International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 5 Issue 6 ǁ June. 2017 ǁ PP. 46-55

Provably Secure Authenticated Key Management Protocol Against De-Synchronization Attacks in the Intra-MME Handover

[1]Khaled Khairy,[2]Mainak Chatterjee
Department of Computer Science, University of Central Florida, USA

ABSTRACT: This paper presents an authenticated key management protocol for Intra-MME handover over LTE networks. The proposed protocol is formalized using Multi-Set Rewriting approach with existential quantification. The rules specifying the Dolev-Yao intruder model for the proposed protocol is presented, and the immunity of the proposed protocol against the de-synchronization attack, which is the most dangerous attack against the standard protocol, is proved formally.

Keywords: LTE, Handover, Formalization, Multi-Set Rewriting (MSR).

I. INTRODUCTION

Intra-MME handover in long-term evolution (LTE) networks takes place between the source evolved node (eNB) and target eNB under the same mobile management entity (MME). The terms handover refers to the process of transferring an ongoing call or data session from one channel connected to the core network to another channel. In the standard intra-MME handover key management, the session keys for ciphering and integrity protection are derived from the subscriber specific root key (K_{ASME}), permanently stored in the universal subscriber identity module (USIM) and also in the core network authentication center (AUC). The MME derives three keys from K_{ASME}: two transient keys K_{NASenc} and K_{NASint}, and a third key K_{eNB} specific for encrypting the traffic between the nodes and the users [1]. The K_{eNB} is transformed to a new key K_{eNB}* by a one way key derivation function (KDF) using fresh parameters, i.e., Next Hop (NH) key and the NH Chaining Counter (NCC) to ensure one-hop forward and backward key separation [2].

Before the local root key K_{ASME} is updated, an intruder using a rogue base station [6] may apply the de-synchronization attack to break the forward key separation. Therefore, the new session keys are compromised. Thus, two-hop forward key separation was introduced [3]. This loophole in the standard handover key management protocol has been presented in some recent works [4-6] which suggested some solutions to overcome the de-synchronization attack [7-10], but those could not completly prevent this attack. Therefore, keeping these keys away from the intruders, during the handover process, by enhancing the current intra-MME handover key management protocol is still needed to prevent the de-synchronization attack and to maintain the one-hop forward security. The security protocol analysis aims to prove that a protocol is correct and tries to find any attack scenarios that may result in the failure of any secrecy properties. The Dolev–Yao abstraction [11] considers the protocols as a form of multi-set rewriting (MSR) with existential quantification [12]. In MSR approach, the protocol execution could be carried out symbolically and the interaction of a well-founded protocol theory with an intruder theory can be considerd by analyzing the traces of the protocol. MSR has so far been applied to many protocols such as Needham-Schroeder and Neumann-Stubblebine [15], Kerberos authentication [16, 17] and Diffie-Hellman protocols [18].

This paper proposes an authenticated key management protocol, provably secure against the desynchronization attack during the intra-MME handover in LTE networks. The proposed protocol can overcome this attack by keeping out the source eNB from the handover process and using the MME as a third trusted party. The traces of the standard protocol showed that the intruder impersonating the source eNB can carry out the handover process and learn the new session key. On the other side, we prove in a formal manner that the proposed protocol can detect the intruder and aborts the handover process. The rest of this paper is structured as follows: section 2 gives an overview of the MSR formalization approach and the rules specifying the Dolev-Yao intruder model. The intra-MME handover key management protocol is presented in section 3. The proposed protocol is presented in section 4. Finally, the conclusions are drawn in section 5.

II. MULTI-SET REWRITING (MSR) WITH EXISTENTIAL QUANTIFICATION

The MSR is a simple logic-oriented language aimed at investigating the decidability of protocol analysis under a variety of assumptions. It yields elegant and precise formalizations, and supports a useful array of static check that include type-checking and access control validation. The protocol execution phase is divided into three stages: the initialization theory, the role generation theory, and the disjoint union of bounded sub-theories. In the initialization theory the vocabularies (first-order signature), the function symbols, and the predicate symbols with specific sorts are chosen. In the role generation theory, the state is modeled as a multi-

set of facts. The existential quantification, symbolically model the generation of fresh data (e.g., nonce or session keys). The memory predicates are used to encode systems consisting of a collection of coordinated sub-protocols. Also the constraints are used for testing objects belonging to complex interpretation domains, e.g., time stamps, in an abstract and modular way [13, 14].

In the third stage, the rules comprising sub-protocols or sub-theories are joined in a protocol role theory parameterized by the principal executing it. Rules in a role are threaded using role state predicates declared inside the role which record the information accessed by a rule. The role is given as the association between a role owner and a collection of rules. Some roles, such as those implementing a server or an intruder, are intrinsically bound to a few specific principals, often just one. We call them anchored roles and denote them as ρ^A. Other roles can be executed by any variable principal. In this case, principal A must be kept as a parameter bound to the role. These generic roles are denoted as $\rho^{\vdash A}$.

2.1 Signature

The signature fragment shown in Table 1 sets up the typing infrastructure used in this paper, with the 'Types' column summarizing the types used. The 'Sub-typing' column expresses the sub-typing relations satisfied with these types: (A<: B) means that A is a sub-sort of B, with indentation used as a visual aid to track dependencies. Table 2 shows the function symbols used in this paper. In any signature, each function symbol must have a fixed set of parameter sorts (one for each function argument) and a result sort. Finally, Table 3 shows the types of predicates that can enter a state or a rewrite rule used in this paper.

Table 1. MSR Sorts Used in this Paper

Sorts	Types	Subtyping	Name
Messages	msg: type.		m
Principals	principal: type. tcs: type ts: type. User: type. eNodeB: type. Server: type.	principal <: msg. tcs <: principal. ts <: tcs. User <: tcs. eNodeB<: ts. Server <: ts.	 UE eNB1, eNB2 MME
Encryption types	etype: type.	etype <: msg.	e
Keys	key: etype → type. dbK: etype → tcs → type. shK: etype → client→ ts → type.	∀e: etype, A: tcs. dbK^e A <: key^e. ∀e: etype, A: tcs, A: ts. shK^e C A <: key^e.	k_ dbK_ shK_ _
Nonce	nonce: type.	nonce <: msg.	n_
Timestamps	time: type.	time <: msg.	t_,_
cipher	cipher: type.	cipher <: msg	X, Y

Table 2. Function Symbols for the Proposed Protocol

Function	Modeling	Arguments
Encryption	enc (_, <_>)	e_key × msg → cipher
Hash	H (<_>)	msg → msg
Pairing	< _, _ >:	msg × msg → msg

Table 3. Predicate Symbols for the Proposed Protocol

Type	Modeling
Public Network Predicate	N(_,_,…,_)
State Predicates	L(_,_,…,_)
Private Memory Predicates	M (_,_,…,_).

2.2 Dolev-Yao intruder model

The rules specifying the Dolev-Yao intruder model [11] for the proposed protocol can be divided into three categories as follows:

2.2.1 Network, pairing and encryption rules

- The Dolev-Yao intruder can work with data on the network or in his possession; the rules in each pair (*e.g.*, encryption and decryption) are symmetric operations.
- The intruder may intercept network messages (INT), remove them from the network, transmit messages he knows (TRN), decompose (DMC) and compose (CMP) compound messages.
- The intruder may duplicate (DPM and DPD) and delete (DLM and DLD) messages or database keys he knows.
- The intruder may introduce a function for pairing, which is abbreviated as: < _, _ >: *msg* × *msg* → *msg*.
- If the intruder knows the shared key, he may decrypt (SDC') and encrypts (SEC') messages using this key as follows:

$$\left[\begin{array}{l}\forall\, UE\text{: tcs.}\\ \forall\, eNB1\text{: ts.}\\ \forall\, K_{eNB}\text{: shK } UE\ eNB.\\ \forall\, m\text{: msg.}\end{array}\quad \begin{array}{c}\mathrm{I}(\mathrm{enc}(K_{eNB}, <m>))\\ \mathrm{I}(K_{eNB})\end{array}\quad \begin{array}{c}\mathrm{SDC'}\\ \rightarrow\end{array}\quad \begin{array}{c}\mathrm{I}\\ \mathrm{I}(m)\end{array}\right]\left[\begin{array}{l}\forall\, UE\text{: tcs.}\\ \forall\, eNB1\text{: ts.}\\ \forall\, K_{eNB}\text{: shK } UE\ eNB.\\ \forall\, m\text{: msg.}\end{array}\quad \begin{array}{c}\mathrm{I}(m)\\ \mathrm{I}(K_{eNB})\end{array}\quad \begin{array}{c}\mathrm{SEC'}\\ \rightarrow\end{array}\quad \begin{array}{c}\mathrm{I}\\ \mathrm{I}(\mathrm{enc}(K_{eNB}, <m>))\end{array}\right]$$

- If the intruder knows a database key, he may decrypt (DDC') and encrypt (DEC') messages using this key as follows:

$$\left[\begin{array}{l}\forall\, UE\text{: tcs.}\\ \forall\, eNB1\text{: ts.}\\ \forall\, K_{eNB}\text{: shK } UE\ eNB.\\ \forall\, m\text{: msg.}\end{array}\quad \begin{array}{c}\mathrm{I}(\mathrm{enc}(K_{eNB}, <m>))\\ \mathrm{I}(K_{eNB})\end{array}\quad \begin{array}{c}\mathrm{SDC'}\\ \rightarrow\end{array}\quad \begin{array}{c}\mathrm{I}\\ \mathrm{I}(m)\end{array}\right]\left[\begin{array}{l}\forall\, UE\text{: tcs.}\\ \forall\, eNB1\text{: ts.}\\ \forall\, K_{eNB}\text{: shK } UE\ eNB.\\ \forall\, m\text{: msg.}\end{array}\quad \begin{array}{c}\mathrm{I}(m)\\ \mathrm{I}(K_{eNB})\end{array}\quad \begin{array}{c}\mathrm{SEC'}\\ \rightarrow\end{array}\quad \begin{array}{c}\mathrm{I}\\ \mathrm{I}(\mathrm{enc}(K_{eNB}, <m>))\end{array}\right]$$

- If the intruder knows a database key, he may decrypt (DDC') and encrypt (DEC') messages using this key as follows:

$$\left[\begin{array}{l}\forall\, eNB1\text{: ts.}\\ \forall\, K_{IP}\text{: dbK eNB.}\\ \forall\, m\text{: msg.}\end{array}\quad \begin{array}{c}\mathrm{I}(\mathrm{enc}(K_{IP}, <\mathrm{m}>))\\ \mathrm{I}(K_{IP})\end{array}\quad \begin{array}{c}\mathrm{DDC'}\\ \rightarrow\end{array}\quad \begin{array}{c}\mathrm{I}\\ \mathrm{I}(m)\end{array}\right]\left[\begin{array}{l}\forall\, eNB1\text{: ts.}\\ \forall\, K_{IP}\text{: dbK eNB.}\\ \forall\, m\text{: msg.}\end{array}\quad \begin{array}{c}\mathrm{I}(m)\\ \mathrm{I}(K_{IP})\end{array}\quad \begin{array}{c}\mathrm{DDC'}\\ \rightarrow\end{array}\quad \begin{array}{c}\mathrm{I}\\ \mathrm{I}(\mathrm{enc}(K_{IP}, <m>))\end{array}\right]$$

2.2.2. Data generation rules

- In general, the intruder should be able to generate everything an honest principal can generate, often fresh nonce (NG), session keys (KG'), and generic messages (MG) as follows:

$$\left[\begin{array}{c}\mathrm{NG}\\ \rightarrow\end{array}\quad \begin{array}{c}\exists\,\mathrm{n: nonce}\\ \mathrm{I}(n)\end{array}\right]^{\mathrm{I}}\left[\begin{array}{l}\forall\,\mathrm{UE: User.}\\ \forall\,\mathrm{eNB: ts}\end{array}\quad \begin{array}{c}\mathrm{KG'}\\ \rightarrow\end{array}\quad \begin{array}{c}\exists\,\mathrm{k: shK\ UE\ eNB}\\ \mathrm{I}(k)\end{array}\right]^{\mathrm{I}}\left[\begin{array}{c}\mathrm{MG}\\ \rightarrow\end{array}\quad \begin{array}{c}\exists\, m\text{: msg}\\ \mathrm{I}(m)\end{array}\right]^{\mathrm{I}}$$

- The intruder can only construct a message digest (MD). However as follows:

$$\left[\forall\, m\text{: msg.}\quad \begin{array}{c}\mathrm{I}(m)\\ \mathrm{I}(k)\end{array}\quad \begin{array}{c}\mathrm{MD}\\ \rightarrow\end{array}\quad \mathrm{I}(\mathrm{h}(m))\right]^{\mathrm{I}}$$

2.2.3. Data access rules

- the intruder has access to the principal's names (PA), or any defined timestamp as follows:

$$\left[\forall\,\mathrm{UE: User.}\quad .\quad \begin{array}{c}\mathrm{PA}\\ \rightarrow\end{array}\quad \mathrm{I}(UE)\right]^{\mathrm{I}}\left[\forall\,\mathrm{t: time.}\quad .\quad \begin{array}{c}\mathrm{TA}\\ \rightarrow\end{array}\quad \mathrm{I}(t)\right]^{\mathrm{I}}$$

- The intruder is entitled to look up any session key, or long-term (database) keys he owns as follows:

$$\left[\begin{array}{l}\forall\,\mathrm{MME: ts.}\\ \forall\,\mathrm{k: shK\ I\ UE.}\end{array}\quad .\quad \begin{array}{c}\mathrm{SA1'}\\ \rightarrow\end{array}\quad \mathrm{I}(k)\right]^{\mathrm{I}}\left[\forall\,\mathrm{k: dbK\ I.}\quad .\quad \begin{array}{c}\mathrm{DA'}\\ \rightarrow\end{array}\quad \mathrm{I}(k)\right]^{\mathrm{I}}$$

- The intruder may store these data in the I (., ., .,) predicate for later use.

III. INTRA-MME HANDOVER KEY MANAGEMENT PROTOCOL

When an UE detects the need to hand over to another node, it sends a measurement report to the source eNB that includes all the candidate eNBs for handover (i.e., message (S1)). Upon receiving the measurement report, source node eNB1 will choose the target eNB (eNB2). The eNB1 will then generate the new K_{eNB}* using the fresh NH_{NCC} key received from the MME and (α_1) target physical cell identity and frequency of eNB1 (i.e., Equation (1)).

$$K_{eNB}\text{*= KDF } (NH_{NCC}, \alpha_1) \qquad (1)$$

Then eNB1 forwards K_{eNB}* with the NCC value to eNB2, (i.e., message (S2)). The subsequent session key (K_{eNB}**) between UE and eNB2 is derived directly from the new K_{eNB}* and α_2.

$$\text{KeNB**= KDF } (K_{eNB}\text{*}, \alpha_2) \qquad (2)$$

Also, eNB2 sends the NCC value to UE, (i.e., message (S3)). The UE compares the received NCC with the value associated with the current security association (i.e., NCC-1). If they are the same, UE uses the NCC value to update the NH_{NCC} key, (i.e., Equation (3)). And then derive the new key using Equations (2) and (3) and sends the handover confirmation to eNB2, (i.e., message (S4)).

$$NH_{NCC} = KDF\ (K_{ASME}, NH_{NCC-1}) \quad (3)$$

When eNB2 completes the handover signaling with UE, it sends the *S1 path switch request*, (i.e., message (S5)) to the MME to increment the NCC value by 1, and computes a new NH (i.e., NH_{NCC+1}) from the K_{ASME} and current NH key. Then, the MME forwards the fresh NH_{NCC+1} and NCC+1 to the eNB2 to be used in the next handover, (i.e., message (S6)). The sequence of messages is summarized below.

msg (S1) *UE → eNB1* : *Measurement Report*
msg (S2) *eNB1→ eNB2* : K_{eNB}*, *NCC*
msg (S3) *eNB2→ UE* : *eNB2, NCC*
msg (S4) *UE → eNB2* : *Handover Confirmation*
msg (S5) *eNB2→ MME* : $\{eNB2, UE, NH_{NCC}, NCC\}\ K_{IP2}$ *//S1 path switch request*
msg (S6) *MME → eNB2* : $\{NH_{NCC+1}, NCC+1\}K_{IP2}$ *//S1 path switch ACK*

3.1 Roles generation theory

Figures 1, 2, and 3 show the user, source eNB, and target eNB generic roles. Figure 4 shows the anchored role of the MME server in the intra-MME handover key management standard protocol formalization.

$\forall eNB1, eNB2: eNodeB.\ \forall K_{eNB}: shK\ UE\ eNB1.\ \forall K_{eNB}**: shK\ UE\ eNB2.\ \forall NCC: msg.$

∀ UE: User

$N_{3.1}$ (eNB2, NCC)
$M_{1.0}$ (UE, eNB1, $K_{eNB,}$ NCC) $\beta_{1,1} \rightarrow$ ∃ K_{eNB}**: shK UE eNB2. $N_{1.1}$ (HO OK) $L_{1.1}$ (UE, eNB2, K_{eNB}**)
IF Valid$_{UE}$ (*NCC*)

Figure 1. User's Role in the Standard Protocol

∀ eNB1: eNodeB

$\forall UE: User.\ \forall eNB2: eNodeB.\ \forall K_{eNB}: shK\ UE\ eNB1.\ \forall K_{eNB}*: shK\ UE\ eNB2.$

$L_{2.0}$(eNB1, UE, eNB2, K_{eNB}) $\beta_{2,1} \rightarrow$ ∃ K_{eNB}*: *shK* UE eNB2. $N_{2.1}(K_{eNB}*, NCC)$ $L_{2.1}$(eNB1, UE, eNB2, $K_{eNB,}$ K_{eNB}*)

Figure 2. Source Node's Role in the Standard Protocol

∀ eNB2: eNodeB

$\forall UE: User.\ \forall eNB1: eNodeB.\ \forall K_{IP2}: dbK\ eNB2.\ \forall K_{eNB}**: shK\ UE\ eNB2.$
$\forall NH_{NCC}, NCC, NH_{NCC+1}, NCC+1: msg$

$N_{2.1}(K_{eNB}*, NCC)$ $L_{3.0}()$ $\beta_{3,1} \rightarrow$ ∃ K_{eNB}**: *shK* UE eNB2. $N_{3.1}$ (*eNB2, NCC*) $L_{3.1}$ (eNB2, UE, K_{eNB}**)

$N_{1.1}$ (*HO OK*) $L_{3.1}$ (eNB2, UE, K_{eNB}**) $\beta_{3,2} \rightarrow$ $N_{3.2}$ ($\{eNB2, UE, NH_{NCC}, NCC\}$ K_{IP2}) $L_{3.2}$ (eNB2, UE, K_{eNB}**)

$N_{4.1}(NH_{NCC+1}, NCC+1)$ $L_{3.2}$ (eNB2, UE, K_{eNB}**) $\beta_{3,3} \rightarrow$. $L_{3.3}$ (eNB2, UE, K_{eNB}**, NH_{NCC+1}, $NCC+1$)

Figure 3. Target Node's Role in the Standard Protocol

MME: Sever

$\forall UE: User.\ \forall eNB1, eNB2: eNodeB.\ \forall K_{IP2}: dbK\ eNB2.$
$\forall NH_{NCC}, NCC, NH_{NCC+1}, NCC+1: msg.$

$N_{3.2}$ ($\{eNB2, UE, NH_{NCC}, NCC\}$ K_{IP2}) $\beta_{4,1} \rightarrow$ ∃ NH_{NCC+1}, $NCC+1$: *msg*. $N_{4.1}(NH_{NCC+1}, NCC+1)$

Figure 4. MME Server's Role in the Standard Protocol

3.1 The Protocol Theory

The sample traces of the intra-MME handover key management standard protocol is depicted in Fig. 5. The UE, eNB1, eNB2 and MME are the bounded role theories, where $L_{2.0}$, $L_{3.0}$ and $L_{4.0}$ are the initial role states, and $L_{1.1}$, $L_{2.1}$ and $L_{3.1}$ are the role states.

$L_{2.0}$(eNB1, UE, eNB2, K_{eNB})	$\beta_{2,1}$ →	$L_{2.1}$(eNB1, UE, eNB2, $K_{eNB,}$ K_{eNB}*) $N_{2.1}$(*K_{eNB}*, *NCC*)
$L_{3.0}$() $N_{2.1}$(*K_{eNB}*, *NCC*)	$\beta_{3,1}$ →	$L_{3.1}$ (eNB2, UE, K_{eNB}**) $N_{3.1}$ (*eNB2*, *NCC*)
$M_{1.0}$ (UE, eNB1, $K_{eNB,}$ NCC) $N_{3.1}$ (*eNB2*, *NCC*)	$\beta_{1,1}$ →	$L_{1.1}$ (UE, eNB2, K_{eNB}**) $N_{1.1}$ (*HO OK*)
$L_{3.1}$ (eNB2, UE, K_{eNB}**) $N_{1.1}$ (*HO OK*)	$\beta_{3,2}$ →	$L_{3.2}$ (eNB2, UE, K_{eNB}**) $N_{3.2}$ ({*eNB2*, *UE*, NH_{NCC}, *NCC*} K_{IP2})
$N_{3.2}$ ({*eNB2*, *UE*, NH_{NCC}, *NCC*} K_{IP2})	$\beta_{4,1}$ →	$N_{4.1}$(NH_{NCC+1}, *NCC+1*)
$L_{3.2}$ (eNB2, UE, K_{eNB}**) $N_{4.1}$(NH_{NCC+1}, *NCC+1*)	$\beta_{3,3}$ →	$L_{3.3}$ (eNB2, UE, K_{eNB}**, NH_{NCC+1}, *NCC+1*) .

Figure 5. Sample Traces of the Standard Protocol

3.2The De-Synchronization Attack

An intruder can control a rogue base station which is a mobile device and impersonates a legitimate base station, either by compromising a commercial station or by deploying a personal station through physical, host, or network protocol vulnerabilities [6]. The goal of the rogue base station attack is to disrupt the updating of the NCC value. Either by manipulating the message between eNBs or by manipulating the S1 path switch ACK, leaving the target eNB desynchronized and the future sessions keys vulnerable. The effect of the de-synchronization attack lasts until K_{ASME} is revoked through the Evolved Packet System Authentication and Key Agreement (EPS-AKA) procedure between MME and UE. In this process, the new session key and subsequent security contexts are freshly created. Figure 6 presents the intruder rule of the standard protocol traces in case of de-synchronization attack.

I_0(eNB1, UE, eNB2, K_{eNB}, NCC, NH_{NCC})	$\beta_{1,1}$ →	∃ NCC': msg. ∃ K_{eNB}*: shK UE eNB2. **I_1(eNB1, UE, eNB2, K_{eNB}, K_{eNB}*)** $N_{2.1}$ (K_{eNB}*, **NCC'**)
$L_{3.0}$() **$N_{2.1}$** (K_{eNB}*, **NCC'**)	$\beta_{3,1}$ →	$L_{3.1}$ (eNB2, UE, K_{eNB}**) $N_{3.1}$ (eNB2, **NCC'**)
$M_{1.0}$ (UE, eNB1, $K_{eNB,}$ NCC) $N_{3.1}$ (eNB2, **NCC'**)	$\beta_{1,1}$ →	$L_{1.1}$ (UE, eNB2, K_{eNB}**) $N_{1.1}$ (HO OK)
$L_{3.1}$ (eNB2, UE, K_{eNB}**) $N_{1.1}$ (HO OK)	$\beta_{3,2}$ →	$L_{3.2}$ (eNB2, UE, K_{eNB}**) $N_{3.2}$ ({eNB2, UE, NH_{NCC}, **NCC'**} K_{IP2})
$N_{3.2}$ ({eNB2, UE, NH_{NCC}, **NCC'**} K_{IP2})	$\beta_{4,1}$ →	$N_{4.1}$(NH_{NCC+1}, **NCC'**+1)
$L_{3.2}$ (eNB2, UE, K_{eNB}**) $N_{4.1}$(NH_{NCC+1}, **NCC'**+1)	$\beta_{3,3}$ →	$L_{3.3}$ (eNB2, UE, K_{eNB}**, NH_{NCC+1}, **NCC'**+1) .

Figure 6. Intruder Rules in the Standard Protocol Traces

As shown in Figure 6, using an initial knowledge I_0 (eNB1, UE, eNB2, K_{eNB}, NCC, NH_{NCC}), the intruder impersonating the genuine eNB may launch rule $\beta_{1,1}$. The intruder may apply the data generation rules presented in section 2.2.2 to generate NCC' (much higher than the value of NCC) of type msg, and use equation (4) to generate the new session key K_{eNB}* from the previous session key K_{eNB}. Also the intruder may update his knowledge by adding the new session key to I_1 (eNB1, UE, eNB2, K_{eNB}, K_{eNB}*). Then the intruder may use the pairing rules presented in 2.2.1 to compose a new message {K_{eNB}*, NCC'} and forward it to the target eNB.

$$K_{eNB}^* = KDF(K_{eNB}, \alpha_1). \quad (4)$$

As a result, the target eNB will derive the new key K_{eNB}** using equation (2), in which the key K_{eNB}* is derived from the previous key K_{eNB}. In addition, the target eNB will forward NCC' to the UE, which will be de-synchronized because the NCC and NCC' are not the same. That enforces the UE to derive the next session key K_{eNB}** based on the current K_{eNB}* (derived from K_{eNB}) instead of using the NH_{NCC+1} Key. Consequently, the intruder will learn the new session key K_{eNB}** and use it for further attacks. The intruder may then intercept network messages between the user and the target eNB, and can decrypt the messages and compromise the data.

IV. THE PROPOSED PROTOCOL

The main idea of the proposed protocol is to keep the source eNB out, and involve the MME as a third trusted party (TTP) during the handover process. MME prepares the challenges needed to authentication the UE and the target eNB. In addition, MME generates the fresh materials needed to drive the new session key, and sends these materials to UE protected by the local root key K_{ASME} and to the target eNB physically protected or encrypted with the pre-shared IP-Sec association keys K_{IP} between the core network and the eNBs.

As in the standard protocol, when an UE detects the need to hand over to another node, it sends a measurement report to the eNB1 that includes all the candidate eNBs for handover (i.e., message (P1)). Upon receiving the measurement report, eNB1 will choose the target eNB (eNB2). In the proposed protocol, eNB1 will not generate the new session key K_{eNB}* but it will send a handover request to the MME server including the UE and the eNB2 names and also a freshly generated time stamp t, all encrypted under K_{IP1}, (i.e., message (P2)). MME will then send back the handover response to eNB1 including an authenticator to authenticate the UE, (i.e., message (P3)). The authenticator contains the received time stamp t, a freshly generated nonce n, and the UE name, all encrypted under K_{ASME} known only to the UE and the MME server. Also MME will send the contents of the authenticator to eNB2, encrypted under K_{IP2}, (i.e., message (P5)). Upon receiving this message, eNB2 will decrypt the message to retrieve its contents and generate the new session key K_{eNB}* using Equation (5), then wait for the user request.

$$K_{eNB}^{*} = KDF(n, \alpha_2) \qquad (5)$$

Upon receiving the handover response, (i.e., message (P3)), eNB1 will decrypt the message and forward the authenticator to the UE with the eNB2 name, (i.e., message (P4)), all encrypted under $K_{eNB.}$ UE will then decrypt the message and learn the target eNB2 and decrypt the authenticator to retrieve the parameters needed to generate the new session key K_{eNB}* using Equation (4). Then, UE will send an authentication request to eNB2 that includes its name and the hash of the nonce n, all encrypted under the new session key K_{eNB}*, (i.e., message (P6)).

The eNB2 will authenticate the UE by comparing the received hash with the calculated one. If they are not the same, eNB2 will abort the process. But if they are the same, eNB2 will compose a message containing the nonce n and the time stamp t, and send it to the UE encrypted under the new session key K_{eNB}* (i.e., message (P7)) to authenticate itself to the UE. Upon receiving message (P7), UE will decrypt the message and check the nonce n and the time stamp t. If they are not the same, UE will abort the process. But if they are the same, UE will authenticate eNB2 and sends back a handover confirmation message, (i.e., message (P8)). The eNB2 will forward the handover confirmation message to acknowledge the MME server that the handover process is successfully accomplished (i.e., message (P9)). The sequence of messages is summarized below.

msg (P1)	*UE → eNB1*	: *Measurement Report*	
msg (P2)	*eNB1 → MME*	: $\{UE, eNB2, t\}K_{IP1}$	*// handover request*
msg (P3)	*MME → eNB1*	: $\{UE, eNB2, Authenticator\}K_{IP1}$	*// handover response*
msg (P4)	*eNB1→ UE*	: $\{eNB2, Authenticator\}K_{eNB}$	
msg (P5)	*MME → eNB2*	: $\{UE, n, t\}K_{IP2}$	
msg (P6)	*UE → eNB2*	: $\{UE, h(n)\}K_{eNB*}$	
msg (P7)	*eNB2→ UE*	: $\{n, t\}K_{eNB*}$	
msg (P8)	*UE → eNB2*	: *Handover Confirmation*	
msg (P9)	*eNB2→ MME*	: *Handover Acknowledgement*	

4.1 Roles generation theory

Figure 7, 8, and 9 show the UE, source eNB, and target eNB generic roles. Figure 10 shows the anchored role of MME server in our formalization. The protocol initiator eNB1, may use rule $\alpha_{2,1}$ shown in Figure (9), to send an encrypted message under K_{IP1}, to the MME server requesting the handover. In addition, eNB1 stores the information from the request in a role state predicate $L_{2.1}$. Upon receiving the handover request $N_{2.1}$, MME checks UE, eNBs names and the timestamp using the validation check $Valid_{MME}$. Then MME may use rule $\alpha_{4,1}$ shown in Figure (10) to send the handover response $N_{4.1}$ to eNB1, encrypted under K_{IP1}, including a challenge to UE. The challenge is encrypted under K_{ASME}, contains UE, a fresh generated nonce n, and $t_{eNB1, MME}$. In addition, MME may use rule $\alpha_{4,2}$ shown in Figure (10) to send message $N_{4.2}$ that has the same contents

of the challenge to eNB2, encrypted under K_{IP2} to authenticate UE. The MME does not have to store any further information as a role state predicate for this rule because it has all the information stored in the database.

∀ UE: user

∀ eNB1, eNB2: eNodeB. ∀ K_{eNB1}: shK UE eNB1. ∀ K_{ASME}: dbK UE. ∀ K_{eNB2}: shK UE eNB2. ∀ n: nonce. ∀ $t_{eNB1, MME}$: time.

$N_{2.2}$(enc (K_{eNB1}, <UE, eNB2, enc (K_{ASME}, <UE, n, $t_{eNB1, MME}$>)>)) $M_{1.0}$ (UE, eNB1, K_{eNB1}) **IF Valid$_{UE}$ (UE, $t_{eNB1, MME}$)**	$\alpha_{1,1}$ →	∃K_{eNB2}: shK UE eNB2. $N_{1.1}$ (enc (K_{eNB2}, < UE, H(n)>)) $L_{1.1}$ (UE, eNB2, n, $t_{eNB1, MME}$, K_{eNB2})
$N_{3.1}$ (enc (K_{eNB2}, < n, $t_{eNB1, MME}$ >)) $L_{1.1}$ (UE, eNB2, n, $t_{eNB1, MME}$, K_{eNB2}) **IF Valid$_{UE}$ (n, $t_{eNB1, MME}$)**	$\alpha_{1,2}$ →	. $M_{1.1}$ (UE, eNB2, K_{eNB2})

Figure 7. User's Role in the Proposed Protocol

∀ eNB1: eNodeB

∀UE: User. ∀ K_{IP1}: dbK eNB1. ∀ K_{eNB1}: shK UE eNB1. ∀ $t_{eNB1, MME}$: time. ∀ t_V: time. ∀ t_{exp}: time. ∀ X: cipher. ∀t_{now}: time.

$L_{2.0}$(eNB1, UE, eNB2, K_{eNB1}) $Clock_{eNB1}$($t_{eNB1, MME}$) (t_{exp} = $t_{eNB1, MME}$ + t_V)	$\alpha_{2,1}$ →	$N_{2.1}$(enc (K_{IP1}, <UE, eNB1, eNB2, $t_{eNB1, MME}$>)) $L_{2.1}$(eNB1, UE, $t_{eNB1, MME}$, t_{exp}, eNB2, K_{eNB1})
$N_{4.1}$(enc (K_{IP1}, <UE, eNB2, $t_{eNB1, MME}$, X>)) $L_{2.1}$(eNB1, UE, $t_{eNB1, MME}$, t_{exp}, eNB2, K_{eNB1}) **IF Valid$_{eNB1}$ (eNB1, UE, $t_{eNB1, MME}$, t_{exp}), Clock$_{eNB1}$(t_{now})**	$\alpha_{2,2}$ →	$N_{2.2}$(enc(K_{eNB1}, <UE, eNB2, X>))

Figure 8. Source Node's Role in the Proposed Protocol

∀ eNB2: eNodeB

∀UE: User. ∀ K_{IP2}: dbK eNB2. ∀ K_{eNB2}: shK UE eNB2. ∀ $t_{eNB1, MME}$: time. ∀ n: nonce.

$N_{4.2}$ (enc (K_{IP2}, <UE, n, $t_{eNB1, MME}$>)) $L_{3.0}$()	$\alpha_{3,1}$ →	. $L_{3.1}$(eNB2, UE, n, $t_{eNB1, MME}$, K_{eNB2})
$N_{1.1}$ (enc (K_{eNB2}, < UE, H(n)>)) $L_{3.1}$(eNB2, UE, n, $t_{eNB1, MME}$, K_{eNB2}) **IF Valid$_{eNB2}$ (UE, H(n))**	$\alpha_{3,2}$ →	∃ K_{eNB2}: *shK* UE eNB2. $N_{3.1}$ (enc (K_{eNB2}, < n, $t_{eNB1, MME}$ >)) $M_{3.1}$ (eNB2, UE, K_{eNB2})

Figure 9. Target Node's Role in the Proposed Protocol

MME: Sever

∀UE: User. ∀eNB1, eNB2: eNodeB. ∀ K_{ASME}: dbK UE. ∀ K_{IP1}: dbK eNB1. ∀ K_{IP2}: dbK eNB2. ∀ n: nonce. ∀ $t_{eNB1, MME}$, t_{now}: time.

$N_{2.1}$(enc (K_{IP1}, <UE, eNB1, eNB2, $t_{eNB1, MME}$>)) **IF Valid$_{MME}$ (UE, eNB1, eNB2, $t_{eNB1, MME}$), Clock$_{MME}$ (t_{now})**	$\alpha_{4,1}$ →	∃ n: nonce. $N_{4.1}$(enc (K_{IP1}, <UE, eNB2, $t_{eNB1, MME}$, enc (K_{ASME}, <UE, n, $t_{eNB1, MME}$>)>))
	$\alpha_{4,2}$ →	$N_{4.2}$ (enc (K_{IP2}, <UE, n, $t_{eNB1, MME}$>))

Figure 10. MME Server's Role in the Proposed Protocol

The eNB1 expects the response $N_{4.1}$ from MME within a certain time, encrypted under one of its database keys K_{IP1}, including a challenge (an opaque message). If a message of this form appears on the network, eNB1 uses the role state predicate $L_{2.1}$ and the validity check $Valid_{eNB1}$ to ensure that eNB1, UE, and $t_{eNB1,MME}$ in this message match those in its original request. If they are the same, eNB1 may read this message

from the network and save the relevant information using rule $\alpha_{2,2}$ shown in Figure (8), which replaces the facts $N_{4.1}$ and $L_{2.1}$ with the fact $N_{2.2}$ (contains, UE, eNB2, and challenge X, encrypted under K_{eNB1}) to be sent to UE.

The UE expects the response $N_{2.2}$ from eNB1, including a challenge encrypted under K_{ASME}. If a message of this form appears on the network, UE uses the memory predicate $M_{1.0}$ and the validation check $Valid_{UE}$ to check UE's name, and the time stamp $t_{eNB1,MME}$. Then UE may read this message from the network and save the relevant information, using rule $\alpha_{1,1}$ shown in Figure (7), to generate K_{eNB2} and replaces the facts $N_{2.2}$ and $M_{1.0}$ with the fact $N_{1.1}$ and $L_{1.1}$. UE will send $N_{1.1}$ to eNB2 to authenticate herself.

If a message of the form, $N_{4.2}$ from the MME server appears on the network, encrypted under K_{IP2}, eNB2 will read this message from the network and store the relevant information, in a role state predicate $L_{3.1}$ after generating K_{eNB2} using n and $t_{eNB1,\ MME}$. At this time, eNB2 expects a request message $N_{1.1}$ from UE, contains UE, and the hash of n. If a message of this form appears on the network, eNB2 uses the role state predicate $L_{3.1}$ and validation check $Valid_{eNB2}$ to authenticate UE by computing the hash of n and comparing it with the received one. If they are the same, eNB2 will read the message from the network and save the relevant information, using rule $\alpha_{3,1}$ shown in Figure (9), which replaces the facts $N_{1.1}$ and $L_{3.1}$ with the fact $N_{3.1}$ and memory predicate $M_{3.1}$. the eNB2 will then send $N_{3.1}$ to UE to authenticate itself. The UE expects a response $N_{3.1}$ from eNB2. If a message of this form appears on the network, UE uses the role state predicate $L_{1.1}$ and validation check $Valid_{UE}$ to authenticate eNB2 by checking n and $t_{eNB1,\ MME}$. If they are the same, UE may read this message from the network and save the relevant information in the memory predicate $M_{1.1}$ using rule $\alpha_{1,2}$ shown in Figure (7), which sends an acknowledge to eNB2 that the handover is done successfully.

4.3 The Protocol Theory

The Sample trace of the proposed protocol in Figure (11) shows that UE, eNB1, eNB2 and MME are the bounded role theories, where L2.0, L3.0 and L4.0 are the initial role states, while the L1.1, L2.1 and L3.1 are the role states. From the formalization, we can deduce that the protocol is formed as a well-founded protocol theory, such that: the knowledge of one participant is separated from the knowledge of another. Also the private knowledge is separated from public knowledge, and the state of a participant from a network message. Also the initialization theory was separated from the key distribution and role-assignment phase and the protocol execution phase, in a way that reflects the bounded nature of the proposed protocol.

$L_{2.0}$(eNB1, UE, eNB2, K_{eNB1}) .	$\alpha_{2,1}$ →	$L_{2.1}$(eNB1, UE, $t_{eNB1,\ MME}$, t_{exp}, eNB2, K_{eNB1}), $N_{2.1}$(enc (K_{IP1}, <UE, eNB1, eNB2, $t_{eNB1,\ MME}$>))
$L_{4.0}$(), $N_{2.1}$(enc (K_{IP1}, <UE, eNB1, eNB2, $t_{eNB1,\ MME}$>))	$\alpha_{4,1}$ →	. $N_{4.1}$(enc (K_{IP1}, <UE, eNB2, $t_{eNB1,\ MME}$, enc(K_{ASME}, <UE, n, $t_{eNB1,\ MME}$>)>))
	$\alpha_{4,2}$ →	. $N_{4.2}$ (enc (K_{IP2}, <UE, n, $t_{eNB1,\ MME}$>))
$L_{2.1}$(eNB1, UE, $t_{eNB1,MME}$, t_{exp}, eNB2, K_{eNB1}), $N_{4.1}$(enc (K_{IP1}, <UE, eNB2, $t_{eNB1,\ MME}$, X>))	$\alpha_{2,2}$ →	. $N_{2.2}$(enc (K_{eNB1}, <UE, eNB2, X>))
$L_{3.0}$(), $N_{4.2}$ (enc (K_{IP2}, <UE, n, $t_{eNB1,\ MME}$>))	$\alpha_{3,1}$ →	$L_{3.1}$ (eNB2, UE, n, $t_{eNB1,\ MME}$, K_{eNB2}) .
$M_{1.0}$ (UE, eNB1, K_{eNB1}), $N_{2.2}$(enc (K_{eNB1}, <UE, eNB2, X>))	$\alpha_{1,1}$ →	$L_{1.1}$ (UE, eNB2, n, $t_{eNB1,\ MME}$, K_{eNB2}), $N_{1.1}$ (enc (K_{eNB2}, < UE, H(n)>))
$L_{3.1}$ (eNB2, UE, n, $t_{eNB1,\ MME}$, K_{eNB2}), $N_{1.1}$ (enc (K_{eNB2}, < UE, H(n)>))	$\alpha_{3,2}$ →	$M_{3.1}$ (eNB2, UE, K_{eNB2}), $N_{3.1}$ (enc (K_{eNB2}, < n, $t_{eNB1,\ MME}$ >))
$L_{1.1}$ (UE, eNB2, n, $t_{eNB1,\ MME}$, K_{eNB2}), $N_{3.1}$ (enc (K_{eNB2}, < n, $t_{eNB1,\ MME}$ >))	$\alpha_{1,2}$ →	$M_{1.1}$ (UE, eNB2, K_{eNB2}), .

Figure 11. Sample Trace of the Proposed Protocol

4.5 Previnting The De-Synchronization Attack

Initially, the intruder impersonating the genuine eNB may use the data access rules presented in 2.2.3 to access the name of any principal (UE, eNB1, eNB2, or MME), and also lookup the session key K_{eNB1} and the long-term (database) key K_{IP1} to have an initial knowledge I_0 (eNB1, UE, eNB2, K_{eNB1}, K_{IP1}). The intruder can also access the defined time stamps {$t_{eNB1,MME}$, t_{exp}}, and compose any message using the rules presented in section 2.2.1.

As shown in Figure 12, with the initial knowledge I_0, intruder could lunch the rule $\alpha_{I,1}$, compose the message $N_{2.1}${UE, eNB1, eNB2, $t_{eNB1,MME}$} and encrypt it using K_{IP1}, then forward it to the MME as a handover request. Also the intruder may store these data in the intruder predicate I_1(eNB1, UE, $t_{eNB1,\ MME}$, t_{exp}, eNB2, K_{eNB1}) for later use. Upon receiving the network message $N_{4.1}$ from the MME server as a handover response, the

intruder may launch another rule $\alpha_{I,2}$. The intruder may decrypt the message $N_{4.1}$ using K_{eNB1}, and decompose its contents. Then the intruder will try to delete the challenge X (cipher) and generate a generic message Z, which is not the encryption of any other message. Then the intruder may send message $N_{2.2}$(enc (K_{eNB1}, <UE, eNB2, Z >)) to the UE, and update his knowledge I_1 by adding the two values X and Z to I_2(eNB1, UE, $t_{eNB1,MME}$, t_{exp}, eNB2, K_{eNB1}, X, Z).

In this case, upon receiving the message $N_{2.2,}$ the UE will first decrypt the message using K_{eNB1} then decrypt the message Z using K_{ASME} and perform the validation test ValidUE (UE, $t_{eNB1, MME}$). The UE will detect the attack because the validation test ValidUE will fail and then abort the handover process in rule $\alpha_{1,1}$. Otherwise, the intruder may forward the challenge X as received from the server MME to the UE. In this case, the handover process will take place and the new session key will be derived and shared between the UE and the target eNB2 away from the intruder. Consequently, the intruder will not learn the new session key.

I_0(eNB1, UE, eNB2, K_{eNB1}, K_{IP1}) .	**$\alpha_{I,1}$** →	**I_1(eNB1, UE, $t_{eNB1, MME}$, t_{exp}, eNB2, K_{eNB1}),** $N_{2.1}$(enc (K_{IP1}, <UE, eNB1, eNB2, $t_{eNB1, MME}$>))
$L_{4.0}$(), $N_{2.1}$(enc (K_{IP1}, <UE, eNB1, eNB2, $t_{eNB1, MME}$>))	$\alpha_{4,1}$ →	. $N_{4.1}$(enc (K_{IP1}, <UE, eNB2, $t_{eNB1, MME}$, enc(K_{ASME}, <UE, n, $t_{eNB1, MME}$>)>))
	$\alpha_{4,2}$ →	. $N_{4.2}$ (enc (K_{IP2}, <UE, n, $t_{eNB1, MME}$>))
I_1(eNB1, UE, $t_{eNB1,MME}$, t_{exp}, eNB2, K_{eNB1}), $N_{4.1}$(enc (K_{IP1}, <UE, eNB2, $t_{eNB1, MME}$, X>))	**$\alpha_{I,2}$** →	**I_2(eNB1, UE, $t_{eNB1,MME}$, t_{exp}, eNB2, K_{eNB1}, X, Z),** $N_{2.2}$(enc (K_{eNB1}, <UE, eNB2, Z >))
$L_{3.0}$(), $N_{4.2}$ (enc (K_{IP2}, <UE, n, $t_{eNB1, MME}$>))	$\alpha_{3,1}$ →	$L_{3.1}$ (eNB2, UE, n, $t_{eNB1, MME}$, K-eNB2) .
$M_{1.0}$ (UE, eNB1, K_{eNB1}), $N_{2.2}$(enc (K_{eNB1}, <UE, eNB2, Z >)) ***IF $Valid_{UE}$ (UE, $t_{eNB1, MME}$)***	$\alpha_{1,1}$ →	**ABORT**

Figure 12. Intruder Rules in the Proposed Protocol Traces

According to the formal analysis results, we can conclude that the standard protocol could not detect the de-synchronization attack since the intruder impersonating the genuine base station could launch the handover process and learn the new session key, as shown in the intruder rules traces in section (3.3). As a result, the standard protocol could not maintain the one-hop forward security and the new session key is compromised. On the other side, though the intruder can initiate the handover process, he cannot complete the handover process because the proposed protocol detects the attack and aborts the process at the UE side, as shown in the trace of the intruder rules in section (4.3). As a result, the proposed protocol maintains the one-hop forward security and protects the new session key from being compromised.

V. CONCLUSIONS

This paper proposes a provably secure authenticated key management protocol against the desynchronization attack in the LTE intra-MME handover. The proposed protocol keeps out the source eNB from the key management process, and uses the MME as a third trusted party. The MME sends the fresh materials needed to drive the new session key for both the user and the target eNB (away from the source eNB) protected by the pre-shared local root key K_{ASME} and the pre-shared IPsec association key $K_{IP2,}$ respectively. An overview of the Multi-Set Rewriting (MSR) formalism with existential quantification technique is presented. Also the signature fragment that sets up the typing infrastructure used in this paper and the rules specifying the Dolev-Yao intruder model are conducted. Formalizations of the LTE intra-MME handover standard protocol and the proposed protocol are conducted using MSR formalism under the Dolev–Yao intruder model, in a way that reflects their bounded nature.

The traces of the intra-MME handover key management protocol and the effect of the de-Synchronization attack showed that the protocol could not detect the attack. As a result, the intruder impersonating the genuine base station could carry out the handover process and learn the new session key after the handover takes place. On the other side; we illustrated in a formal manner that the proposed protocol can prevent the de-Synchronization attack. The intruder can initiate the handover process, but he cannot learn the new session key, since the proposed protocol detects the attack at the user side, and aborts the handover process.

REFERENCES

[1]. Dahlman, E., S. Parkvall, and J. Skold, 4G: LTE/LTE-advanced for mobile broadband. 2013: Academic press.

[2]. 3GPP, 3GPP System Architecture Evolution (SAE); Security architecture, in TS 33.401 Version 11.2.0. 2012: 3rd Generation Partnership Project.

[3]. 3GPP, 3GPP System Architecture Evolution (SAE); Security architecture, in TS 33.401 Version 11.56.0 2013: 3rd Generation Partnership Project.

[4]. Han, C.-K. and H.-K. Choi, Security analysis of handover key management in 4G LTE/SAE networks. IEEE Transactions on Mobile Computing, 2014. **13**(2): p. 457-468.
[5]. Zhang, M. and Y. Fang, Security analysis and enhancements of 3GPP authentication and key agreement protocol. IEEE Transactions on wireless communications, 2005. **4**(2): p. 734-742.
[6]. Park, Y. and T. Park. A survey of security threats on 4G networks. in IEEE Globecom Workshops, 2007.
[7]. Sridevi, B., D. Mohan, and R. Neelaveni. Secured Handover Key Management among LTE Entities Using Device Certification. in IEEE Eco-friendly Computing and Communication Systems (ICECCS), 2014.
[8]. Forsberg, D., LTE key management analysis with session keys context. Computer Communications, 2010. **33**(16): p. 1907-1915.
[9]. Xiao, Q., B. Cui, and L. Li. An Enhancement for Key Management in LTE/SAE X2 Handover Based on Ciphering Key Parameters. in IEEE P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2014.
[10]. Chandavarkar, B. Mitigation of desynchronization attack during inter-eNodeB handover key management in LTE. in IEEE Contemporary Computing (IC3), 2015.
[11]. Mao, W. A structured operational modelling of the dolev-yao threat model. in International Workshop on Security Protocols. 2002. Springer.
[12]. Cervesato, I., et al. A meta-notation for protocol analysis. in Computer Security Foundations Workshop. Proceedings of the 12th IEEE. 1999.
[13]. Cervesato, I., A specification language for crypto-protocols based on multiset rewriting, dependent types and subsorting. Computer Science Department, 2001: p. 38.
[14]. Cervesato, I. Typed MSR: Syntax and examples. in International Workshop on Mathematical Methods, Models, and Architectures for Network Security. 2001. Springer.
[15]. Mitchell, J., et al. Undecidability of bounded security protocols. in Workshop on Formal Methods and Security Protocols. 1999.
[16]. Butler, F., et al., Formal analysis of Kerberos 5. Theoretical Computer Science, 2006. **367**(1-2): p. 57-87.
[17]. Butler, F., et al., A formal analysis of some properties of Kerberos 5 using MSR. 2004.
[18]. Schmidt, B., et al. Automated analysis of Diffie-Hellman protocols and advanced security properties. in IEEE Computer Security Foundations Symposium (CSF), 2012.

Authers

Khaled Khairy received the BSc degree and the MSc degree in electrical engineering from the Military Technical College (MTC), Department of Communication at Cairo Egypt. He is a PhD candidate and in a visiting scholar in the Department of Computer Science at the University of Central Florida, in Orlando, Florida. He has nearly ten years' experience in security systems design and implementation. His research interests include security issues in LTE mobile networks, Modeling and Simulation.

Mainak Chatterjee is an associate professor in the Department of Computer Science at the University of Central Florida, in Orlando, Florida. He received the BSc degree in physics (Hons.) from the University of Calcutta, the ME degree in electrical communication engineering from the Indian Institute of Science, Bangalore, and the PhD degree from the Department of Computer Science and Engineering from the University of Texas at Arlington. His research interests include economic issues in wireless networks, applied game theory, cognitive radio networks, dynamic spectrum access, and mobile video delivery. He has been supported by research grants from federal, state, and local agencies. He has published more than 180 conference and journal papers. He got the Best Paper Awards in IEEE Globecom 2008 and IEEE PIMRC 2011, and the AFOSR sponsored Young Investigator Program (YIP) Award. He co-founded the ACM Workshop on Mobile Video (MoVid). He serves on the editorial board of Elseviers Computer Communications and Pervasive and Mobile Computing Journals. He has served as the TPC co-chair of a dozen conferences. He also serves on the executive and technical program committee of several international conferences.