Security Analysis of Handover Key Management in 4G LTE/SAE Networks

Chan-Kyu Han and Hyoung-Kee Choi

Abstract—The goal of 3GPP Long Term Evolution/System Architecture Evolution (LTE/SAE) is to move mobile cellular wireless technology into its fourth generation. One of the unique challenges of fourth-generation technology is how to close a security gap through which a single compromised or malicious device can jeopardize an entire mobile network because of the open nature of these networks. To meet this challenge, handover key management in the 3GPP LTE/SAE has been designed to revoke any compromised key(s) and as a consequence isolate corrupted network devices. This paper, however, identifies and details the vulnerability of this handover key management to what are called desynchronization attacks; such attacks jeopardize secure communication between users and mobile networks. Although periodic updates of the root key are an integral part of handover key management, our work here emphasizes how essential these updates are to minimizing the effect of desynchronization attacks that, as of now, cannot be effectively prevented. Our main contribution, however, is to explore how network operators can determine for themselves an optimal interval for updates that minimizes the signaling load they impose while protecting the security of user traffic. Our analytical and simulation studies demonstrate the impact of the key update interval on such performance criteria as network topology and user mobility.

Index Terms—Authentication and key agreement, evolved packet system, handover key management, long-term evolution security, mobile networks, system architecture evolution

1 Introduction

Recent increases in mobile data usage and the emergence of new applications drive the motivation to move the 3GPP into the fourth generation of cellular wireless technology. In response, designers of the 3GPP Long Term Evolution/System Architecture Evolution (LTE/SAE) system have announced the Evolved Packet System (EPS) as the fourth generation of the 3GPP mobile network. The access network used in the EPS network improves radio access technologies of the 3GPP mobile networks so as to offer a higher data rate with low latency. The EPS is also designed to support flat Internet Protocol (IP) connectivity and full interworking with heterogeneous radio access networks and service providers.

This architectural design decision brings to the fore implications of LTE/SAE for security. The flat all-IP architecture allows all radio access protocols to terminate in one node called evolved NodeB (eNodeB). In the Universal Mobile Telecommunications System (UMTS), the functionality of eNodeB was divided into NodeB and the Radio Network Controller (RNC). The placement of the radio access protocols in eNodeB makes them vulnerable to unauthorized access because eNodeB is located in unattended place. Further, internetworking with heterogeneous radio access networks exposes the vulnerability of these networks to direct external threats and carries grave implications for LTE security.

The unique characteristics of LTE/SAE gave rise to a number of features in the design of the security mechanism in the EPS network. Of these, key management in handovers [1], [26], [36] and minimizing the security risk involved is the focus of this paper. The main threat to handover key management is that an attack will compromise session keys in a base station. Handover key management typically alleviates this threat through separation of the session keys in a handover between base stations. This separation keeps a session key compromised in one base station from compromising another base station; in other words, the goal is to keep security breaches as local as possible.

For reasons of efficiency, handover preparations in LTE/SAE do not involve the core network. Source eNodeB provides a session key to target eNodeB for use after the handover. In this way, the core network does not need to maintain a state of individual User Equipment (UE). In this design, handing over an unchanged session key would permit target eNodeB to know which session key the source eNodeB used. To prevent this, the source eNodeB computes a new session key by applying a one-way function to a current session key. This ensures *backward key separation* in the handover. However, backward key separation blocks an eNodeB only from deriving past session keys from the current session key. Otherwise, this eNodeB would know all session keys used in further sessions in a whole chain of handovers. As a consequence, *forward key separation* was introduced to ensure that network elements add fresh materials to the process of creating a new session key for the next serving eNodeB. The current eNodeB, unaware of this additive, would be unable to derive the next key.

- *C.-K. Han is with Samsung Electronics, 129 Samsung-ro, Yeongtong-gu, Suwon, Geyonggi-do, South Korea. E-mail: ckyu.han@samsung.com.*
- *H.-K. Choi is with the Department of Computer Science and Engineering, School of Information & Communication Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon, Geyonggi-do, South Korea. E-mail: meosery@skku.edu.*

Manuscript received 1 Apr. 2011; revised 4 Feb. 2012; accepted 10 Nov. 2012; published online 28 Nov. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-04-0173.
Digital Object Identifier no. 10.1109/TMC.2012.242.

1536-1233/14/$31.00 © 2014 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

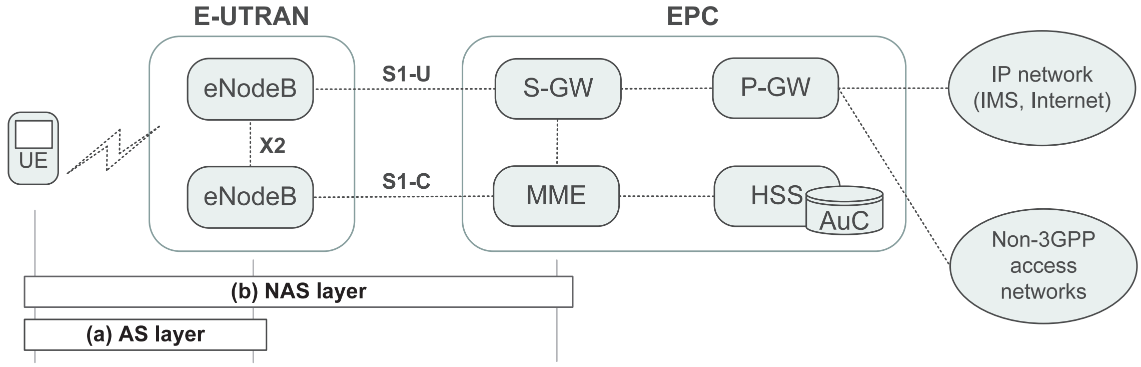


Fig. 1. EPS architecture composed of E-UTRAN and EPC.

We were able to demonstrate that, under certain circumstances, handover key management fails to ensure forward key separation against a variant attack by a rogue base station; such an attack is herein referred to as a desynchronization attack. A desynchronization attack prevents a target eNodeB from maintaining the freshness of the handover key. The vulnerability of this synchronization to disruption represents a potential security flaw in handover key management that could allow an adversary to compromise all future keys between a specific user and subsequent eNodeBs.

This attack may continue until the next update of the root key when handover key materials are generated from scratch instead of by derivation from the previous key. At this point, a potentially devastating effect through a compromised key comes to an end. Without delving into the technical challenges of a specific solution to prevent a desynchronization attack, the most practical remedy is to periodically refresh the root key. A very short-term root key seems an intuitive solution to minimizing the impact of a compromised key. However, frequent refreshing is not considered the best operational choice because of the signaling load that such root key updating imposes. On the other hand, the longer the update interval the more packets are exposed to a desynchronization attack.

The key question network operators and service providers might have is how to effectively choose a root key update interval that is the best balance between the signaling load and the number of user data packets exposed to attack because of a compromised handover key. Unfortunately, because this value is so dependent on time and place, a universally acceptable interval does not exist. Nor are there any proven ways to arrive at acceptable tradeoffs appropriate to different circumstances. In the face of this threat to the next generation of cellular networks, the motivation of this paper is to determine how to formulate this value to fit the circumstances of time and place.

As a first step toward a formula for an acceptable tradeoff, we diagramed the timing of handover key management in terms of the root key update interval as a way to measure the period during which a compromised key is operative. We then investigated a mathematical model to measure the expected operative period of the compromised key and to represent the expected value of the signaling load and volume of compromised packets during this period. Our methodology permits optimal management of the root key update interval according to network policies. This optimal interval is a value that minimizes the signaling traffic overhead required to update the root key while simultaneously limiting the volume of packets exposed to the compromised key.

The main contributions of this paper are threefold: 1) We identified flaws in the handover key management of the EPS security mechanism; 2) we designed a promising mathematical model for the EPS handover key management to measure the effect of a compromised key; and 3) we investigated the performance criteria (e.g., user mobility, network topology, and so on) involved in selecting an optimal operational point for key updating. Extensive simulation results validate the analytical model and reveal how the optimal key update interval changes in practice.

The rest of the paper is organized as follows: Section 2 contains an overview of EPS security, including handover key management. Section 3 discusses the security flaws in handover key management. In Section 4, we use a mathematical model to analyze the length of exposure to a compromised key during handover key management and to determine the tradeoff between the signaling load and the volume of compromised packets during this period. Section 5 evaluates the accuracy of the model and reports the empirical results obtained with realistic mobility models. In Section 6, we present an optimal key management and investigate the implementation issues with network providers. In Section 7, we review the literature related to the security and mathematical analysis of 3GPP networks before presenting our conclusions in Section 8.

2 Description of EPS Security

Some design decisions were made in the security of the EPS. These decisions were made after taking into consideration both practical implementation issues and performance issues.

2.1 Design Decisions on EPS Security

The EPS architecture as shown in Fig. 1 is composed of the access network and the core network, which are the Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and the Evolved Packet Core (EPC), respectively.

The design goal of the E-UTRAN is to adapt flat and all-IP network architecture so as to efficiently and flexibly deliver and distribute mobile services. The E-UTRAN is designed to be flat by integrating the functions of the hierarchically deployed NodeB and RNC in the access network of the UMTS. The architectural change has shifted the termination point of the air interface from the RNC in

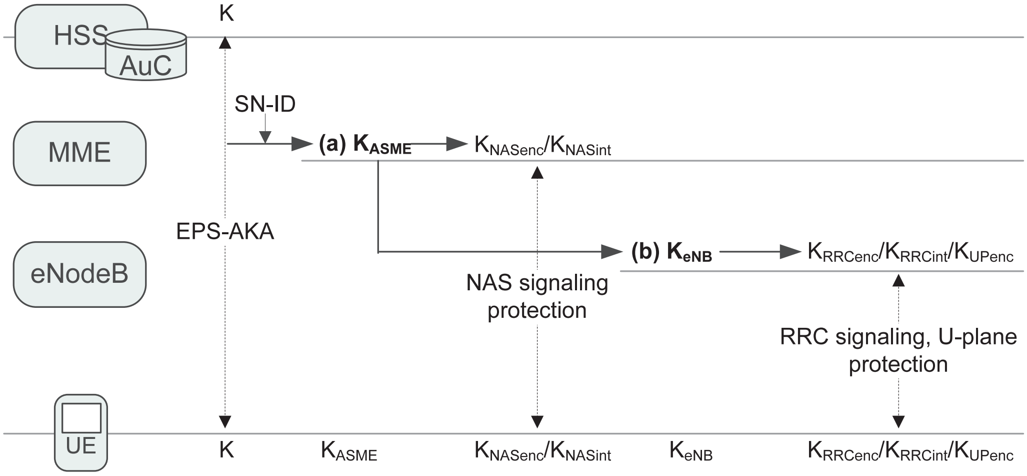


Fig. 2. Extended key hierarchy in the EPS security with intermediate keys; (a) $\mathrm{K_{ASME}}$ and (b) $\mathrm{K_{eNB}}$, respectively, protect the NAS and AS layers.

the UMTS to eNodeB in the EPS. Such a termination point would constitute a security weakness. As a base station in the EPS, eNodeB is located at an exposed location and connected to the core network over the IP layer. In an effort to make eNodeB secure, the two layers of LTE security protect traffic passing through it. The first layer, called the Access Stratum (AS) layer (see (a) in Fig. 1), enforces security between the UE and eNodeB. This layer is created when data in radio links need to be exchanged and protects the signaling and user data. In contrast, the second layer, called the Nonaccess Stratum (NAS) layer (see (b) in Fig. 1), remains active whenever the UE is registered to the network and is responsible for securing the signaling in the region between the UE and the Mobility Management Entity (MME). Concerns about insecure links beyond the MME are the responsibility of the optional IP Security (IPSec) association between network elements.

One of the changes in the EPS is separation between the Control plane (C-plane) signaling traffic and the User data plane (U-plane) data traffic. A C-plane signaling traffic path, designated as *S1-C* in Fig. 1, is established between a UE and an MME, and a path for the U-plane data traffic, designated as *S1-U* in Fig. 1, is set up between a UE and a Serving Gateway (S-GW). This new change implies not only physically separate paths for these two types of traffic but also separate key management for encryption and integrity protection. The next section discusses the extended key hierarchy and key management in EPS security.

2.2 Extended Key Hierarchy in EPS-AKA

The key hierarchy in the EPS is considerably elaborate and extended for efficient managements of the increased number of keys. The MME hosts the Access Security Management Entity (ASME) to handle access security and acts as a key distributor in the EPS security. The first intermediate keys (see (a) K_{ASME} in Fig. 2) are derived and distributed to the MME to protect the NAS layer. Further, the second intermediate keys (see (b) K_{eNB} in Fig. 2) are derived in the MME and distributed to eNodeB to protect the AS layer.

Each time a UE registers itself with an EPS network, an Authentication and Key Agreement (EPS-AKA) [1], [2] occurs between a UE and the MME on behalf of the Home Subscriber Server (HSS)/Authentication Center (AuC). The EPS-AKA is the EPS security mechanism to execute 1) authentication between a UE and an MME on behalf of the HSS/AuC, and 2) a key agreement between a UE and an MME as well as between a UE and eNodeB. Once mutual authentication succeeds, the two parties generate the first intermediate key, K_{ASME}, from the permanent master key, K. In the course of performing EPS-AKA, the HSS/AuC delivers the first intermediate key to the MME after binding to the serving network identity (*SN-ID* in Fig. 2). Clearly, the evolution to LTE and its flat all-IP core network emphasizes the urgent need for a revision of the trust relationships between operators and network components. Any threats arising from untrusted networks are alleviated in the EPS by a new feature, namely cryptographic network separation. Network separation tries to isolate the impact of any security breach in the local network and prevent its spillover to other networks. This is achieved by binding any cryptographic keys to the identity of the serving network for which the keys are intended. The UE can ensure that it communicates with the intended serving network by authenticating an identity in the current network. In the UMTS, a UE was unable to authenticate a serving network [3].

The local master key, K_{ASME}, also called the first intermediate key, is valid at a maximum interval determined by the timing of the next EPS-AKA procedure. The UE can choose to invoke the EPS-AKA protocol whenever the serving MME changes because of roaming to another serving network. In the same situation, the UE also can choose to transfer the security context between the old and new MMEs in an effort to lower the overhead of the full EPS-AKA. The UE may, of course, also need to run the EPS-AKA protocol periodically without interrupting service. Hence, the frequency of EPS-AKA runs is rather random or configurable by a network operator. In general, the lifetime of K_{ASME} varies from a few hours to a couple of days [36].

As shown in Fig. 2, the MME derives three keys from K_{ASME}. The two transient keys, denoted as K_{NASenc} and K_{NASint}, are used for encryption and integrity checks, respectively, of signaling traffic in the NAS. The third key, denoted as K_{eNB}, is the second intermediate key and is specific for an eNodeB and a UE. After being transferred to eNodeB, K_{eNB} is used to derive another three transient keys (see Fig. 2). Among these three keys, two are used to encrypt and check the integrity of Radio Resource Control (RRC) signaling traffic in the AS (i.e., K_{RRCenc} and K_{RRCint}). The last key is used to encrypt U-plane data traffic in the AS (i.e., K_{UPenc}). The UE should be able to derive from the permanent master key the two intermediate keys, the two transient keys for the NAS, and the three transient keys for the AS.

The key used for the AS protection keys (i.e., K_{eNB}) requires updating whenever a UE serves a different eNodeB as a result of an inter-eNodeB handover. The EPS security uses only a single set of K_{ASME} and defines the handover key update without involving an MME. MME involvement at every inter-eNodeB handover levies excessive computational and signaling loads and causes communication delays in the EPC. To avoid these MME problems, the EPS permits the K_{eNB} update to occur directly between eNodeBs.

2.3 Key Management in the Handover

The EPS supports two types of handovers that are referred to as intra- and inter-MME handovers, with the names reflecting the anchor points involved. In the intra-MME handover, preparation for it occurs between the source and target eNodeBs in the same MME through a direct interface between base stations (see *X2* interface in Fig. 1). In contrast, in the inter-MME handover, the preparation occurs via the MME without any direct signaling between base stations. As an alternative to the inter-MME handover, the UE and the MME may decide to run the full EPS-AKA to generate all security contexts from scratch. This alternative is more common in the inter-MME handover for security reasons. If different providers operate the two MMEs, the link between them is far from secure [37]. In this paper, we only consider the intra-MME handover in discussing the security weakness of key management in the handover because any security risks related to the inter-MME handover can be eliminated by running the full EPS-AKA.

For efficiency, source eNodeB provides the next K_{eNB} (K^*_{eNB}) to the target network for use after the handover. Before the next EPS-AKA, a set of K_{eNB} are linked to each other in what is known as *handover key chaining* [1]. To achieve backward key separation, source eNodeB generates the next K_{eNB} from the current one by applying a one-way hash. To ensure forward key separation, the source eNodeB must capitalize on fresh keying material from an MME. An MME can provide fresh keying material to the target eNodeB only after the inter-eNodeB handover, and this fresh material is to be used in the *next* handover. The result is two-hop forward key separation in which the source eNodeB does not know the target eNodeB key only after two inter-eNodeB handovers. Handover key chaining includes two additional parameters as fresh keying material; these two are the Next Hop (NH) key and the NH Chaining Counter (NCC). An MME recursively generates a new NH key derived from K_{ASME} for each handover. NCC is a counter value for the NH key.

Fig. 3 illustrates the message flow of the inter-eNodeB handover. We assume that the source eNodeB has fresh keying material, $\{NH_{NCC}, NCC\}$, from the *previous* handover (see message (0) in Fig. 3). NH_{NCC} denotes that the NH key is updated NCC times. Assume the current security association between a UE and the source eNodeB is based on K_{eNB}. The handover key chaining provides two key derivation mechanisms for a source eNodeB. The source eNodeB computes the new K_{eNB} (K^*_{eNB}) value for the target eNodeB from either the currently active K_{eNB} or from the NH key received from an MME on the *previous* handover, respectively, in the horizontal and vertical key derivations. Equations (1) and (2) represent the horizontal and vertical key derivations, respectively. The Key Derivation Function (KDF) refers to generic keyed one-way hash functions:

$$K^*_{eNB} = KDF(K_{eNB}, \alpha), \tag{1}$$

$$\begin{aligned} &K^*_{eNB} = KDF(NH_{NCC}, \alpha),\\ &\text{where } NH_{NCC} = KDF(K_{ASME}, NH_{NCC-1}), \end{aligned} \tag{2}$$

where α represents the cell-level values such as the target cell's physical cell identity and frequency. The initial value of the NH key (NH_0) is computed as $KDF(K_{ASME}, K_{eNB})$.

The horizontal handover is for cases in which the source eNodeB does not have a fresh NH key available. Such instances occur after a UE enters an MME's territory for the first time. They also happen when the $\{NH, NCC\}$ pair does not arrive in time before the occurrence of a new inter-eNodeB handover. The vertical handover denoted in (2) is more common. The source eNodeB should have a fresh NH key (i.e., NH_{NCC} in Fig. 3) that was from an MME in the *previous* inter-eNodeB handover. The NH_{NCC} is derived from the previous NH value (NH_{NCC-1}) and K_{ASME} (see (2)); thus, only an MME and a UE can derive a NH key. A compromised eNodeB cannot compromise any future K_{eNB} because an MME at a higher security anchor point is involved. Accordingly, the horizontal key derivation provides only backward key separation but the vertical key derivation presents both backward and, with a well-defined limitation (i.e., two-hop), forward key separation.

The source eNodeB forwards the $\{K^*_{eNB}$, $NCC\}$ pair to the target eNodeB (see message (2) in Fig. 3). In this figure, we assume that the source eNodeB executes the vertical handover key derivation. The subsequent session keys between a UE and the target eNodeB are derived directly from K^*_{eNB}.[1] The target eNodeB sends the NCC to a UE (see message (3) in Fig. 3). The UE compares the received NCC with the NCC value associated with the current security association (i.e., $NCC - 1$). If they are the same, the UE uses (1) to derive the K^*_{eNB} from the currently active K_{eNB}. If the received NCC is greater than the current NCC, the UE will first synchronize these two NCC values by computing the NH key and the NCC value iteratively until the two NCC values match and then use (2) to derive the K^*_{eNB}. When the target eNodeB has completed the handover signaling with the UE, it sends the *S1 path switch request* message (message (5) in Fig. 3) to an MME. The MME increases the NCC value by one, then computes a new NH (i.e., NH_{NCC+1}) from the K_{ASME} and current NH key. The MME forwards the fresh $\{NH_{NCC+1}, NCC + 1\}$ pair to the target eNodeB for use in the *next* handover.

[1] In fact, a targeted eNodeB renews $K_{eNB}{}^*$ by hashing the received $K_{eNB}{}^*$, and a cell-level temporary identifier. However, the cell-level temporary identifier is sent in plain text on a link layer from the target eNodeB to a source eNodeB.

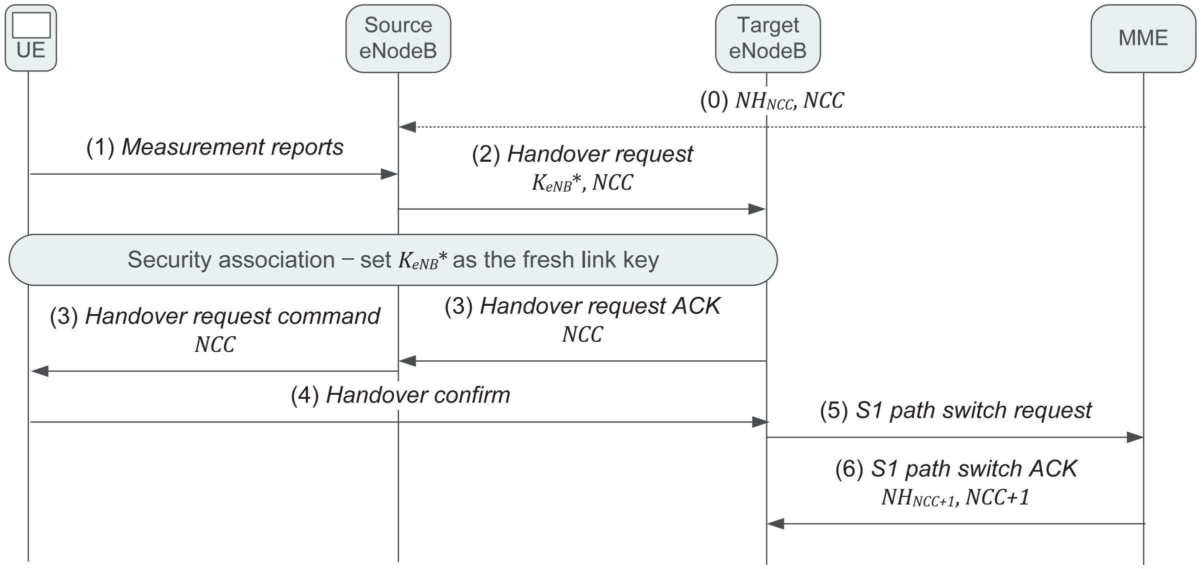

UE
Source eNodeB
Target eNodeB
MME
(0) NH_{NCC}, NCC
(1) *Measurement reports*
(2) *Handover request*
K_{eNB}*, NCC
Security association – set K_{eNB}* as the fresh link key
(3) *Handover request command*
NCC
(3) *Handover request ACK*
NCC
(4) *Handover confirm*
(5) *S1 path switch request*
(6) *S1 path switch ACK*
NH_{NCC+1}, $NCC+1$

Fig. 3. Message flow of the inter-eNodeB handover in the EPS.

3 Security Analysis of Inter-eNodeB Handover

As noted in Section 2, handover key chaining is protected by backward and forward key separation. In this section, however, we probe the security vulnerabilities of the inter-eNodeB handover by modeling a rogue base station attack.

3.1 Attack Preparations

A rogue base station (i.e., eNodeB) is a mobile device that duplicates the functionality of a base station. It can impersonate a legitimate base station and entice subscribers to camp on the radio channel of the rogue base station. An adversary can control a rogue base station either by compromising a commercial eNodeB or by deploying a personal eNodeB. The following discussion explores how an adversary can compromise a commercial eNodeB and deploy a personal eNodeB.

A commercial eNodeB can be exploited and compromised through physical, host, and network protocol vulnerabilities. By physically penetrating an eNodeB, an adversary can access its stored cryptographic materials. This physical vulnerability is theoretically possible [28] because eNodeBs in the LTE architecture are placed in locations that include public indoor sites. Because eNodeBs are Internet endpoints, an adversary also can gain access to the operating systems of eNodeBs by disseminating viruses and worms and commandeer eNodeBs as members of a botnet [30]. Furthermore, a commercial eNodeB can be compromised by vulnerabilities because of the IP stack such as identity forgery, eavesdropping, packet injection, packet modification, denial-of-service (DoS) attacks, and so on. An attacker can masquerade as legitimate eNodeBs by stealing identities and using them to send messages [31]. Meanwhile, self-motivated users can deploy a personal eNodeB by purchasing small, low-cost eNodeBs available at commercial network providers (e.g., Sprint Airave, AT&T 3G microcell). Otherwise, they can use a commercial software library [32] to manufacture rogue eNodeBs.

3.2 Desynchronization Attacks

Execution of a desynchronization attack requires that an adversary control a rogue base station either by compromising a commercial eNodeB or deploying a personal eNodeB. At the outset, an adversary entices a UE to camp on the radio channels of the rogue eNodeB. The goal of this rogue eNodeB attack is to disrupt updating of the NCC value, leaving the targeted eNodeBs desynchronized and future session keys vulnerable to compromise. In turn, the rogue eNodeB attack allows an adversary to force the targeted eNodeBs to abandon forward key separation by performing only horizontal handover key derivation. The refreshing of the NCC value, essential to the forward key separation of handover key chaining, can be disrupted by either manipulating the message between eNodeBs (see message (2) in Fig. 3) or the message from an MME to a targeted eNodeB (see message (6) in Fig. 3).

To desynchronize the NCC value in a targeted eNodeB, the rogue eNodeB purposely sets an extremely high value for the NCC value denoted as ψ and sends it to the targeted eNodeB in the *handover request* message in (2) of Fig. 3. This extremely high ψ value ranges near the highest value permitted for an NCC value (i.e., 8 bits). Even a naïve adversary without a rogue eNodeB can manipulate the *handover request* message in (2) of Fig. 3 if the IPSec association between eNodeBs is not adopted. An adversary sends to a UE the original NCC value denoted as ω and, by synchronizing the false NCC value (i.e., ψ), orders it not to perform vertical key derivation. The NCC value from the *S1 path switch acknowledgement* (*ACK*) message is considerably smaller than that received from the rogue eNodeB (i.e., $\omega + 1 \ll \psi$). In turn, this size difference causes the targeted eNodeB and the UE to generate the next session key based on the current K_{eNB} instead of on the new $NH_{\omega+1}$ key. In such an instance, the compromised eNodeB possesses the further K_{eNB} because the forward key separation of K_{eNB} has been lost. The eNodeB acquiring this K_{eNB} can now know the future K^*_{eNB}s because the α value can be exposed through the physical layer information [25]. After an initial desynchronization attempt, an adversary has to keep deceiving the UE into sending an original NCC value (i.e., ω) while continuing to track the UE for further active attacks.

An adversary can also desynchronize the NCC value by manipulating the *S1 path switch ACK* message in (6) of Fig. 3. Note that the EPS architecture inherits most of the IP-specific security vulnerabilities [6], [7]. An eNodeB compromised by the IP vulnerabilities would be in a position to launch IP spoofing and man-in-the-middle attacks onto the S1-C interface to modify the NCC update message from an MME to the targeted eNodeB. A forged message that includes a lower NCC value than a current NCC value would cause the targeted eNodeB not to acknowledge the fresh NCC value. The use of IPSec for the *S1 path switch request* and *ACK* message can be a good mechanism to protect against this attack. However, the IPSec for a *S1 path switch request* and *ACK* message is not mandatory for performance reasons because an MME needs to establish a number of IPSec associations with eNodeBs. On the other hand, however, an attacker has only to launch a DoS attack (e.g., packet dropping or packet flooding) on the S1-C interface to prevent a targeted eNodeB from receiving the updated NCC values in the *S1 path switch ACK* message.

Desynchronization attacks force a targeted eNodeB to fail to refresh the NCC value, and leave the UE capable of performing only horizontal key derivation. Once this breaks the security of forward key separation, an attacker with a rogue eNodeB can decipher messages between the genuine eNodeBs and a UE, including RRC signaling and U-plane information. In turn, a compromised K_{eNB} would then be used for further active attacks such as initiating call spoofing, promulgating voice spam, committing billing fraud, and degrading quality of service.

3.3 The Significance of Root Key Update

The effect of compromising a key by a desynchronization attack lasts until K_{ASME} is revoked through the EPS-AKA procedure required between an MME and a UE; in this procedure, the new K_{eNB} and subsequent security contexts are created from scratch. Some argue that MME involvement at every inter-eNodeB handover can prevent desynchronization attacks through increased handover delays and signaling overhead in the core network. Otherwise, enhanced message authentication between eNodeBs can detect the desynchronization attack. In turn, the UE can detect any changes to the NCC value in message (3), if the NCC value is associated with it, to derive the K^*_{eNB} key in (1) and (2). After detecting the desynchronized status, however, a correction mechanism (e.g., resynchronization) should be necessary.

Outside of specific preventive solutions, we emphasize the significance of a root key update to minimize the effect of a desynchronization attack and another key compromise by unknown attacks. In general, a cryptographic key has a specific defined lifetime so as to limit the risk of key exposure and compromise. It is an intrinsic procedure to continually refresh a cryptographic key beyond its lifetime limit to diminish the risk of its exposure and compromise. Because finding an optimal lifetime for a cryptographic key is challenging, determination of this key update interval has been explored in numerous papers [33], [34]. Hence, we believe that our approach is contextually valid as one of the options available to minimize the effect of key compromise by known and unknown attacks.

Our preference, however, leaves us with the problem of determining the appropriate interval for updating a root key. A short update interval requires frequent authentication procedures that lead to higher signaling traffic in the core network. Conversely, a long interval exposes users to attacks, such as loss of confidentiality of the RRC signaling and U-plane data traffic. In the next section, we analyze the tradeoff between signaling load and data exposure that is involved in determining the update interval for a root key.

4 Analytic Model for Inter-eNodeB Handover

To analyze the effect of the root key update interval, consider the timing diagram for an inter-eNodeB handover in terms of a root key update. Fig. 4 illustrates a timing diagram for one full MME residence time as determined by the time difference between entering and leaving an MME area ($t_R = \tau_4 - \tau_1$). The six unshaded arrows on top indicate the times when the intra-MME handover occurs. In particular, Fig. 4 shows two notable incidents, each marked by a shaded arrow; the left arrow is the launch of a desynchronization attack at τ_2, and the right arrow is the expiration of a root key update interval at τ_3. The effect of a desynchronization attack can last until the next update of the root key; that is, when a UE either moves to a new MME at τ_4 or requests a manual key update at τ_3. Such a move or request triggers full EPS-AKA between an MME and a UE; as a result, the new K_{ASME} is agreed upon and a new K_{eNB} is derived from the fresh K_{ASME}.

Let $t_U (= \tau_3 - \tau_1)$, and $t_R (= \tau_4 - \tau_1)$ in Fig. 4 denote, respectively, the interarrival time of the key update and the MME residence time. $t_u (= \tau_3 - \tau_2)$ and $t_r (= \tau_4 - \tau_2)$ are the residual time (i.e., the period from the time of the desynchronization attack to each termination event), respectively, of the key update and the MME residence time. As shown in Fig. 4, residual time can be expressed as *key exposure time* because the successive K_{eNB}s in the handover key chaining are compromised after τ_2. $f_U(t)$, $f_R(t)$, $f_u(t)$, and $f_r(t)$ represent the probability density functions (PDF) of t_U, t_R, t_u, and t_r with the corresponding Laplace transforms $f^*_U(s)$, $f^*_R(s)$, $f^*_u(s)$, and $f^*_r(s)$, respectively.

We define the *vulnerable period* as the time difference between a desynchronization attack on an eNodeB and the time of updating of the root key by either a manual update or a departure from an MME. Once a desynchronization attack is launched, an adversary can compromise the future session keys until the root key update. The vulnerable period t_c is given by $min\{t_u, t_r\}$, where $0 \leq t_u < t_R$ and $0 \leq t_r < t_R$. t_c is just calculated as t_r when either $\tau_3 \leq \tau_2$ or $\tau_4 \leq \tau_3$ because t_u has a negative value. The vulnerable period can be zero when there is no desynchronization attack. Because the key update and MME residence are

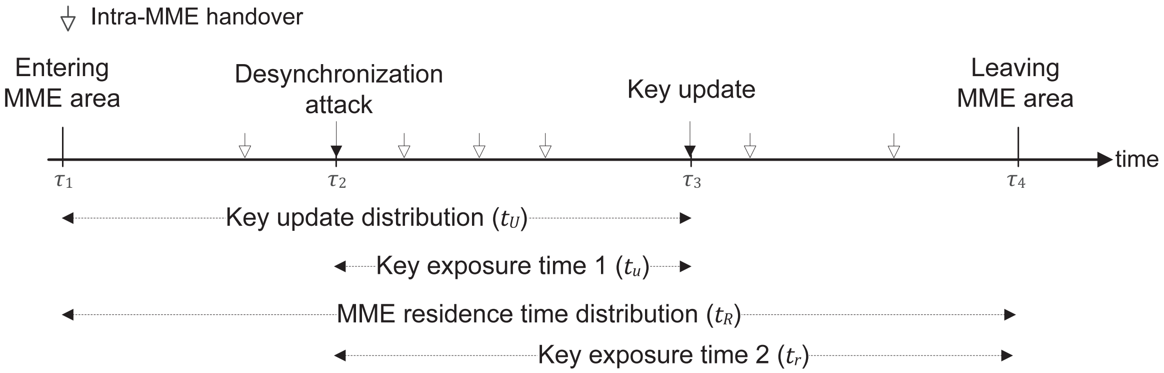

Intra-MME handover
Entering MME area
Desynchronization attack
Key update
Leaving MME area
time
τ_1
τ_2
τ_3
τ_4
Key update distribution (t_U)
Key exposure time 1 (t_u)
MME residence time distribution (t_R)
Key exposure time 2 (t_r)

Fig. 4. Timing diagram of vulnerable period regarding MME residence time and key update time.[2]

independent events, the Cumulative Distribution Function (CDF) of t_c can be expressed as[2]

$$\begin{aligned} F_c(t) &= \Pr\{min(t_r, t_u) \le t\} \\ &= \Pr(t_r \le t) + \Pr(t_u \le t) - \Pr(t_r \le t) \cdot \Pr(t_u \le t). \end{aligned} \tag{3}$$

Differentiating both sides of (3), we obtain the PDF of t_c, $f_c(t)$:

$$\begin{aligned} f_c(t) &= f_r(t) + f_u(t) - f_r(t) \cdot Pr(t_u \le t) - Pr(t_r \le t) \cdot f_u(t) \\ &= f_r(t) \cdot \int_t^{\mathrm{t_r}} f_u(\tau) d\tau + f_u(t) \cdot \int_t^{\mathrm{t_r}} f_r(\tau) d\tau. \end{aligned} \tag{4}$$

We can calculate the Laplace transforms of $f_c(t)$ (i.e., $f_c^*(s)$) by applying Laplace transforms to both sides of (4):

$$\begin{aligned} f_c^*(s) &= \int_0^\infty f_r(t) \cdot \left[\int_t^{\mathrm{t_R}} f_u(\tau) d\tau \right] e^{-st} dt \\ &\quad + \int_0^\infty f_u(t) \cdot \left[\int_t^{\mathrm{t_R}} f_r(\tau) d\tau \right] \cdot e^{-st} dt \\ &= \int_0^\infty f_r(t) \cdot e^{-st} dt - \int_0^\infty f_r(t) \cdot \int_0^t f_u(\tau) d\tau \cdot e^{-st} dt \\ &\quad + \int_0^\infty f_u(t) \cdot e^{-st} dt - \int_0^\infty f_u(t) \cdot \int_0^t f_r(\tau) d\tau \cdot e^{-st} dt \\ &= f_r^*(s) + f_u^*(s) - \int_0^\infty e^{-st} \cdot f_r(t) \cdot \int_0^t f_u(\tau) d\tau \cdot dt \\ &\quad - \int_0^\infty e^{-st} \cdot f_u(t) \cdot \int_0^t f_r(\tau) d\tau \cdot dt. \end{aligned} \tag{5}$$

According to the paradox of residual life [18], the residual time distribution of an original distribution is not equivalent to the original distribution. The residual time, γ_t, is defined as the time from t to the next arrival if t is an arbitrary point in the original renewal process, $\mathcal{R}_t$. The PDF of residual time in the Laplace form, $f_\gamma^*(s)$, is calculated by the residual life theorem [18] as shown in (6), where $f_\mathcal{R}^*(s)$, and $E(\mathcal{R}_t)$ represent the Laplace transform of PDF and the expectation value of the original renewal process $\mathcal{R}_t$:

$$f_\gamma^*(s) = \frac{1 - f_\mathcal{R}^*(s)}{s \cdot E(\mathcal{R}_t)}. \tag{6}$$

We assume that the distribution of the key update interval follows an exponential distribution with the mean value of T_U. The PDF of the key update interval and the Laplace transform are in (7), where $\mu_u = 1/T_U$:

$$f_U(t) = \mu_u \cdot \mathrm{e}^{-\mu_u t}, f_U^*(s) = \frac{\mu_u}{s + \mu_u}. \tag{7}$$

According to (6), the Laplace transform of $f_u(t)$ is calculated as follows:

$$f_u^*(s) = \frac{1 - f_U^*(s)}{s \cdot \int_0^\infty t \cdot f_U(t) \cdot dt} = \frac{\mu_u}{s + \mu_u}. \tag{8}$$

Through its Laplace transform, we can deduce that the PDF of the residual time of the key update, $f_u(t)$, would follow the exponential distribution with the mean value $1/\mu_u$. We expand $f_c^*(s)$ in (5) as shown in

$$\begin{aligned} f_c^*(s) &= f_r^*(s) + f_u^*(s) - \int_0^\infty e^{-st} \cdot f_r(t) \cdot \int_0^t f_u(\tau) d\tau \cdot dt \\ &\quad - \int_0^\infty e^{-st} \cdot f_u(t) \cdot \int_0^t f_r(\tau) d\tau \cdot dt \\ &= f_r^*(s) + f_u^*(s) - \int_0^\infty e^{-st} \cdot f_r(t) \cdot \left(1 - \mathrm{e}^{-\mu_u \cdot t}\right) dt \\ &\quad - \mu_u \cdot \int_0^\infty e^{-(s+\mu_u)t} \cdot \int_0^t f_r(\tau) d\tau \cdot dt \\ &= \frac{\mu_u}{s + \mu_u} + \frac{s}{s + \mu_u} \cdot f_r^*(s + \mu_u). \end{aligned} \tag{9}$$

We assume the distribution of the MME residence time follows a gamma distribution because most nonnegative random variables are a special case of gamma distribution. The PDF of the MME residence time with mean k/μ_r and variance k/μ_r^2 is shown in (10). The Laplace transform of $f_R(t)$ is also shown in

$$\begin{aligned} f_R(t) &= \frac{\mu_r^k \cdot t^{k-1} \cdot e^{-\mu_r \cdot t}}{\Gamma(k)}, \\ \text{where} \quad \Gamma(k) &= \int_0^\infty x^{k-1} \cdot e^{-x} dx, f_R^*(s) = \left(\frac{\mu_r}{s + \mu_r}\right)^k. \end{aligned} \tag{10}$$

According to (6), the Laplace transform of $f_r(t)$ is given as shown in

[2] We do not limit the number of key updates in an MME residence time to one. By definition of a vulnerable period, only the first key update right after the attack is modeled in the analysis of our study. On the contrary, we only consider a single desynchronization attack in our model because the second attack while the first attack is still valid does not generate any further consequences.

$$f_r^*(s) = \frac{1-f_R^*(s)}{s\cdot\int_0^\infty t\cdot f_R(t)\cdot dt} = \frac{\mu_r}{s\cdot k}\left\{1-\left(\frac{\mu_r}{s+\mu_r}\right)^k\right\}. \tag{11}$$

Thus, we can complete (9) with the MME residence process as shown in

$$\begin{aligned} f_c^*(s) &= \frac{\mu_u}{\mathrm{s}+\mu_u} + \frac{s}{\mathrm{s}+\mu_u}\cdot f_r^*(s+\mu_u) \\ &= \frac{\mu_u}{\mathrm{s}+\mu_u} + \frac{s}{\mathrm{s}+\mu_u}\cdot\frac{\mu_r}{(s+\mu_u)\cdot k}\left\{1-\left(\frac{\mu_r}{s+\mu_u+\mu_r}\right)^k\right\}. \end{aligned} \tag{12}$$

During the vulnerable period, U-plane data and RRC signaling traffic in the AS packets are subject to eavesdropping. The expected volume of exposed packets during the vulnerable period, $E[N]$, is defined as follows, where λ_p and $h^{(i)}(x)$ are the mean arrival rate of AS packets and the ith derivative of function $h(x)$ at point x:

$$\begin{aligned} E[N] &= \lambda_p\cdot -f_c^{*(1)}(0) \\ &= -\lambda_p\cdot\frac{d}{ds}\left\{\frac{\mu_u}{\mathrm{s}+\mu_u}+\frac{s}{\mathrm{s}+\mu_u}\cdot f_r^*\left(s+\mu_u\right)\right\}\bigg|_{s=0} \\ &= \frac{\lambda_p}{\mu_u}\cdot\left\{1-\frac{\mu_r}{\mu_u\cdot k}\left\{1-\left(\frac{\mu_r}{\mu_u+\mu_r}\right)^k\right\}\right\}. \end{aligned} \tag{13}$$

The distribution of interarrival time between key renewals, $f_l(t)$, is the convolution of $f_U(t)$ and $f_R(t)$. The Laplace transform of $f_l(t)$ can be calculated as in (14). The expected value of the signaling overhead rate, $E[S]$, is shown in (15), where ρ denotes the number of bits in messages for individual authentication among the UE, the MME, and the HSS/AuC:

$$f_l^*(s) = f_U^*(s)\cdot f_R^*(s) = \frac{\mu_u}{s+\mu_u}\cdot\left(\frac{\mu_r}{s+\mu_r}\right)^k, \tag{14}$$

$$E[S] = \frac{\rho}{-f_l^{*(1)}(0)} = \rho\bigg/\left(\frac{1}{\mu_u}+\frac{k}{\mu_r}\right). \tag{15}$$

Obviously, as $T_U(=1/\mu_u)$ increases, $E[N]$ increases whereas $E[S]$ is reduced. Having analyzed and measured $E[N]$ and $E[S]$, we need to validate our analytical model.

5 Simulation Results

5.1 Simulation Setting and Model Validation

We used the EURANE module [20] and a LTE queue development package [21] in the ns-2 simulator [19] to implement the EPS security framework—which includes EPS-AKA, the inter-eNodeB handover described in Section 2. For the KDF operation, we manually added the processing delay that is part of the EPS-AKA by using Hash-based Message Authentication Code (HMAC) with the Secure Hash Algorithm (SHA)-256 as measured by a PolarSSL [22] on an Intel Pentium IV 3.0 GHz with 1 GB of random-access memory. The average operation speed and standard deviation for HMAC-SHA-256 are 16.635 and 0.081 microseconds, respectively. A source eNodeB and the MME require one HMAC-SHA-256 operation each to calculate a new K_{eNB} and an NH value, respectively. The UE needs to synchronize NCC values by performing HMAC-SHA-256 operations equal to the number of horizontal handovers and computes the current NH value once. The length of all key materials is defined as 128 bits except that K_{eNB} and NH are 256 bits [1]. We enumerate the number of bytes in the EPS-AKA, including the security transaction (i.e., Security Mode Command (SMC) operation [1]), which is 384 bytes ($\rho = 384$ bytes). We used File Transfer Protocol (FTP) traffic that has a rate of 64 Kbps ($\lambda_p = 64$ Kbps) as background data traffic for a UE. In the simulation, the FTP session arrival is generated by the exponential distribution. We ran the simulation until we could obtain a sufficient number of renewal intervals (i.e., $\tau_4-\tau_1$ in Fig. 4), which we determined to be 500 intervals, to speculate on the characteristics of a UE without outliers. The simulation time depends on the MME residence time. We excluded the first 10 runs to remove unrelated initialization bias.

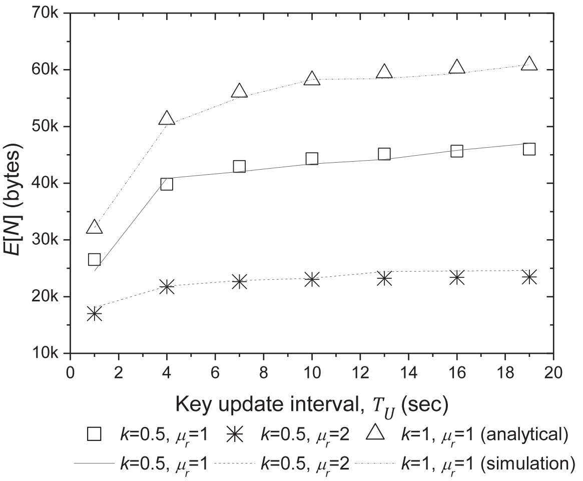


Fig. 5. The expected volume of exposed packets over the key update interval.

As part of the validation of our analytical model, we restricted a UE's mobility so that it follows a given gamma distribution by using the random generator in a gamma distribution built in ns-2. The two parameters that define a gamma distribution are the shape (k) and the mobility rate (μ_r). We varied the k value and the μ_r value, respectively, from 0.5 to 1 and from 1 to 2, to understand the effect on MME residence time.

Figs. 5 and 6 show the expected volume of exposed packets ($E[N]$) and the expected signaling overhead ($E[S]$), respectively, against different key update intervals. As T_U increases, $E[N]$ increases, and $E[S]$ decreases, as shown in Figs. 5 and 6, respectively. As the mobility rate (μ_r) increases, $E[N]$ decreases and $E[S]$ increases because high mobility implies frequent changes of the MME areas and, hence, frequent performing of the EPS-AKA because of the inter-MME handover. As the shape value (k) increases, $E[N]$ increases and $E[S]$ decreases because the average MME residence time increases. Note that the average MME residence time is calculated as k/μ_r. Figs. 5 and 6 contain results from the analytical model and the simulation. The two results have good agreement over a range of parameters.

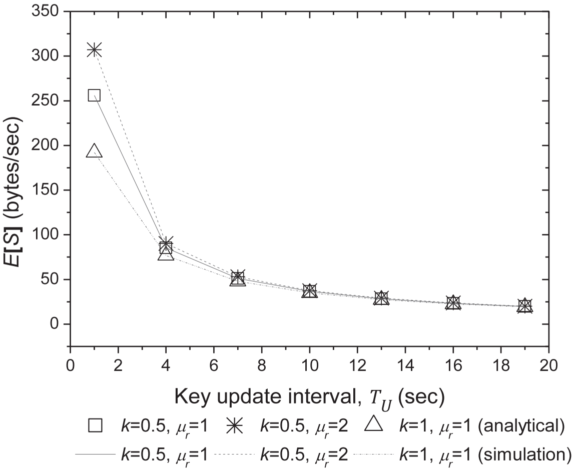


Fig. 6. The expected signaling overhead over the key update interval.

5.2 Investigation of MME Residence Time

Although the previous simulation validates our model, it is important to use a realistic mobility model to ensure that the simulation results accurately reflect the real-world performance of mobile networks. We extended our simulation based on the following empirical scenarios so as to explore the effect of realistic environments on MME residence time. The MME residence time can be affected by the following three factors:

- *Inter-eNodeB distance* (d_{eNB})—The distance between eNodeBs determines the size of an MME area. Under the same constraints of user movement, the area of an MME region determines the residence time.[3]
- *UE velocity* (v_{UE})—MME residence time can be directly affected by the UE velocity provided by an identical number of eNodeBs. Intuitively, a fast-moving UE would have less residence time than a slower one.
- *Road characteristics* (c_{ROAD})—Road characteristics determine MME residence time by controlling a UE's movement. To represent road characteristics numerically, we adapted a concept of a clustering coefficient [24] in a graph theory. The clustering coefficient ranges from zero to one and its value is determined to an extent by how close its neighbors are to being a complete graph. We found that the more intersections in road characteristics, the greater the value of the clustering coefficient.

We took advantage of map data, called Topologically Integrated Geographic Encoding and Referencing (TIGER) [23]. This map contains detailed street information in the United States. We used another tool called MOVE [27] to generate realistic movement of a UE on the TIGER map. We truncated each map to a size of $10\ \mathrm{km} \times 10\ \mathrm{km}$, and placed a UE on the map to investigate the characteristics of mobility. The UE moves from a randomly chosen starting point to a destination point on a given map. We assumed that the UE does not go out of eNodeB communication range. We selected three counties to represent three realistic mobility models of urban, suburban, and rural movement. These counties are New York in New York State, DeKalb in Georgia, and Chautauqua in Kansas.

The clustering coefficient (c_{ROAD}) of New York is the greatest because of a lot of intersections in Manhattan. We expected that the UE's MME resident time for New York would be the greatest because it is probable that the UE's mobility pattern is circular and repetitive and, thus, the UE resides longer within the control of a single MME. This expectation was verified by our simple test, and we concluded that the greater the clustering coefficient, the longer the MME residence time.

5.3 The Effect of a Key Update Interval on the Vulnerable Period

Three figures in Fig. 7 depict the vulnerable period (t_c) versus the key update interval in terms of d_{eNB}, v_{UE}, and c_{ROAD}, respectively.

Note that the vulnerable period corresponds to the minimum of two parameters: MME residence time and the key update interval (see (3)). If the MME residence time is fixed, as the key update interval value increases, vulnerability is greatest at the point when the key update interval value equals the MME residence time. Thus, any value for the key update that is greater than the point at which this maximum vulnerability occurs is unnecessarily long, even when a less frequent key update would serve to reduce signaling overhead.

In Fig. 7c, we confined our attention to a fixed v_{UE} (15 mps) and d_{eNB} (300 m). We did not focus on the relationship among performance criteria. However, in reality, a UE in a rural area might move faster than one in an urban area because of less congestion and higher speed limits. Besides, the inter-eNodeB distance in an urban area may be shorter than in a rural area because of man-made obstacles in an urban area that would interfere with signal propagation. However, we checked to see how our assessment matched with real-world operations by manually inspecting the inter-eNodeB distance of Verizon. After visually inspecting the inter-eNodeB distance in New York, NY, and Chautauqua, KS, we concluded that the relationship of independent performance factors is too uncertain for precise definition. As a result, we place the responsibility on a network operator to arrive at an optimal interval for updating a root key. This determination should be based on the operator's examination of an individual UE.

6 Optimal Key Management

Selection of an appropriate root key update interval should be a high priority for network administrators. An unnecessarily frequent key update interval wastes signaling overhead. On the other hand, a lethargically infrequent key update interval may endanger an end user's security and privacy. With these parameters in mind, we are ready in this discussion to find an optimal operating point for a root key update interval that will minimize both the volume of exposed packets and the signaling traffic overhead. According to [38], the optimal value may lie on the balanced value between two decisive factors when they have inverse relationship. We define an optimal value as one with which, with a given range of T_U, network operators can operate their systems with a *balance* between signaling overhead and risk of security breaches; in other words, such

[3] For simplicity, we placed an identical number of eNodeBs in an MME area and deployed them in a rectangular manner.

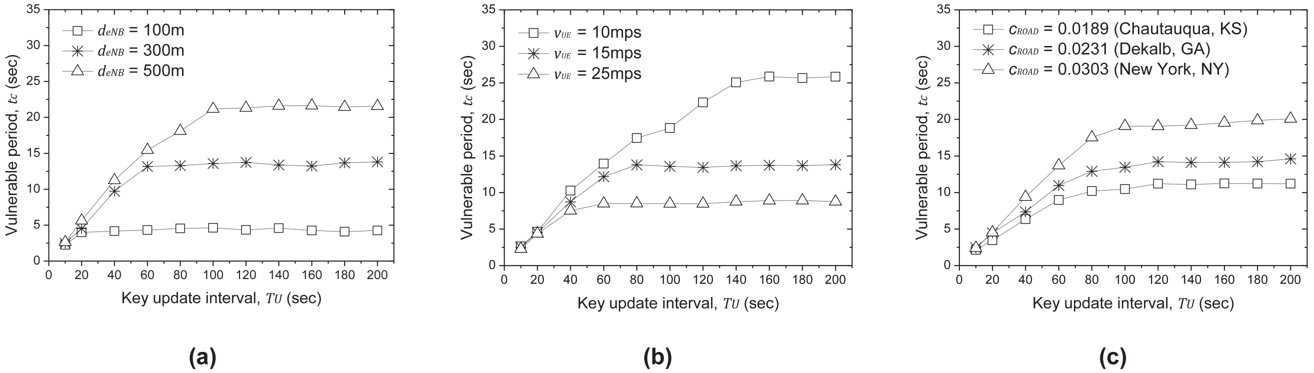


Fig. 7. The vulnerable period in terms of (a) inter-eNodeB distance (meters, m), (b) UE velocity (meters per second, mps), (c) road characteristics.

a value has a maximum T_U that brings $E[N]$ and $E[S]$ to their lowest possible values. However, in general, a globally accepted balanced value does not exist because such a value should be determined by a network operator and must take management policy into account. Thus, we want to provide network operators with an option to give different weight, in accordance with the management policies, to $E[N]$ and $E[S]$ in the course of determining a proper T_U value.

6.1 Algorithm for Selecting an Optimal Key Update Interval

We propose an algorithm (see Algorithm 1) to determine such an optimal T_U value. The inputs to Algorithm 1 are $\overline{E[N]}, \overline{E[S]}$ and relative importance δ. A relative importance $\delta (0 < \delta < \infty)$ is determined by a network operator's choice as the ratio of the signaling traffic overhead to the volume of exposed packets. $\overline{E[N]}$ and $\overline{E[S]}$ correspond to, respectively, the maximum values of $E[N]$ and $E[S]$ in the network. We assure that a system administrator can ascertain the $\overline{E[N]}$ and $\overline{E[S]}$ values by monitoring his or her network and adjusting them empirically. Note that the Operation and Maintenance Center (OMC) in the EPS network [1] provides real-time network monitoring of signaling traffic and data traffic for each UE. As an initial value, T_U is set to 1 second in line 1. For each T_U value, the $\overline{E[N]}$ and $\overline{E[S]}$ values are used, respectively, to normalize $E[N]$ and $E[S]$ in lines 3 and 4. $E[N]$ and $E[S]$ are calculated, respectively, according to (13) and (15). If the ratio of normalized $E[N]$ to normalized $E[S]$ $(= S/N)$ is greater than δ, T_U is increased by a step value (e.g., $\epsilon = 0.1$ second) and continues the loop. Otherwise, T_U is returned as the optimal key update interval.

Algorithm 1: Selecting an optimal key update interval.

```
 Input:  $\overline{E[N]}, \overline{E[S]}, \delta$
Output:  $T_U^+$
     1: Initialize $T_U$ as 1
     2: while $T_U < \infty$ do
     3:    $N = E[N]/\overline{E[N]}$
     4:    $S = E[S]/\overline{E[S]}$
     5:    if   $S/N \geq \delta$ then
     6:       $T_U = T_U + \epsilon$
     7:    else
     8:       return $T_U$
     9: end while
```

Fig. 8 shows a graphical plot of normalized $E[N]$ and $E[S]$ values in terms of T_U value in which each T_U value is marked in seconds. One point in the curve is drawn from a paired normalized $E[N]$ and $E[S]$ with one T_U value. Multiple points can be calculated by varying the T_U values. These points make up a convex curve as shown in Fig. 8. $E[N]$ and $E[S]$ have inverse relationship according to T_U because $E[N]$ and $E[S]$, respectively, increase and decrease when T_U is increased from top left to bottom right. The lower and upper limits of T_U value are closely located, respectively, at (0, 1) and (1, 0). Our proposed Algorithm 1 examines diverse update interval values to find the balance point according to the given δ value (see $\delta = 1$ and $\delta = 0.5$ in Fig. 8).

Fig. 8 demonstrates two curves with different MME residence times of 63.23 and 102.93 seconds. At the same key update interval, $E[N]$ increases as the MME residence time increases because the vulnerable period lasts longer (see Fig. 7). In addition, $E[S]$ decreases as the MME residence time increases because, as a result of the EPS-AKA, the key update procedure is performed infrequently if the periodical key update interval is fixed. According to the Algorithm 1, the junction value of the dotted (i.e., $\delta = 1$) and solid lines is considered to be an optimal operating value for T_U. Thus, the optimal point lies on the lower convex hull of the curve near (0, 0), which minimizes the $E[N]$ and $E[S]$ values. As shown in Fig. 8, the optimal update interval decreases when the δ value increases

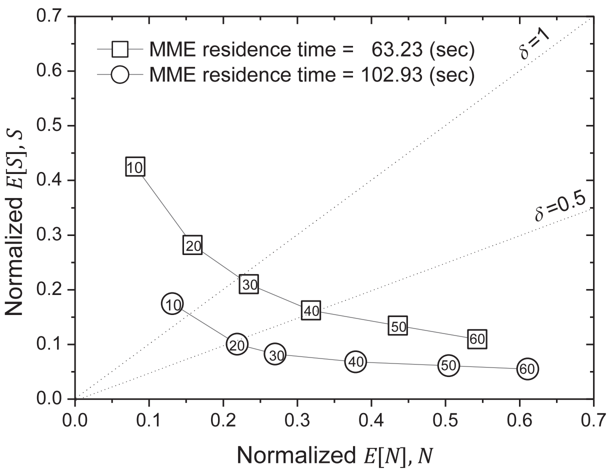


Fig. 8. Determination of the optimal key update interval for a given relative importance value.

because greater δ values imply that the network operator is concerned less with signaling overhead than with system security. Hence, the network operator requests more frequent key updates, thus lowering the optimal T_U value.

6.2 Parameter Estimation

To practice Algorithm 1, we need to calculate $E[N]$ and $E[S]$ by estimating the parameters in (13) and (15). MME residence time is continuously observed by the HSS/AuC from time stamps collected when the cancel location message [35] is exchanged between the old MME and the HSS/AuC. The cancel location message indicates that the handover from old MME to new MME is terminated. The parameters (μ_r and k) of distribution can be estimated by Bayesian minimum mean-squared error and maximum-likelihood methods. To calculate the mean arrival rate of application data packets, the Internet Protocol Detail Record (IPDR) [35] is adopted for detailed information on IP-based communication sessions such as types of services and quantities of services in kilobytes per time. The IPDRs are sent to the HSS/AuC via the billing gateway (i.e., Policy and Charging Control (PCC)) when the user's data session is established and finished. Then, the HSS/AuC can calculate the mean arrival rate of application data packets (λ_p) for a UE.

6.3 Implementation Issues

One might think that executing Algorithm 1 at every MME residence time might impose overhead on the core network. Although this *global* optimum can precisely find optimal key update interval, efficiency may be scarified. The following local optimum can help a network administrator to efficiently find a key update interval. Let $T_U^+(j)$ be the optimal update interval selected for the jth MME area. The local optimum of a key update interval for $(j+1)$th MME area should be adjusted to one of three options: those are $T_U^+(j) - \epsilon$ (decrement by a unit), $T_U^+(j)$ (no changes), and $T_U^+(j) + \epsilon$ (increment by a unit). The next optimal key update $T_U^+(j+1)$ will be selected as a interval that has corresponding S/N value closest to δ.

An implementation of Algorithm 1 on a per-UE basis might impose overhead on the core network in terms of the cost of subscriber management. Although a single value could be set across the network, setting different values for each UE would be ideal because each UE differs in its degree of security vulnerability and tolerance of signaling load. To reduce the management overhead in the core network, it would be useful to group UEs in a same serving eNodeB and assign the same δ value to a group of UEs. UEs in an eNodeB that share the same degree of vulnerability might require the same level of security. An eNodeB's vulnerability can be measured as a numerical value when a network administrator decides based on, for example, whether it is deployed in open environments such as public spaces and hot spots. When a UE moves into a new eNodeB's area, the HSS/AuC identifies the serving eNodeB's vulnerability, and then sets the relative importance value of the group of UEs. We ruled out exhaustive fragmentation that requires frequent group dynamics and increases the load of group management.

7 Literature Review

We surveyed the literature on security weaknesses of the UMTS and the EPS networks. We also reviewed the literature and methodology of mathematical analyses of the AKA in 3GPP mobile networks.

7.1 Security Analysis

The security weaknesses of the AKA in 3GPP mobile networks have been increasing the possibility of rogue base station (i.e., false base station) attacks [3], [28], [29]. Mitchell [29] first identified rogue base station attacks in the Global System for Mobile Communications (GSM); these attacks took the form of call stealing on unencrypted networks and call spoofing. Zhang et al. [3] pointed out that the UMTS security displays vulnerabilities to a variant of rogue base station attacks. To the best of our knowledge, no serious rogue base station attacks on the EPS architecture have been reported in the public literature. Only the 3GPP standard has discussed theoretical rogue base station attacks [28]. A few researchers initially surveyed EPS security. The authors in [4] and [5] provided a tutorial overview of EPS security, including the EPS-AKA and key management. The authors in [26] looked into handover key chaining and explored the operation of vertical and horizontal key derivation. The potential for DoS attacks on a specific UE by using radio signals was discussed in [25]. Recently, Køien [37] pointed out that the delegation from the authentication server requires strong trust assumptions, which seems outdated in the LTE heterogeneous networks. He presented a mutual authentication directly between the user and the authentication server in online.

7.2 Mathematical Analysis

An intuitive and simple approach was widely adapted to calculate the round-trip time of the UMTS-AKA [8]. The authors in [8] enumerated the number of handshakes among authentication entities in measuring the handover signaling. The authentication delay and total signaling load were calculated based on such statistical data as the velocity of a user, a registration area boundary, and the total number of users in the UMTS network [10], [11]. Lin et al. [9] have done pioneering work in expressing the timing diagram of the UMTS-AKA and in devising a probability model for authentication processes. They investigated the impact of the size of the authentication vector to minimize the signaling cost in the UMTS network. Lin's scheme has been expanded in terms of relaxing the Poisson assumption in the underlying authentication process [12], tuning the authentication vector management [13], and calculating the authentication processing delay [14]. Recently, the mathematical model for the AKA in 3GPP mobile networks has been expanded into integration with the mobility models [15], [16], [17]. Wu et al. [15] investigated the effect of the time-out period of the authentication vector on system performance by using a two-dimensional random walk. Zhang et al. [16] and Wang et al. [17] expanded the random walk model to a hexagonal grid map to study the effect of mobility on the evaluation of the traffic load in the UMTS-AKA and the EPS-AKA, respectively.

8 Conclusion

In this paper, we were concerned that forward key separation in handover key management in the 3GPP LTE/SAE network can be threatened because of what are known as rogue base station attacks. Although periodically updating the root key minimizes the effect of the attacks, selecting an optimal key update interval is an ill-defined problem because of the difficulty of achieving a balance between the signaling load and the volume of exposed packets. We have derived a mathematical framework for selecting an optimal handover key update interval that helps a network operator select an optimal value that fits best with network management policies.

Acknowledgments

H.-K. Choi was the corresponding author for this paper.

References

[1] "3GPP System Architecture Evolution (SAE); Security Architecture (Release 11)," 3GPP TS 33.401, Version 11.2.0, Dec. 2011.

[2] "3G Security, Security Architecture (Release 8)," 3GPP TS 33.102, Version 11.1.0, Dec. 2011.

[3] M. Zhang et al., "Security Analysis and Enhancements of 3GPP Authentication and Key Agreement Protocol," *IEEE Trans. Wireless Comm.*, vol. 4, no. 2, pp. 734-742, Mar. 2005.

[4] C.B. Sankaran, "Network Access Security in Next-Generation 3GPP Systems: A Tutorial," *IEEE Comm. Magazine*, vol. 47, no. 2, pp. 84-91, Feb. 2009.

[5] V. Niemi et al., "3GPP Security Hot Topics: LTE/SAE and Home (e)NB," *Proc. ETSI Security Workshop*, Jan. 2009.

[6] Y. Park et al., "A Survey of Security Threats on 4G Networks," *Proc. IEEE GlobeCom Workshop Security and Privacy in 4G Networks*, Nov. 2007.

[7] I. Bilogrevic et al., "Security and Privacy in Next Generation Mobile Networks: LTE and Femtocells," *Proc. Int'l Femtocell Workshop*, June 2010.

[8] Y.-B. Lin et al., "One-Pass GPRS and IMS Authentication Procedure for UMTS," *IEEE J. Selected Areas in Comm.*, vol. 23, no. 6, pp. 1233-1239, June 2005.

[9] Y.-B. Lin et al., "Reducing Authentication Signaling Traffic in Third-Generation Mobile Network," *IEEE Trans. Wireless Comm.*, vol. 2, no. 3, pp. 493-501, May 2003.

[10] H. Yangzhi et al., "An Improved Authentication Protocol with Less Delay for UMTS Mobile Networks," *Proc. IEEE Int'l Conf. Networking and Digital Soc. (ICNDS)*, May 2009.

[11] J.-A. Saraireh et al., "A New Authentication Protocol for UMTS Mobile Networks," *EURASIP J. Wireless Comm. and Networking*, vol. 2006, no. 2, p. 19, Apr. 2006.

[12] Y. Zhang, "Authentication Overhead in Wireless Networks," *Proc. IEEE Int'l Conf. Comm. (ICC)*, May 2008.

[13] J.-A. Saraireh et al., "Analytical Model for Authentication Transmission Overhead between Entities in Mobile Networks," *ELSEVIER Computer Comm.*, vol. 30, no. 8, pp. 1713-1720, June 2007.

[14] Y. Zhang et al., "An Improvement for Authentication Protocol in Third-Generation Wireless Networks," *IEEE Trans. Wireless Comm.*, vol. 5, no. 9, pp. 2348-2352, Sept. 2006.

[15] L.-Y. Wu et al., "Authentication Vector Management for UMTS," *IEEE Trans. Wireless Comm.*, vol. 6, no. 11, pp. 4101-4107, Nov. 2007.

[16] Y. Zhang et al., "A Study on Evaluating Authentication Traffics in the Next Generation Wireless Networks," *Proc. IEEE Int'l Conf. Comm. (ICC)*, June 2006.

[17] M. Wang et al., "Signaling Cost Evaluation of Mobility Management Schemes for Different Core Network Architectural Arrangements in 3G LTE/SAE," *Proc. IEEE Vehicular Technology Conf. (VTC)*, May 2008.

[18] L. Kleinrock, *Queueing Systems: Theory*, vol. 1, first ed. Wiley-Interscience, Jan. 1975.

[19] "The Network Simulator - ns-2," http://www.isi.edu/nsnam/ns, 2013.

[20] "Enhanced UMTS Radio Access Network Extensions (EURANE) for ns-2," http://eurane.ti-wmc.nl/, 2013.

[21] Q.-L. Qui et al., "LTE/SAE Model and Its Implementation in NS-2," *Proc. IEEE Fifth Int'l Conf. Mobile Ad-Hoc and Sensor Networks (MSN)*, Dec. 2009.

[22] "PolarSSL: Open Source Embedded SSL/TLS Cryptographic Library," http://polarssl.org/, 2013.

[23] U.S. Census Bureau, "TIGER, TIGER/Line and TIGER-Related Products," http://www.census.gov/geo/www/tiger/, 2013.

[24] M. Dehimer, "A Novel Method for Measuring the Structural Information Content of Networks," *Cybernetics and Systems*, vol. 39, no. 8, pp. 825-842, Nov. 2008.

[25] D. Forsberg et al., "Enhancing Security and Privacy in 3GPP E-UTRAN Radio Interface," *Proc. IEEE 18th Int'l Symp. Personal, Indoor and Mobile Radio Comm. (PIMRC)*, Sept. 2007.

[26] D. Forsberg, "LTE Key Management Analysis with Session Keys Context," *ELSEVIER Computer Comm.*, vol. 33, no. 16, pp. 1907-1915, Oct. 2010.

[27] F.K. Karnadi et al., "Rapid Generation of Realistic Mobility Models for VANET," *Proc. IEEE Wireless Comm. and Networking Conference (WCNC)*, Mar. 2007.

[28] "Rationale and Track of Security Decisions in Long Term Evolution (LTE) RAN/3GPP System Architecture Evolution (SAE) (Release 9)," 3GPP TS 33.821, Version 9.0.0, June 2009.

[29] C. Mitchell, "The Security of the GSM Air Interface Protocol," Technical Report RHUL-MA-2001-3, Aug. 2001.

[30] P. Traynor et al., "On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core," *Proc. 16th ACM Conf. Computer and Comm. Security (CCS)*, Nov. 2009.

[31] N. Chilamkurti et al., "Next-Generation Wireless Technologies: 4G and Beyond," Springer, June 2013.

[32] Nomor Research, "LTE Protocol Stack Library," http://www.nomor.de/home/solutions-and-products/products/lte-protocol-stack-library, 2013.

[33] S. Pack et al., "Optimal Binding Management Key Refresh Interval in Mobile IPv6 Networks," *IEEE Trans. Vehicular Technology*, vol. 58, no. 7, pp. 3834-3837, Sept. 2009.

[34] A.D. Gregorio, "Cryptographic Key Reliable Lifetimes: Bounding the Risk of Key Exposure in the Presence of Faults," *Proc. Third Int'l Conf. Fault Diagnosis and Tolerance in Cryptography*, Oct. 2006.

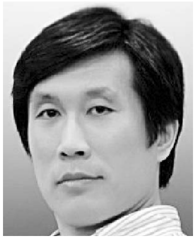
[35] "General Packet Radio Service (GPRS); Service Description; Stage 2," 3GPP TS 23.060, Version 10.5.0, Sept. 2011.

[36] G. Horn, *LTE Security*, first ed. Wiley-Interscience, Nov, 2010.

[37] G.M. Køien, "Mutual Entity Authentication for LTE," *Proc. IEEE Seventh Int'l Wireless Comm. and Mobile Computing Conf. (IWCMC)*, July 2011.

[38] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers," HP Laboratories, HPL-2003-4, http://www.hpl.hp.com/techreports/2003/HPL-2003-4.html, 2013.

Chan-Kyu Han received the PhD degree in mobile systems engineering from Sungkyunkwan University in 2012. She is a software engineer at Samsung Electronics. Her research interests include security and privacy in mobile communications.



Hyoung-Kee Choi received the PhD degree in electrical and computer engineering from the Georgia Institute of Technology in 2001. He is an associate professor and a director at the Education Center for Mobile Communications, Sungkyunkwan University, South Korea. He joined Lancope in 2001 and remained until 2004, where he guided and contributed to research in Internet security. His research interests include network security and Internet traffic modeling. He serves as an associate editor for the *ACM Transactions on Internet Technology*.