user entities (UEs) vulnerable to many kinds of attacks, especially when eNodeB (eNB) becomes out of service. Security requirements such as data origin authentication, entity authentication, privacy, and other malicious behavior mitigation must be achieved to prevent the deviation from the objectives gained by these new technologies. Furthermore, the optimization of performance could not be achieved without mitigating the possible threats, and taking positive procedures toward malicious UEs during the normal and abnormal operations.

The core concept in our work is to demonstrate the powerful of the idea of pre-distributing key materials to the UEs. The key materials are distributed using pre-defined secure channel, which is guaranteed in LTE-A by using generic bootstrapping architecture (GBA) [3] between key management server (KMS) and UE; this will form the basic infrastructure to derive the functions and the transmissions to achieve such new security objectives. MIKEY-SAKKE algorithm [2] shares a

secret key between two entities participating proximity services and signs the secure message. This algorithm guarantees the confidentiality of sharing these keys, message authentication, message integrity, and sender non-repudiation. Further steps can be taken by integrating SDN architecture to accommodate the security objectives. This architecture is provided with hierarchical building and many capabilities to get more flexibility, and impose the best method to achieve the security objectives policy.

This paper is organized as follows: section II discusses the related works. Section III presents our proposed system model and discusses the results achieved. Section IV concludes the paper and proposes some future works.

2. Related Works

Many security aspects are addressed in [7] when two user entities reside within network coverage. Privacy is handled in conditional states. More security steps are needed to prevent advanced threats such as: the selected entity may fall as a victim to malicious behavior of requesting UE by sending a beacon report to eNB while receiving a true message.

The authors in [8] propose a secure key establishment using Diffie Hilman key exchange with commitment scheme. They have addressed the case when the active adversary makes independent connections with victims, making them believe that they are talking to each other directly.

The security management scheme for Out of Coverage UEs (OoC UEs) is proposed in [9] to enable authentication of UEs, confidentiality and integrity of the messages. The pre-distributed keys with indexes are proposed to establish D2D communications by intersecting the owned index of the UE with the received index. Trade off between connectivity and overhead as a function of network related parameter is addressed.

In [10, 11], physical layer based relay-assisted key generation approach is proposed. The main idea is to explore some relay nodes in the vicinity of two target nodes and use the random channels associated with these relay nodes as additional random sources for secret key generation between the two target nodes. In [10], the mobile nodes are partitioned into disjoint coalitions. Game theory is introduced to motivate the nodes within the same coalition at the vicinity to play a rule as relays, in which every node assists other nodes to establish a secure connection and gets help from the others in return. While in [11], the adversary is an eavesdropper, which assumed to be passive. The channel is proposed to change slowly; the cooperative scheme proposes to keep the generated key secure from both the adversary and relay. The multiplexing gain in the key rate grows linearly with the number of relays.

The authors in [12] propose a Secure Message Delivery Game [SMDG] protocol. The primary objective is to choose the most secure path to deliver a message from a sender to a destination. Energy consumption and QoS are considered by giving certain weights to the involved parameters (security, energy, QoS, etc.).

3. System Model

3.1. System Overview

Suppose we have a system as illustrated in figure 1. In this system, the global controller (GC) has an overall overview about the network activities, and provides programmable connectivity between core network entities at the northbound interface and user plane connectivity, with QoS needed at the southbound interface. GC applies the specified policy. The local controller (LC) drives many eNBs, and combines radio access network functionalities; thus, the access network is not restricted to LTE-A network. Therefore, the transmissions to achieve certain security objectives can be applied, such as in WiMAX, WiFi, and W-CDMA. In addition, LC provides a hierarchical architecture of eNBs which simplifies transitions between frequency bands and resource allocations. Suppose that one of the eNBs has dropped for some reason; the GC decides to use relays at the edge of the injured area to make secure connections between LC and OoC UEs, and use MIKEY-SAKKE protocol [2] to share the secret keys and sign it. The key materials that allow MIKKE-SAKKE protocol (ID_i, K_i, SSK_i, PVT_i) are previously provided to the UEs inside the injured area by the key management server (KMS) [4,6], which is a trusted party to all stations reside within the domain. Where IDi is a short, unambiguous identifier assigned to each user. Ki is the received secret key provisioned by KMS to share a secure message of a certain length with other parties using a SAKKE algorithm [6]. (SSK_i, PVT_i) pair is used to sign the secure message by the signer and validate the signed message by the verifier using Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption algorithm (ECCSI). Where SSK_i is the secret part of the algorithm and PVT_i is the public part [4]. On the other hand, the LC is assumed to participate within KMS domain with key materials (ID_{eNB}, K_{eNB}, SSK_{eNB}, PVT_{eNB}). The secure connection between LC and OoC UE is established over a relay using D2D communication between the relay and cluster head (CH). To reduce collisions due to announcements in the injured area, OoC UEs elect CHs and join to it.

CHs plan the resources temporarily between the stations, and the CH is the only part that announces its presence inside the injured area as illustrated in figure 2. UE_1 is called UE-to-Network relay as the name suggested by [3].

3.2. Cluster Head Election and out of Coverage Group Formation

When a node that is out of coverage needs to announce itself, it will be vulnerable to collisions with other nodes due to collisions. On the other hand, if a node is using previously dedicated band, it will not guarantee that someone on the same band listens to it. To overcome this problem and reduce the bands dedicated to such announcements, we need to control the percentage of stations that are announcing themselves and serving the other parties in the injured area by electing CHs. If a CH does not serve the other stations; the stations may elect another CH. In our proposed framework, the LEACH protocol

[13], which is used in Wireless Sensor Networks (WSN), is used to elect CHs.

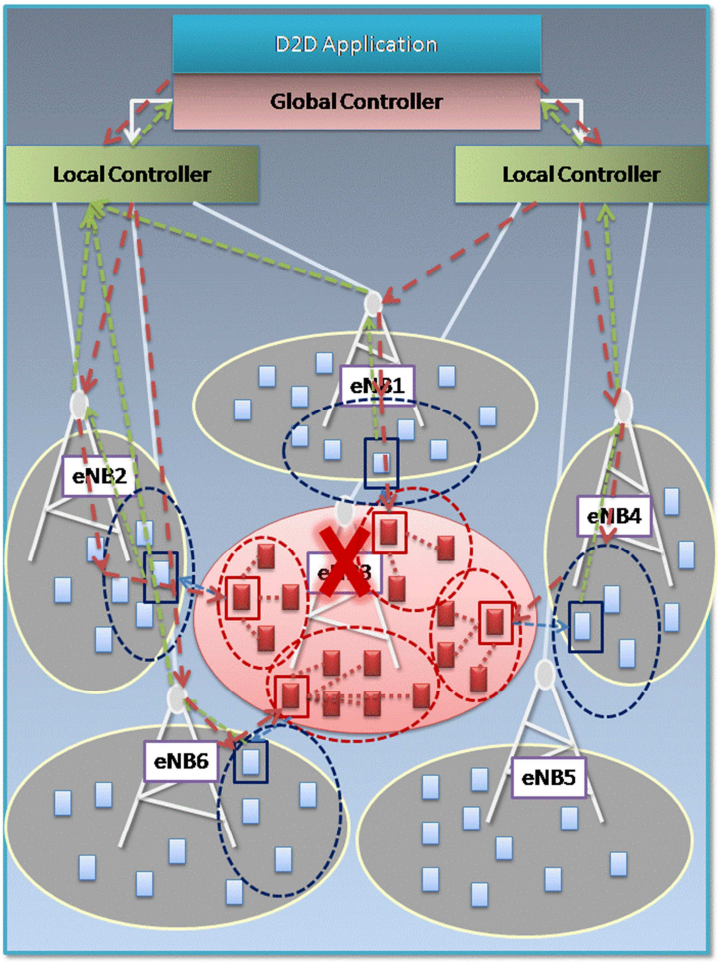

Figure 1. *System Overview.*

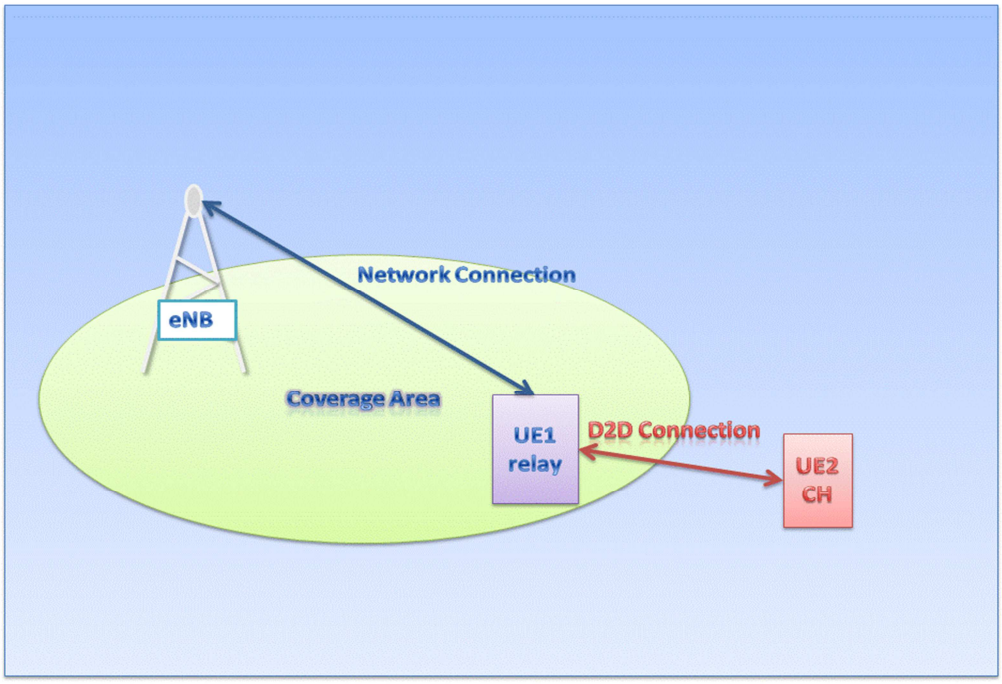

Figure 2. *D2D Communications: UE-to-network relay.*

Choosing CHs is done by using the following steps:

Each node chooses a random number between 0 and 1.

The node is chosen as CH if the chosen number is less than the threshold:

$$T(n) = \begin{cases} \dfrac{p}{1 - p\left(r \bmod \dfrac{1}{p}\right)} if n \in G \\ 0 \; if n \notin G \end{cases}$$

Where:

p: is the desired percentage of CHs.

r: is the current round.

G: The set of the nodes that have not been CH for the last 1/p round.

By this formula, we can control the desired percentage of CHs. The other nodes join the nearest CH and regulate the resources inside injured area temporarily, by using the LEACH protocol access method. If one of the CHs does not serve the stations in its area, they can do another round to choose another CH.

3.3. Relays

In the following discussion, we want to rejoin the CH securely to the mobile network over relay. The bottleneck in the process is the relay, since it has no motivations to participate in the process due to frequency resource sharing, hardware storage, and computation loads. Moreover, it may act malicious behaviors such as:

1) It may ignore the CH transmissions.
2) It may impersonate other stations like eNB, if it has enough information.
3) It may pretend that it cannot access the CH, and make it out of reach permanently.
4) It may pretend that CH has performed wrong transmissions, so the network will apply a certain policy toward CH.

Besides that, the relay that helps CH may impose radio resource and hardware sharing, which has negative effects on the relay performance. Three steps can be taken to help relays:

a) To avoid the frequency sharing, eliminate the performance degradation of the relay, and allow it to use its own band. The dropped eNB radio resources may be reused again, or dedicated D2D radio resources are allocated to establish a D2D radio bearer between LC and relay. By this way, the D2D radio bearer is used to provide OoC UE connectivity over relay.

b) The required computation processes and data storage volume of D2D communications must be reduced as possible as we can.

c) Distribution of the dedicated D2D communication loads to the minimum limits is achieved by invoking more than one relay that is located nearby CH as illustrated in figure 3 (this involves another task as we will see later).

Table 1. *The statistical table.*

Relay	Public Keys	KMS	True Transmissions (T)	Fault Transmissions (F)	Totla Transmissions (N = T + F)	Percentage F / N	S-TMSI	C-RNTI	Category
Relay 1	PID1, PVT1	KMS1, KPAK1, Z1							
Relay 2	PID2, PVT2	KMS2, KPAK2, Z2							
Relay n	PIDn, PVTn	KMSn, KPAKn, Zn							

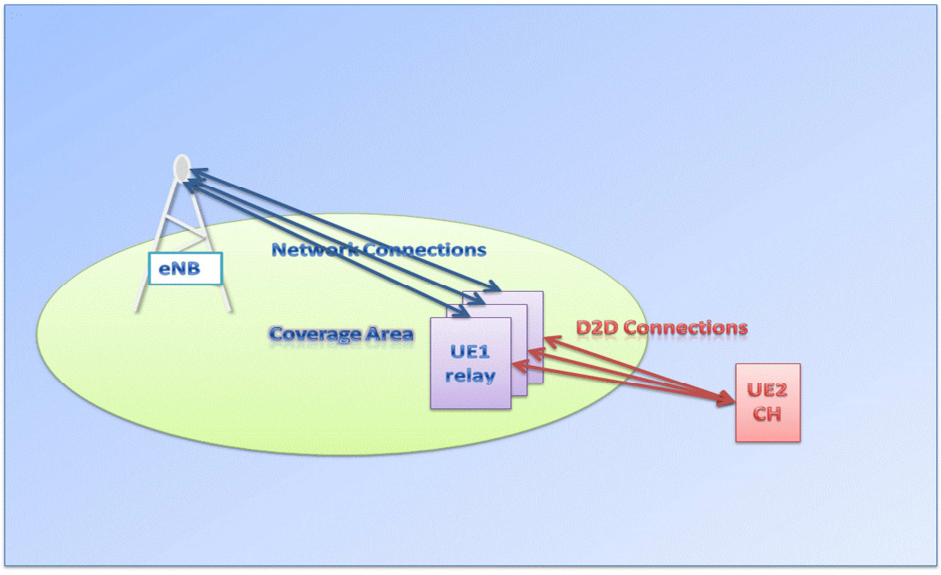

Figure 3. *The possibility of using more than one of in coverage area as relay.*

By using these steps, we can make D2D communications between the relay and the CH as transparent as possible from the relay viewpoint.

Choosing possible relays can be done by using evolved serving mobile location center (E-SMLC) that is located at the core network. CHs search process by relays is not effective, since it involves many transmissions, which may cause redundant transmissions. At the same time, the CH announcement may be vulnerable to ignorance by relays. We will call the relay as UE_1, and CH as UE_2 for simplicity, and we will differentiate between them as needed. LC can be replaced by eNB to make sense that the radio connection is between eNB and UEs. At the beginning, we suppose some principles that allow secure connection between eNB and UE_2 to achieve some security objectives:

1) UE_1 must not ignore UE_2 messages; the massages must not be modified or forged by UE1.
2) UE_1 cannot pretend UE_2 error transmissions without evidence.
3) Since UE_2 is isolated from the outside world, so any connection with it must be certified by KMS to ensure source authenticity.
4) When UE_1 malicious behavior oversteps a certain level, the network has the right to apply a certain policy to prevent such behaviors.

First of all, we use the statistical database figured in table 1 similar to that used in [7], for tracing the behavior of relays at the edge of the injured area. We send a token message to the relays. The token message is the obligation by the network on the relay; this obligation will make the relay selects the CH and never ignore its message, or modify or forge the messages. A token is sent to more than one relay. The basic condition is that relays don't know each other, and thus we can assume that relays cannot collude with each other to make CH out of service.

The local controller is responsible for making decisions about true/false transmissions; the decision making rule is that:

1) If some of relays admit that the transmission by the CH is false, but at least one of the relays admits that the transmission is true with evidence, then the decision is that all the relays that admit false transmission will get false transmission in their records in statistical table, and the relays with true transmission will get true transmission.
2) If the network detects that one of the relays has forged the message by CH, then it will get a false transmission in its record.
3) If all relays admit false transmission by CH, then the CH is a malicious station.
4) If the relays admit that no reply from a CH, then the CH is considered inaccessible.
5) If the wrong transmission rate of relays oversteps a certain level, the network will apply a certain policy toward it.

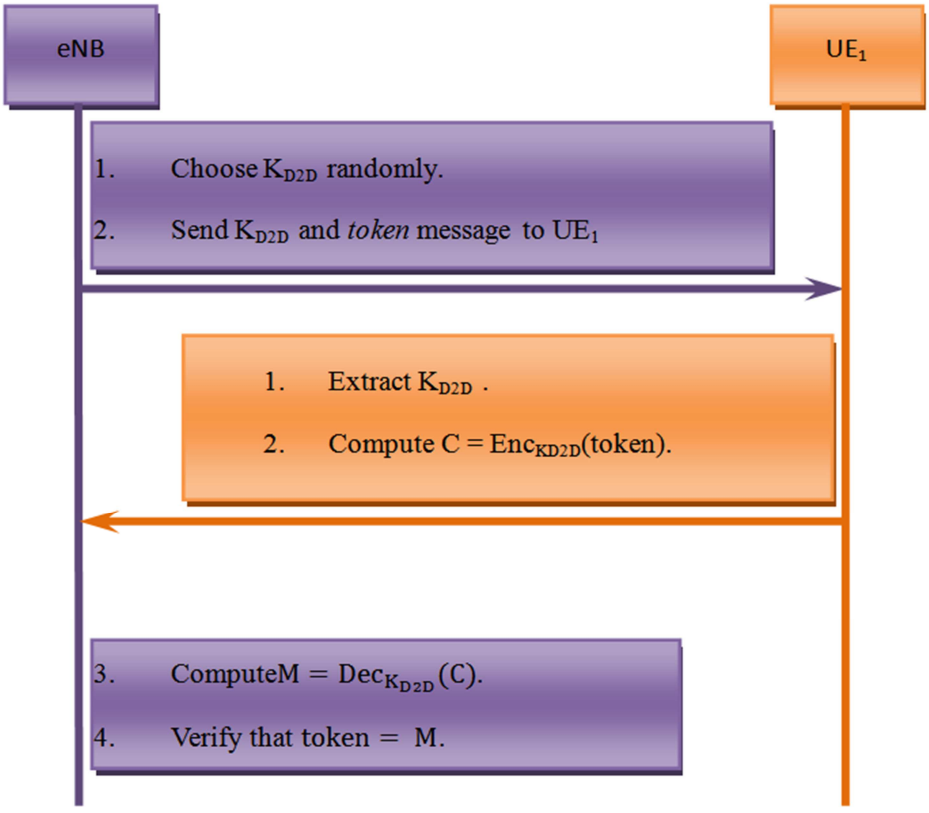

Figure 4. *Transmission between eNB and Relay to establish Secure D2D Radio Bearer.*

Table 1 is a convenient tool as it can be used to evaluate the performance of each node, which is the authentication degree of each relay. Another benefit is to distribute the load equally between relays. More complicated decisions with invoking the game theory could be used to increase the performance of the relays, especially for nodes with low authentication degree levels; and make the relay exploits its best effort to find the CH or any UE inside the injured area.

3.4. D2D Radio Bearer Between eNB and UE_1

In this section, we demonstrate how to establish a secure D2D radio bearer between eNB and UE_1 (relay), which is different from Radio Resource Control (RRC) bearer. Non-Access-Stratum (NAS) protocol [14] is still used in here, since we need the same objectives to achieve D2D radio, and the key lengths are the same. Mutual authentication is achieved previously when UE has attached the network. In D2D radio bearer establishment, we need to deliver the secret key K_{D2D} securely. Hence, send the token message to UE_1, and confirm that K_{D2D} and the token are delivered correctly. The procedure is illustrated by figure 4, and explained in the following:

1) Find the secret key between eNB and UE_1 which called K_{D2D}.
2) A token message must be delivered correctly to the relay.
3) UE_1 encrypts the token message with K_{D2D} and returns the result to the eNB.
4) The eNB verifies that the message and the key are delivered correctly.
5) Mutual authentication is not needed, since it is previously preserved. The establishment is done by using RRC radio bearer.

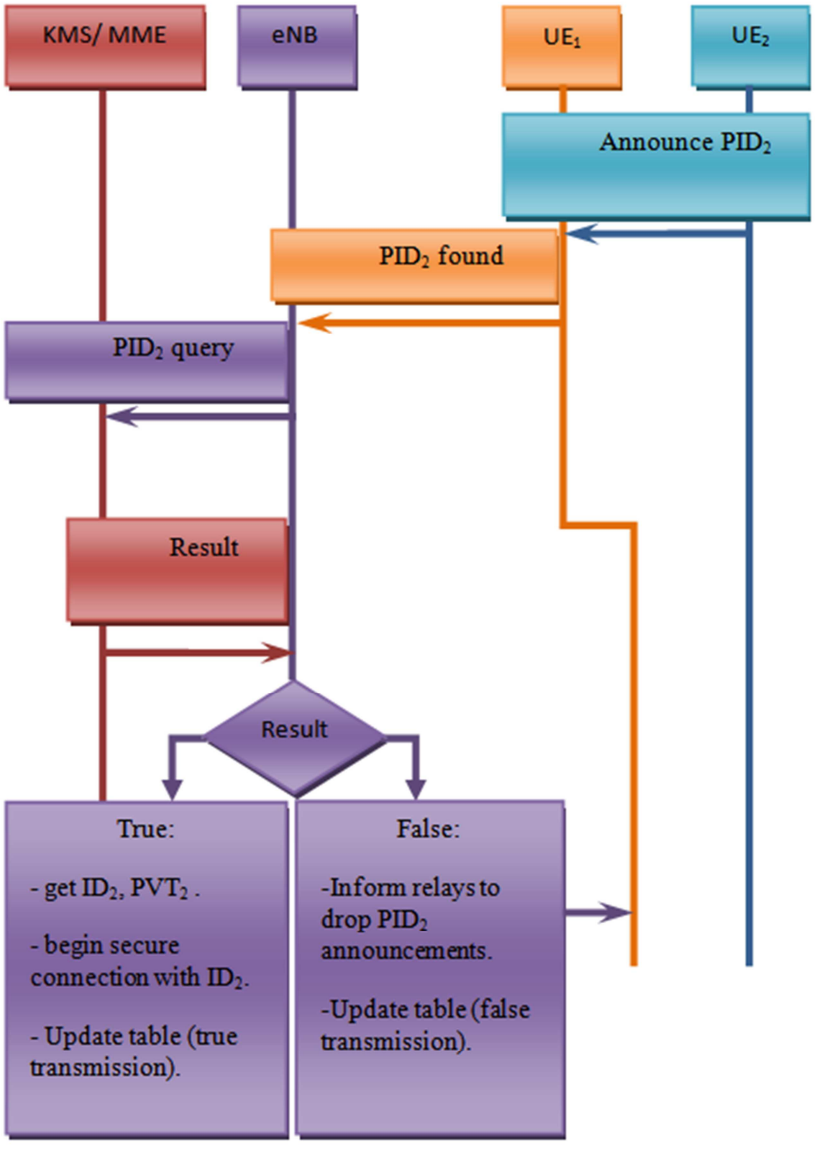

Figure 5. *CH discovering stage and the processes taken by eNB.*

3.5. Searching Stage

As UE_1 receives the token message, it will begin to search the UE_2 announcement. The UE_2 is prevented from announcing its ID directly, but it uses another format such as hashing its ID; PID_2 = hash (ID_2). When UE_1 captures the announcement message that contains PID_2, it will redirect the message to the eNB, which enquiries KMS to get more information about CH of UE_2. If this step is not completed properly, the eNB will inform relays to drop any announcement that contains PID_2. But, if it is completed properly, then the statistical table is updated with true transmissions for relays that found PID_2; as illustrated in figure 5.

3.6. Joining Cluster Head to Mobile Network

The CH is vulnerable to many kinds of threats, since it does not have any way to identify the stations that are trying to connect with it. Furthermore, it does not guarantee that these stations are ready to provide the requirements. The CH needs to communicate securely with trusted stations that are authorized by the mobile network as relays; and communicate securely with the nearest eNB over relays. To join the CH securely, we need:

1) Secure connections with relays.
2) A secure connection with the nearest eNB over relays.

We introduce the concept of certificate, which allows the CH of UE_2 to validate that the eNB or the relay of UE_1 is authorized by the KMS (the trusted party to all stations within a domain) to perform this connection. On the other hand, the eNB, by getting sub-authority from the KMS to perform entity authentication, can identify the CH. Another security requirement at this stage is to get the confirm messages from UE_1 with evidences, when a correct message is sent by a CH. An additional security requirement, the eNB needs to prevent the attack, when the CH sends a true confirm message to a relay and a wrong evidence message to the eNB. This may be considered as a malicious behavior by the relay, while the CH is the malicious station. The eNB is responsible to prevent malicious behaviors of relays, by providing them with a certificate from the KMS, and to apply network policy when a malicious behavior is detected. If we achieve this task, then we get a secure connection between the eNB and the CH over a set of unsafe stations from the CH viewpoint.

Finally, the privacy requires that IDs must be preserved, so, the announcements must not involve sending IDs in clear. Two different functions of a real ID are used; one for transmission and the other for computational processes. Where, a random number from the KMS is sent each time D2D communications is needed.

Suppose that, the eNB wants to send a shared secret key (M) to the CH, to secure the messages between the eNB and the CH (the key is derived as in NAS protocol). MIKEY-SAKKE algorithm [2] is invoked to share the secret message and sign it. The message of the form: mikey-sakke(eNB, M) means that the eNB sends a message M to the CH with a secure message:

$$\text{sec}_{msg} = R_{(eNB,S)} \parallel H_{eNB} \quad [10]$$

and a signature:

$$signature = h_{eNB} \parallel s_{eNB} \parallel PVT_{eNB} \quad [4]$$

The procedure to form the certificate and to achieve security requirements is as the following and is illustrated in figure 6.

eNB:

1) Compute:

$$S_{token,UE_1} = \left[token.SSK_{UE_1}\right].P.$$

2) Compute:

$$S_{M,eNB} = [M.SSK_{eNB}].P$$

3) Send:

$$S_{token,UE_1}, S_{M,eNB} \text{ to KMS.}$$

KMS:

1) Choose a; Compute:

$$A = [a.ID_2].P$$

To achieve the privacy, a is the coefficient that protects the ID of UE_2. The variable a takes place inside the mikey-sakke algorithm as we will see.

2) Compute:

$$C_{token,UE_1} = < S_{token,UE_1}, K_2 > = g^{token.SSK_{UE_1}(ID_2+z)^{-1}}$$

$$where\ K_2 = [(ID_2 + z)^{-1}].P$$

3) Compute:

$$c_{token,UE_1} = hash\left(C_{token,UE_1} \parallel PVT_{UE_1} \parallel a\right)$$

4) Compute:

$$C_{M,eNB} = < S_{M,eNB}, K_2 > = g^{M.SSK_{eNB}(ID_2+z)^{-1}}$$

$$where\ K_2 = [(ID_2 + z)^{-1}].P$$

5) Compute:

$$c_{M,eNB} = hash\left(C_{M,eNB} \parallel PVT_{eNB} \parallel a\right).$$

6) Compute:

$$L = hash\left(K_{x,2} \parallel ID_2\right);$$

$$where\ K_2 = \left(K_{x,2}, K_{y,2}\right)\ in\ affine\ coordinates$$

L means that the KMS authorizes the eNB to confirm an entity authentication of UE_2.

7) Send $c_{token\ ,UE_1}, C_{M,eNB}, a, A\ and\ L$ to eNB

eNB:

1) Compute:

$$L_1 = hash(token \parallel L)$$

2) Compute:

$$V_1 = hash(token \parallel L_1)$$

3) Compute:

$$L_2 = hash(M \parallel L).$$

4) Compute:

$$V_2 = hash(token \parallel L_2).$$

5) Send:

(Connection with UE_2 request $\parallel PID_{eNB} \parallel PID_2 \parallel KPAK_{eNB} \parallel mikey - sakke(M, eNB) \parallel c_{M,eNB} \parallel V_2 \parallel c_{token,UE_1} \parallel V_1, a, A$) to UE_1.

UE_1: UE_1 Challenges the CH and shares token with it.

Send.

(Challenge CH request $\parallel PID_{UE_1} \parallel PID_2 \parallel a \parallel KPAK_{UE_1} \parallel mikey - sakke(token, UE_1) \parallel c_{token,UE_1}$) to CH.

UE_2:

1) Extract the message *token* and confirm its correctness.
2) Compute:

$$Y_{UE_1} = [HS_{UE_1}].PVT_{UE_1} + KPAK_{UE_1} = [SSK_{UE_1}].P$$

3) Confirm the signature.
4) To confirm the certificate :

a) Compute:

$$D_{token} = < [token].Y_{UE_1}, K_2 > = g^{token.SSK_{UE_1}(ID_2+z)^{-1}}$$

b) Compute:

$$\mathrm{d_{token}} = \mathrm{hash}(\mathrm{D_{token}} \parallel \mathrm{PVT_{UE_1}})$$

c) Compare if the calculated $\mathrm{d_{token}}$ equals to the received $\mathrm{c_{token}}$.

d) Compute:

$$\mathrm{L} = \mathrm{hash}(\mathrm{K_{x,2}} \parallel \mathrm{ID_2})$$

e) Compute:

$$\mathrm{Confirm_1} = \mathrm{hash}(\mathrm{token} \parallel \mathrm{L})$$

f) Send

$$\mathrm{PID_2} \parallel \mathrm{Confirm_1} \text{ to UE1.}$$

UE_1:

1) Compute:

$$U_1 = hash(token \parallel Confirm_1).$$

2) Compare: $if\ U_1 == V_1$. If true, send true message to eNB with $Confirm_1$.

eNB:

a) Validate $Confirm_1$ message, UE_2 is considered accessible.
b) Add a true transmission to CH record in the DB.

UE_1:

Send the message from eNB.

(Connection with UE_2 request $\parallel PID_{eNB} \parallel PID_2 \parallel a \parallel KPAK_{eNB} \parallel mikey - sakke(M, eNB) \parallel c_{M,eNB}$) to UE_2

UE_2:

1) Extract the message *M* and confirm its correctness.
2) Compute:

$$Y_{eNB} = [HS_{eNB}].PVT_{eNB} + KPAK_{eNB} = [SSK_{eNB}].P.$$

3) Confirm the signature.
4) To confirm the certificate :

a) Compute:

$$\mathrm{D_M} = < [\mathrm{M}].\mathrm{Y_{eNB}}, \mathrm{K_2} > = \mathrm{g}^{\mathrm{M.SSK_{eNB}(ID_2+z)^{-1}}}$$

b) Compute:

$$\mathrm{d_M} = \mathrm{hash}(\mathrm{D_M} \parallel \mathrm{PVT_{eNB}}).$$

c) Compare if the calculated $\mathrm{d_M}$ equals to the received $\mathrm{c_M}$.

d) Compute:

$$\mathrm{Confirm_2} = \mathrm{hash}(\mathrm{M} \parallel \mathrm{L}).$$

e) Send

$$\mathrm{PID_2} \parallel \mathrm{Confirm_2} \text{ to } \mathrm{UE_1}.$$

UE_1:

1) Compute:

$$U_2 = hash(token \parallel Confirm_2).$$

2) Compare: $if\ U_2 == V_2$. If true, send true message to eNB with $Confirm_1$.

eNB:

Validate $Confirm_2$ message, UE_i is considered Connected Securely.

The modifications that are suggested in SAKKE algorithm [6] to conceal ID usage is as follows:

Sender:

1) Select a random ephemeral integer value (SSV) in the range 0 to $2^n - 1$;
2) Compute h_1 = HashToIntegerRange(SSV $\parallel \mathcal{A}$, r, Hash); where $\mathcal{A} = [a.ID_i].P$, and Hash is a hash function.
3) Compute R_($\mathcal{A}$,S) = [h_1]([$\mathcal{A}$] + a.Z) in E(F_p); where, F_p denotes the finite field of p elements, where p is a prime. E is an elliptic curve defined over F_p.
4) Compute the Hint, H;

a) Compute $g^{a.h_1}$. Note that g is an element of PF_p[q] represented by an element of F_p.

b) Compute:

H := SSV XOR HashToIntegerRange ($g^{a.h_1}$, 2^n, Hash);

1) Form the Encapsulated Data (R_($\mathcal{A}$,S), H), and transmit it to B;

2) Output SSV for use to derive key material for the application to be keyed.

Receiver:

1) Parse the Encapsulated Data (R_($\mathcal{A}$,S), H), and extract R_($\mathcal{A}$,S) and H;
2) Compute w: = < R_($\mathcal{A}$,S), K_($\mathcal{A}$,S) >. Note that by bilinearity, w: = $g^{a.h_1}$;
3) Compute SSV = H XOR HashToIntegerRange(w, 2^n, Hash);
4) Compute h_2 = HashToIntegerRange(SSV || b, q, Hash);
5) Compute TEST = $[h_2]([\mathcal{A}]P + Z\_S)$ in E(F_p). If TEST does not equal R_($\mathcal{A}$,S), then B MUST NOT use the SSV to derive key material;
6) Output SSV for use to derive key material for the application to be keyed.

Sending a message from the eNB to the CH of UE2 means that the KMS authorizes the eNB to send the message M, and UE2 can validate that, so the message is considered secure. As a result, it is connected securely with mobile network via the relay of UE1.

We notice that the procedure involves exchanging two secret messages with UE2. The first one for exchanging a token message, which confirms that UE2 is accessible and the eNB guarantees that relays work properly. The importance of exchanging the token message is that when the relay is moving away from the injured area, while another station is moving close to this area, the eNB immediately establishes D2D bearer with the new station, by exchanging token message with it, and sending a message to UE2. This will make the access to UE2 as transparent as possible. Even if the token message is vulnerable to disclosure, this is worthless without authorization from the eNB and the KMS. The second message is exchanging the M message between the eNB and UE2. This message involves the basic key materials to extract further keys just like NAS protocol; it is designed to be of the same length (128 bits).

3.7. Joining OoC UEs to Mobile Network

After the CH joins the mobile network over relays; we need to rejoin the other stations inside the injured area. Since the CH has the same token message, the same imposition and the same procedure are applied by the network. UE_1 is the station that possesses the token message. We want to deliver the token message to UE_2. When a station is rejoined the mobile network, it will participate in the joining process to distribute the processing load. The joining message to an OoC UE, which is originated from the eNB, can be delivered either by the CH or another recently joined station.

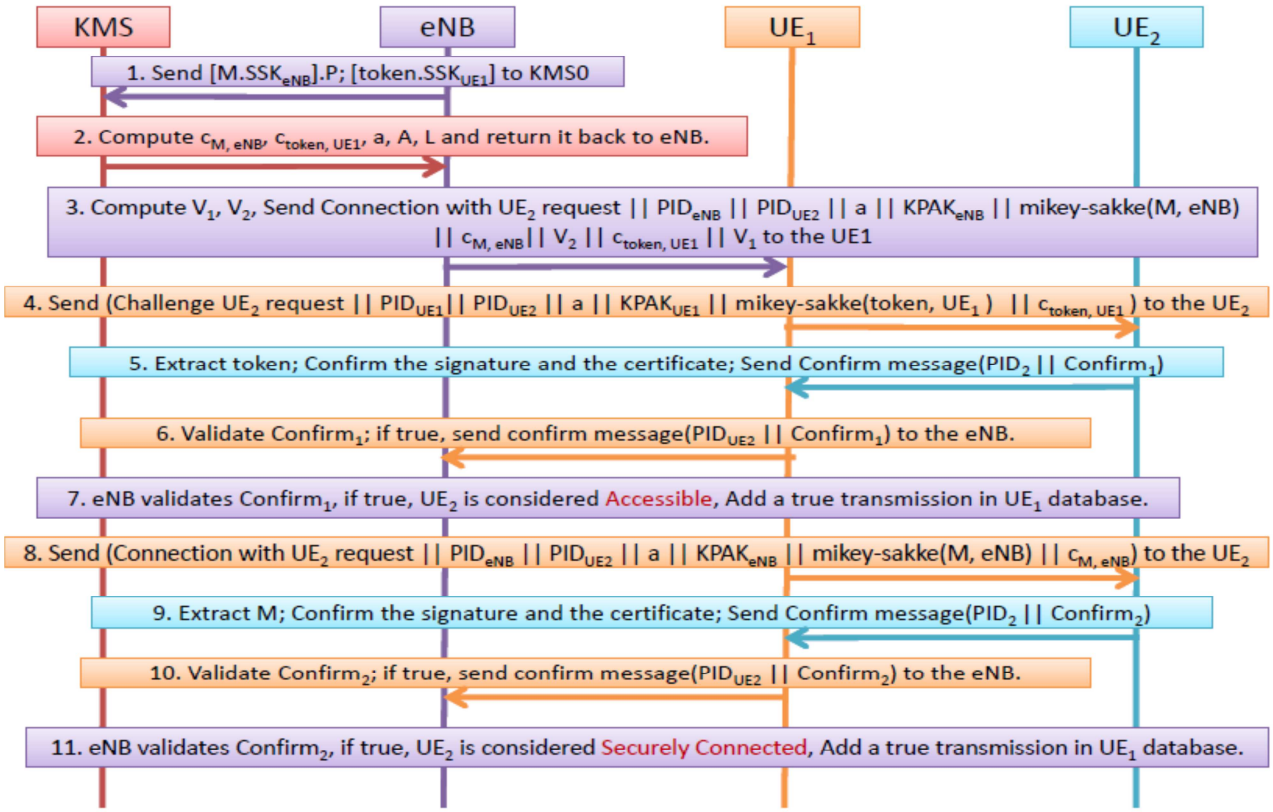

Figure 6. *Transmissions between eNB, KMS, UE1 and UE2.*

4. Conclusion and Future Works

We have introduced the D2D radio bearer to reduce radio sharing and performance degradation with relays. This framework employs more than one relay to distribute the hardware load, storage capacity, and heavy processing.

Moreover, it conceals the relays' IDs and does not depend on one party. The statistical table is introduced for tracing relays behaviors. The decision that is taken by the LC is important to improve relays reliability. The game theory can be used in relays to build the decision rules that are based on relays reliabilities. By using certificates with MIKEY-SAKKE protocol to help CH communicating with trusted stations, we can share keys securely, provide message origin authentication, message integrity, and source non-repudiation. Moreover, the framework provides entity authentication for OoC UEs, in addition to receiver non-repudiation. The sub-authority technique allows the KMS, which has a full authorization capability, to authorize the eNB or a trusted station to perform a number of security services such as entity authentication. Further steps can be taken to authorize the eNB to certificate its message without communicating with the KMS. The framework conceals the IDs of participating stations to maintain the stations privacy, since we do not have any idea about the attackers' capabilities. Therefore, this will mitigate unpredictable malicious behaviors of some parties. As we have seen; the functions and parameters that are used for sending and computing are different, which means that the only entity that can send and compute the transmissions properly is the station that has the real identity. Through our work; we have introduced a realistic scenario that may 5G mobile network encounter. This scenario can be generalized with the same technique, and the same security objectives to be a framework for D2D communications, when user entities are out of coverage.

References

[1] Beyond LTE: Enabling the Mobile Broadband Explosion, Rysavy Research/4G Americas, August 2015.

[2] RFC 6509; MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY).

[3] 3GPP TR 33.833 V1.7.0 (2016-02); 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on Security issues to support Proximity Services (ProSe) (Release 13).

[4] RFC 6507; Elliptic Curve-Based Certificateless Signaturesfor Identity-Based Encryption (ECCSI).

[5] M. Wang and Z. Yan, "Security in D2D Communications: A Review", IEEE computer society, 2015.

[6] RFC 6508; Sakai-Kasahara Key Encryption (SAKKE).

[7] SeDS: Secure Data Sharing Strategy for D2D Communication in LTE-Advanced Networks; Aiqing Zhang, Student Member, IEEE, Jianxin Chen, Member, IEEE, Rose Qingyang Hu, Senior Member, IEEE, and Yi Qian, Senior Member, IEEE.

[8] Secure Key Establishment for Device-to-Device Communications Wenlong Shen, Weisheng Hong, Xianghui Cao, Bo Yin, Devu Manikantan Shilaand Yu Cheng; 2014.

[9] Connectivity and Security in a D2D Communication Protocol for Public Safety Applications; Leonardo Goratti, Gary Steri, Karina M. Gomez and Gianmarco Baldini; CREATE-NET Research Centre, Trento, Italy; 2014.

[10] SYNERGY; A Game-Theoretical Approach for Cooperative Key Generation in Wireless Networks Jingchao Sun, Xu Chen, Jinxue Zhang, Yanchao Zhang, and Junshan; 2014.

[11] KEEP: Fast Secret Key Extraction Protocol for D2D Communication Wei Xi, Xiang-Yang Li, Chen Qian, Jinsong Han, Shaojie Tang, Jizhong Zhao, Kun Zhao; 2014.

[12] Secure Message Delivery Games for Device-to-Device Communications; Emmanouil Panaousis, Tansu Alpcan, Hossein Fereidooni, and Mauro Conti; 2014.

[13] Descendant of LEACH Based Routing Protocols in Wireless Sensor Networks; Rajendra Prasad Mahapatra, Rakesh Kumar Yadav; 2015.

[14] 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3 (Release 8); 2009.