Securing Locations of Mobile Nodes in Wireless Mesh Networks

Sultan Alkhliwi
Lecturer at Faculty of Science,
Northern Border University, KSA

***Abstract*—The current deployment of wireless mesh networks requires mobility management to track the current locations of mobile nodes around the network without service interruption. To do so, the Hierarchical Mobile IPv6 protocol has been chosen, which minimises the required signalling by introducing a new entity called the mobile anchor point to act as a local home agent for all visiting mobile nodes in a specific domain. It allows a mobile node to register its local/regional care-of addresses with a mobile anchor point by sending a local binding update message. However, the local binding update is quite sensitive; it modifies the routing to enable mobility in the wireless mesh networks. When a local binding update message is spoofed, an attacker can redirect traffic that is destined for legitimate mobile node either to itself or to another node. This situation leads to an increased risk of attacks. Therefore, this paper contributes to addressing this security issue based on wireless mesh networks by cryptography generation and verification of a mobile node's local and regional care-of addresses, as well as the application of a novel method to verify the reachability of mobile node at claimed local care-of address. This is called the enhanced mobile anchor point registration protocol. The Scyther tool has been used to ensure the proposed protocol accuracy. Furthermore, the performance, in terms of the mobile anchor point registration delay and signalling overhead, is evaluated by using the OPNET modeller simulator.**

Keywords*—*Wireless mesh networks; hierarchical mobile IPv6 protocol; authentication; secret key; Scyther tool; OPNET simulation

I. INTRODUCTION

The number of wireless devices, such as laptops, PDAs, Bluetooth devices and so on, is substantially increasing all over the world. As a result, the demand for broadband wireless access is increasing rapidly. Currently, it seems that wireless mesh networks (WMNs) [1] will play a major role in future anywhere–anytime communications. The WMNs have gained significant attention from the research community, as well as the industry and standard organisations, due to their wireless access flexibility, combined with their high coverage area, excellent reliability and proven cost efficiency. The WMNs have a wide range of applications, such as in-home broadband services, enterprises, communities, metropolitan areas, intelligent transportation, industrial automation, sensors and emergency/rescue networks. The WMNs have received considerable interest as a promising way for reliable wireless broadband services to gain access with minimal upfront investments. The WMN features (i.e., dynamic, self-organised and self-healing) can be deployed incrementally, one node at a time, as needed. As more nodes are installed, both reliability and connectivity will increase, which all users will enjoy[1],[2].

A WMN comprises dedicated backbone wireless access routers (ARs) and gateway routers to offer last-mile broadband connectivity to users. Since roaming is related to the desire to access the internet from a WMN, an efficient mobility management protocol is required [1]. To facilitate WMN abilities to locate a mobile node (MN) point of attachment for delivering data packets and to maintain an MN's connection as it continues to change its point of attachment in the domain, the Engineering Task Force (IETF) proposed the Mobile IPv6 (MIPv6) protocol [3] to both permit this roaming connectivity and reduce the required signalling with the Hierarchical Mobile IPv6 (HMIPv6) protocol [4]. Therefore, the HMIPv6 protocol has been chosen to support mobility location management for WMNs. The HMIPv6 is a lightweight protocol (compared with others) for the following reasons: (1) It supports hierarchical tunnelling approaches that are flexible, modular and scalable for supporting IP-based macro and micro mobility for WMNs. (2) It introduces a new node agent called the mobile anchor point (MAP) to act as a local home agent for the MNs. Thus, when an MN changes its attached point in the same domain, it will update its location with the MAP instead of the home link, as defined in the MIPv6 protocol [3]. As a result, the HMIPv6 protocol requires minimal bandwidth and computational resources, as well as reduces registration delays when tracking the current location of the MNs. Finally, the HMIPv6 protocol enables the deployment of a group of ARs into different subnets for easy employment of WMNs.

In the HMIPv6 protocol, an MN has three IPv6 addresses, including a permanent home-of-address, which identifies the MN in its home link and remains the same during the MN's movement, and two transients: the regional care-of address (RCoA), which is generated based on the MAP option that is included in the router advertisement (RA) message, and the local care-of address (LCoA), which is generated based on an AR advertisement. When an MN moves into a MAP domain and configures a new LCoA and RCoA, it initiates a process to register both its LCoA and RCoA with the MAP by sending a local binding update (LBU). The MAP then replies to the MN with a binding acknowledgement (BA) message. This registration allows the MAP to create a binding for the MN between its LCoA and RCoA. The MAP uses this binding to intercept all packets that are destined for the MN's RCoA from its home link and/or correspondent nodes and forwards these packets to the MN's current location in the MAP domain by using the MN's LCoA.

However, the LBU is obviously quite sensitive; it modifies the routing to enable mobility in the WMNs. When an LBU message is spoofed, an attacker can redirect traffic either to itself or to another node, preventing the original MN from receiving any traffic that is destined for it. This situation leads to an increased risk of attacks (e.g., denial-of-service attack, man-in-the-middle attack). Thus, the greatest security vulnerabilities of the HMIPv6 protocol are both the authentication and the authorisation of an LBU message. Therefore, the use of appropriate security provisions for the MAP registration process is fundamental to the HMIPv6 protocol. It is believed that the deployment of an HMIPv6 protocol without securing the MAP registration process could result in a breakdown of the entire internet [5][6].

This paper presents a novel scheme, called the Enhanced Mobile Anchor Point (E-MAP) registration protocol, to support the location authentication of MNs in the MAP in the WMN domain and to authorise the MN to use the services of the WMN domain. By executing the E-MAP protocol, the MAP is able to verify the ownership of the claimed LCoA and RCoA and confirm not only the authenticity of the LCoA but that it is indeed an MN's real location. The E-MAP registration protocol also allows the MAP to securely identify and establish a shared secret key with the MN. As a result, the E-MAP protocol can reduce the likelihood that a malicious MN can successfully steal a third party's node addresses (i.e., LCoA and RCoA), prevent a malicious MN from launching a flooding attack and protect any future binding update (BU) that could be sent from the MN to the MAP against the false BU attack.

The rest of this paper is organised as follows. Section II provides an overview of the cryptographically generated address (CGA) technique, discusses the proposed protocols to secure the MAP registration process and related works of the reachability test mechanism. Section III presents the preliminaries behind the design of the E-MAP registration protocol. Section IV covers an overview of this study's novel protocol, including the idea behind the public key certificate CGA-based technique and the idea of simultaneously conducting the LCoA reachability test and generating a shared secret key between the MAP and the MN. Section V describes the E-MAP protocol in detail. Section VI presents a formal security analysis of the proposed protocol using the Scyther tool. Section VII evaluates the performance of the E-MAP protocol compared with both the basic MAP (B-MAP) registration protocol and the most related work in terms of the MAP registration delay and signalling overhead, using the OPNET modeller simulation. Section VIII concludes this paper and suggests future work.

II. Related Works

This section provides an overview of cryptographically generated IPv6 addresses, surveying the existing protocols, which are used to secure the MAP registration process in the HMIPv6 protocol, and related works of the reachability test mechanism.

A. Cryptographically Generated Address (CGA) Protocol

The CGA technique is used to prevent the stealing and spoofing of existing IPv6 addresses [7]**.** A CGA is an IPv6 address for which the interface identifier part is generated, using a cryptographic one-way hash function that takes the address owner's public key and some auxiliary parameters as its input. The address owner can protect a message sent from the address by attaching its public key and auxiliary parameters to the message and signing it with the corresponding private key [8]. Thus, the owner asserts its ownership of the address by using the corresponding private key. Upon receipt of the signed message, the intended recipient verifies the binding between the public key and the address by recomputing and comparing the hash value with the interface identifier part of the address. Additionally, the recipient authenticates the address by verifying the signature. However, the CGA-based technique suffers from several limitations. First, it relies on the digital signature that is added to each message sent, but the IP header (the source address) is excluded from the signature. Therefore, an attacker could easily find and store its victim's messages while obtaining the victim's modifier and public key. Second, as a standalone solution, the CGA-based technique does not guarantee the owner's reachability at the claim address; an attacker can easily use its own public key to cryptographically generate a non-used address with a subnet prefix from the victim's network. Third, despite the CGA-based technique's ability to effectively prevent attackers from impersonating valid IPv6 addresses to launch attacks, it cannot prevent attacks on a network that involve redirecting data to a non-used address. Fourth, the CGA-based technique requires heavy computations to calculate and verify the digital signature, which could expose the network entities to denial-of-service attacks, particularly when the entity is an MN and has limited computation power or when it needs to verify digital signatures for many peers at the same time. Fifth, the MN can self-generate a public–private key pair that is not certified by a trusted third party. In this case, the malicious node can easily enter the network and use these keys to assign its care-of addresses from a specific domain, then access the network resources illegally. Finally, as the address owner's public key and digital signature and the auxiliary parameter values are carried in the message delivery procedure to generate the address cryptographically, a certain amount of overhead is incurred due to bandwidth consumption.

B. Existing Protocols to Secure Mobile Anchor Point (MAP) Registration in HMIPv6

The specification of the HMIPv6 protocol [4] suggests the use of the Internet Key Exchange version 2 (IKEv2) protocol [9] to establish a security association between the MN and the MAP, along with the IPSec protocol[10],[11], to protect the LBU and the BA messages. However, the IKEv2 protocol has limitations regarding communication and computational overhead, making it inefficient due to cryptographic operations and the need for four to six messages with two to three round trips to create a security association between the MN and the MAP. Additionally, the IKEv2-based key setup is difficult to achieve in a multi-hop communication environment with dynamic connections, such as the WMN [12]. Conversely, by using the IPSec protocol, the MAP registration protects against outside attacks (i.e., an attacker cannot send a spoofed LBU message instead of the MN). However, the IPSec protocol can authenticate neither the claimed LCoA nor the RCoA, and it cannot prevent the legitimate MN from sending a fake LCoA,

which will cause the MAP to redirect traffic to the victim's location. Additionally, it does not provide a mechanism for the MAP to verify the ownership of the MN's RCoA, through which the MN can use the services in the MAP's domain. It also cannot prevent the attacker from replying to the LBU message that the MN sent earlier to the MAP. As a result, the MAP will redirect all subsequent traffic to the MN's old location. This situation can cause a denial-of-service attack to both the MN and the node that is currently located in the AR.

Kang-Park's security protocol [13] aims to secure the LBU and the fast handover in the HMIPv6. To protect the MAP registration process, the protocol leverages the authentication, authorisation and accounting (AAA) infrastructure [14], through which the MAP issues the authentication ticket to the MN. Two types of AAA servers are employed: one is operated by a home service provider (AAAH), and the other is operated by a service provider in a foreign network (AAAF). In this protocol, the MAP partially protects against resource exhaustion during denial-of-service attacks because the attacker cannot know the session key that is used to secure the LBU message. Moreover, it protects the entities against any replay attack, as the timestamps have been used. However, using the AAA infrastructure causes a long authentication and registration delay. Specifically, when the MAP receives the LBU message from the MN, it cannot authenticate the MN directly and asks the AAAH through the AAAF to generate and send the session key. If the distance between the AAAF and the AAAH is too long and frequent handovers occur, then large delays occur. Note that the MN cannot send and receive data sent from the correspondent node via the MAP until the MAP registration process is complete. This situation causes critical problems in the mobile network (e.g. registration delay). In contrast, Kang-Park's security protocol cannot verify the reachability of an MN at the claimed LCoA and the authenticity of the RCoA; thus, it is vulnerable to malicious flooding attacks from MNs and allows the visiting MNs to use the MAP domain resources illegally. Furthermore, it forces the MN to perform heavy computations, both to generate the session key to secure the LBU message and to verify the signature, which could expose the MN to a denial-of-service attack when the MN has limited computational power.

The ESS-FH protocol [15] is proposed to enhance Kang-Park's protocol [13] by combining the CGA technique [8] and the public-key cryptography operation. The protocol provides a strong key exchange and key independence based on both the public key encryption and the CGA technique. Additionally, it requests the MN to sign the LBU update message with its private key; thus, the MAP protects against redirect attacks. It also allows the MAP to verify the reachability of the MN at the claimed LCoA and the authenticity of the RCoA; thus, it prevents malicious MNs from launching flooding attacks against the MAP and prevents the MNs from using the MAP domain resources illegally. However, the ESS-FH protocol requires the MN to perform four public key operations to complete the registration with the MAP, which could expose the MN to denial-of-service attacks when the MN has limited computational power. Moreover, the protocol requires verification of the CGA-based technique [8] and the entity signature in each message; thus, the MAP registration delay, packet loss and signalling overhead could be increased. The protocol also requires the MN to self-generate its public–private key pair (unauthenticated key pair), and the RCoA is generated based on the CGA technique; thus, the malicious node can easily enter the network and use these keys to assign its RCoA from a specific domain, accessing the network resources illegally.

The HMIPv6sec protocol [16] aims to create an SA between the MN and the MAP based on the CGA technique [8]. It employs the following requirements: (1) The MN has a self-generated public–private key pair. (2) The LCoA of the MN is generated by using the CGA-based technique (3) The RCoA of the MN is generated by using the secret key shared between the MN and the AR, and the MAP's prefix is advertised by the MAP through the AR. (4) There are existing secure links between all the ARs located within the MAP tree. (5) The MN and the MAP use the Diffie-Hellman key exchange protocol to compute the secret shared key. The HMIPv6sec protocol requires the MN to sign the LBU message using the secret key shared with the AR, which partially protects the MAP against a denial-of-service attack. Furthermore, it increases security by ensuring the LCoA and the RCoA ownership, protecting the MAP against return-to-home spoofing attacks and preventing MNs from using the MAP domain resources illegally. However, this protocol cannot guarantee the reachability of the MN at the claimed LCoA; thus, it cannot protect third parties against denial-of-service attacks.

C. Reachability Test

A reachability test mechanism provides assurance that the MN is indeed located the claimed care-of address [17]. This section will examine various protocols have been adopted reachability test in their proposed.

Mobile IPv6 protocol used Return Routability procedure [3] to assist the correspondent node to assure that the MN can receive messages sent to the claimed home-of address and care-of address. The Return Routability procedure performs two reachability tests: a home-of address test and care-of address test. The MN performs the both tests simultaneously by sending a Home Test Init message its home agent and a Care-of Test Init message directly to the correspondent node. When the correspondent node receives the Home Test Init message, it sends a Home-of Test message to MN via home agent, including a secret home keygen token. Additionally, the correspondent node sends a Care-of Test message directly to MN in response to the Care-of Test Init message contains a secret care-of keygen token. However, the Return Routability procedure fails to provide sufficient protection for the correspondent registration. The attacker can sniff only the Home-of Test message from the correspondent node to forge and initiate a Care-of Test Init message by using its care-of address instead of the legitimate care-of address of the MN. Thus, the correspondent node replies to the Care-of Test message that the attacker uses in integration with the intercepted Home-of Test message to compute a binding management key. The attacker then impersonates the legitimate MN and sends a fake BU message to the correspondent node. As a result, all traffic redirects to the attacker instead of the MN.

The early binding update (EBU) protocol [18] is improved the Return Routability protocol [3] by shifting the home-of-address and care-of address reachability test to the handover phase where they cannot impact the registration delay. The home-of-address test is executed prior to the handover (i.e. the MN still in the old care-of address). After the handover, the care-of address test runs in parallel with data transfer to and from the new care-of address. However, the EBU protocol pays the cost for this reduction delay as the MN needs to run the home-of-address test periodically, that could increase the signalling overhead. Furthermore, it suffers from the same on-path attacks applicable to the Return Routability protocol.

Applying CGAs technique to optimise mobile IPv6 (CGA-OMIPv6) protocol [19] suggested to authenticate MN's home-of-address by using the CGA technique [8] together with an exchanging Home Test Init message and Home Test messages with the correspondent node to proof the reachability of MN at the home-of-address. Thus, the protocol partially protects against return-to-home spoofing attacks. On the other hand, to proof the reachability of MN at the care-of address, the MN exchanges Care-of Test Init message and Care-of Test message with the correspondent node, which partially protects correspondent node against denial-of-service attacks. However, the protocol required the MN and the correspondent node to perform two public key operations during the correspondent registration process, which is increased the complexity imposed on the both entities and led to increase the registration delay.

A novel scheme for supporting location authentication of mobile nodes [20] has been proposed to enhance the basic operation of the home registration defined in [3]. In this scheme, the authors suggested to add two extra mobility-related messages to allow the home agent to confirm the reachability of MN in the claimed care-of address. Both these messages are authenticated using the secret key shared between the MN and home agent. By using this method of reachability test, the scheme prevents the MN from lunching flooding attack. However, this method of reachability test suffers the following limitation. (1) Two additional messages are required to verify the reachability of the MN at claimed care-of address which could expose the MN to denial-of-service attack, as the MN in its nature is mobile entity and has limited computational power. (2) The MN wish to receive the packet from its correspondent node as soon as handoff to foreign network, the proposed reachability test introduced longer delay compared with basic operation defined in the MIPv6 to complete the home agent registration.

III. Preliminaries

Before presenting the E-MAP registration protocol, this section provides details of the assumptions and notations that the following sections use.

A. Assumptions

- Each AR with the MAP has a preconfigured security association for encrypting and authenticating communication exchanges. This assumption is justified by the fact that the HMIPv6 protocol and this study's contribution require ARs within the MAP tree to be involved in delivering the control mobility exchange messages and other packets sent by the MAP to MNs. Therefore, the IPsec Encapsulating Security Payload (ESP) protocol is used, and the AR and the MAP share the secret key ($K_{AR\text{-}MAP}$) [4], [16].
- A MAP has a public–private key pair (PK_{MAP}, SK_{MAP}). The private key SK_{MAP} is kept by the MAP and obtains a public key certificate ($Cert_{MAP}$) from a CA. The MAP possesses the key pair before the invocation of the protocol. The MAP then distributes its $Cert_{MAP}$ among the ARs that are located under the same coverage in the WMN domain. When the ARs receive the MAP's public key certificate, they check its validity with the Certificate Authority before the protocol is invoked.
- An MN and its attached AR have a preconfigured security association for encryption and authenticated communication after the MN completes a link layer handoff. They use the IPsec ESP protocol to protect mobility-related message exchanges. Thus, the MN and the AR share a secret key ($K_{MN\text{-}AR}$).

B. Notations

Table 1 lists the notations to be used in the E-MAP registration protocol description.

TABLE I. List of Notations

Notation	*Indication*		*Notation*	*Indication*
LCoA	Lcaol care-of address		Seq_X	A sequence number of node *X*
RCoA	Regional care-of address		T*x*	A timestamp of node *X*
CoT	Care-of keygen token value		Ack	An acknowledgement value sent by MN to MAP
Modifier	A 128-bit value		MAC*x*	A keyed hash value used to ensure the integrity and authenticity of the message
N_X	A nonce of node *X*		K*x-y*	A shared secret key between two entities
ENG_{KBM}	An encyption value using K_{BM}		K_{BM}	A binding key management
IID	Interface identifire		||	Concatenation

IV. E-MAP Registration Protocol

The E-MAP registration protocol is designed based on two combined ideas. First, the E-MAP registration uses a novel, lightweight, improved version of the traditional CGA-based technique [8] to cryptographically generate and verify the MN's LCoA and RCoA. This is called the public key certificate CGA-based technique. Second, the E-MAP registration protocol uses the light-weight LCoA reachability test method to allow the MAP to confirm reachability of the MN at a claimed LCoA. In addition, the E-MAP registration protocol allows the MAP to securely identify and establish a shared secret key with the MN protect any future mobility messages that could be sent from the MN to the MAP against the possibility of a false mobility messages attack.

A. Public Key Certificate CGA-based Technique

The first idea of this study's proposed protocol aims to reduce the likelihood of a malicious MN stealing other nodes' addresses (i.e., LCoA and RCoA). It uses an improved version of the CGA-based technique [8], that is, a public key certificate CGA-based technique for cryptographic generation and verification of IPv6 addresses (i.e., LCoA and RCoA). The public key certificate CGA-based technique requires that a $Cert_{MAP}$ be distributed among the ARs in the WMN's domain by the MAP before the invocation of the E-MAP registration protocol. This step allows the ARs to cryptographically generate the MN's LCoA and RCoA on behalf of visiting MNs, and the MAP uses the same public key certificate to cryptographically verify the ownership of those addresses.

When the AR generates a CGA-based LCoA, it uses two input values: (1) a 64-bit subnet prefix of the AR and (2) the MAP's public key. The AR then uses the result of the MN's LCoA to compute the MN's RCoA by inputting two values: (1) the leftmost 64-bit of the output interface identifier (*IID*), which is computed based on the LCoA, and (2) the 64-bit prefix of the MAP, which is included in the RA message that is generated by the MAP. The AR runs the public key certificate CGA-based generation to compute the LCoA and the RCoA for the MN as soon as it receives the secure request message from the MN. The details are presented below and shown in Fig.1.

a) Generate a 128-bit random number called a modifier that is used to further randomise the LCoA generated from the same subnet prefix and the MAP's public key.

b) Concatenate from the left modifier and the AR subnet prefix. The AR then executes the HMAC_SHA1 function on the concatenation, using the MAP's public key (PKMAP), and obtains the leftmost 64 bits of the output. The result is HG: HG = First (64, HMAC SHA1 (PKMAP, (modifier || AR-subnet prefix))).

c) Form an IID from HG by setting both the U/L and the I/G bits to zero.

d) Concatenate the AR subnet prefix-(64 bits) and the IID-(64 bits) to form an IPv6 address-(128 bits) with the subnet prefix to the left and the IID to the right.

e) Perform a duplicate address detection [21] test at the LCoA. If an address collision is detected, increment the modifier by one and return to step *b*.

The outputs of the above steps are (1) a new CGA-based IPv6 address (an LCoA) and (2) the final value of the modifier-(128 bits). The AR then computes the RCoA as follows:

a) Concatenate the cryptographic generation of the LCoA with the MAP's public key (PK_{MAP}), execute the HMAC_SHA1 function to compute the *IID* that was used to compute the RCoA for the MN, and obtain the leftmost 64 bits of the output (*IID*). The result is (*IID*): *IID = First (64, HMAC SHA1 (PK_{MAP}|| LCoA)).*

b) Concatenate the MAP-subnet prefix-(64 bits) with 64 bits of the *IID* to form the 128-bit IPv6 address with the MAP prefix to the left and the *IID* to the right. The result is RCoA: *RCoA = (MAP-subnet prefix (64 bit) || IID″ (64 bit)).*

The output of the above steps is a new IPv6 address (i.e., an RCoA). In this case, the LCoA and the RCoA address generation is completed. In parallel, the AR then securely sends a pre-binding update (PBU) message to the MAP and a reply request message to the MN. The PBU message includes the MN's LCoA and RCoA. When the MAP receives the PBU message, it creates a binding cache entry for the MN and stores the values of the LCoA and the RCoA. Once the cache entry is created, the MAP waits for a limited amount of time for the owner of those addresses to send the BU message. If no valid BU message is received during the binding cache entry's preconfigured lifetime, the MAP will delete this cache entry. When the MN receives the reply request message from the AR, it securely sends the LBU message to the MAP that includes the LCoA and the RCoA.

Upon receiving the LBU message, the MAP verifies the ownership of the claimed LCoA and the RCoA, using the public key certificate CGA-based verification. The process of verification is shown in Fig.2 and detailed below:

a) The MAP first verifies the claimed LCoA. The MAP divides the LCoA into a subnet prefix-(64-bit) and an *IID*-(64-bit).

b) Concatenate from the left of the modifier and the subnet prefix. Execute the HMAC_SHA1 function on the concatenation using the MAP's public key (PK_{MAP}) and obtain the leftmost 64 bits of the output. The result is H_V, that is, H_V *= First (64, HMAC_SHA1 (P_{MAP}, (modifier ||subnet prefix)))*.

c) Compare the calculated hash value (H_V) obtained from step *'b'* with the *IID*-(64-bit) obtained from step *'a'*; the differences in the U/L and the I/G bits are ignored.

d) The MAP then verifies the claimed RCoA. The MAP divides the RCOA into the MAP prefix-(64-bits) and the *IID*-(64-bit).

e) Concatenate the LCoA with the MAP's public key (PK_{MAP}) to compute the *IID″*, that is, *IID″ = First (64, HMAC_SHA1 (PK_{MAP}|| LCoA)).*

f) Compare the calculated *IID″*-(64-bit) value obtained from step *'e'* with the *IID*-(64-bit) obtained from step *'d'*. The MAP then forms the RCoA, that is, *RCoA = (MAP-subnet prefix (64 bit) || IID″ (64 bit)).*

g) The MAP then performs a duplicate address detection test at the RCoA. If an address collision is detected, increment the modifier by one and return to step '*b'*.

If both address verifications are successful, the MAP will gain confidence that the CGA-based LCoA and RCoA were generated and sent first by the AR and then by the MN within the WMN domain, either belonging to the MN itself or are non-used addresses. The reason is that with the public key certificate CGA-based technique in place, a malicious MN will need about (2^{61}) attempts to produce either the LCoA or the RCoA that matches a third party's IPv6 address.

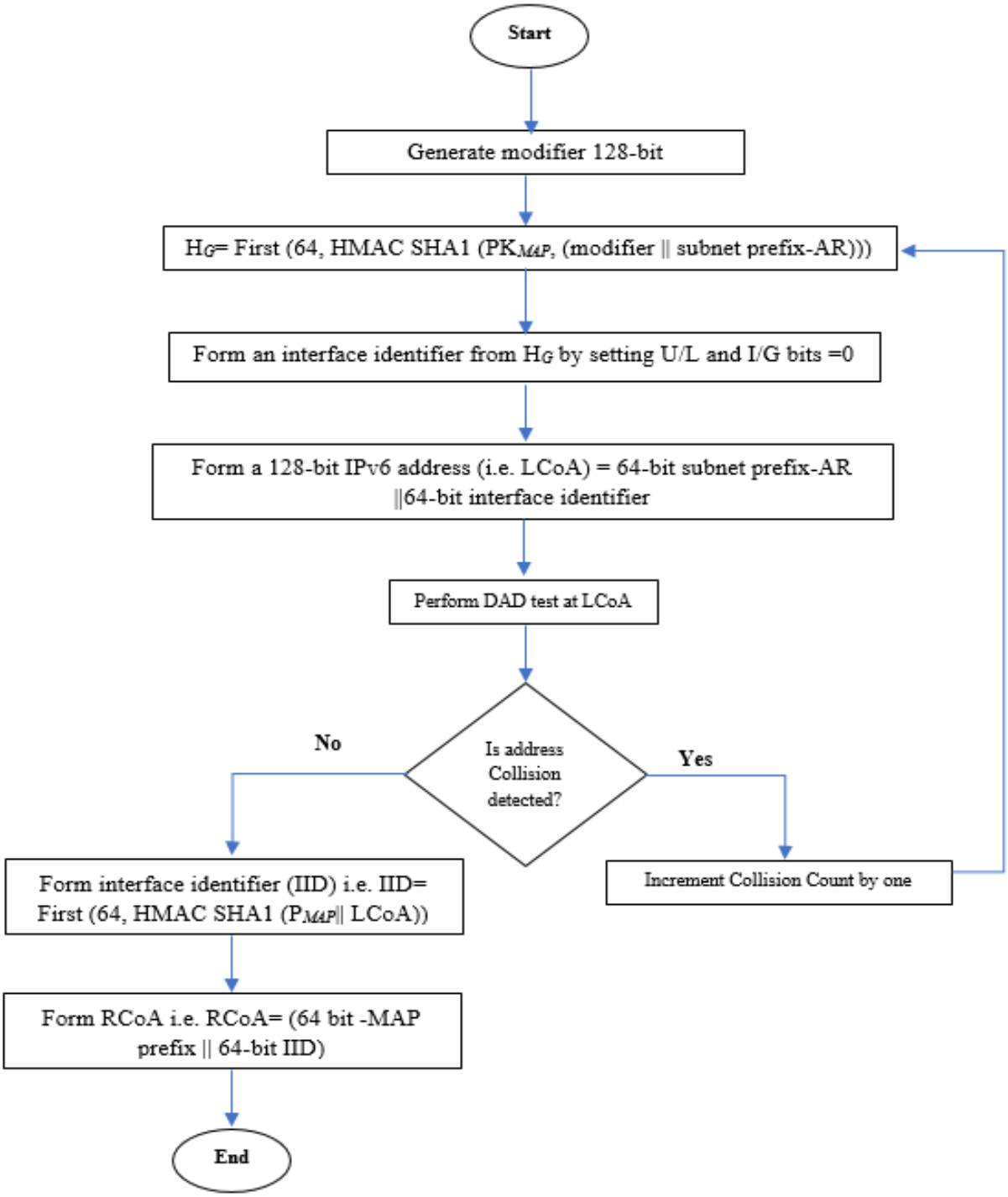

Fig. 1. LCoA and RCoA Generation Process.

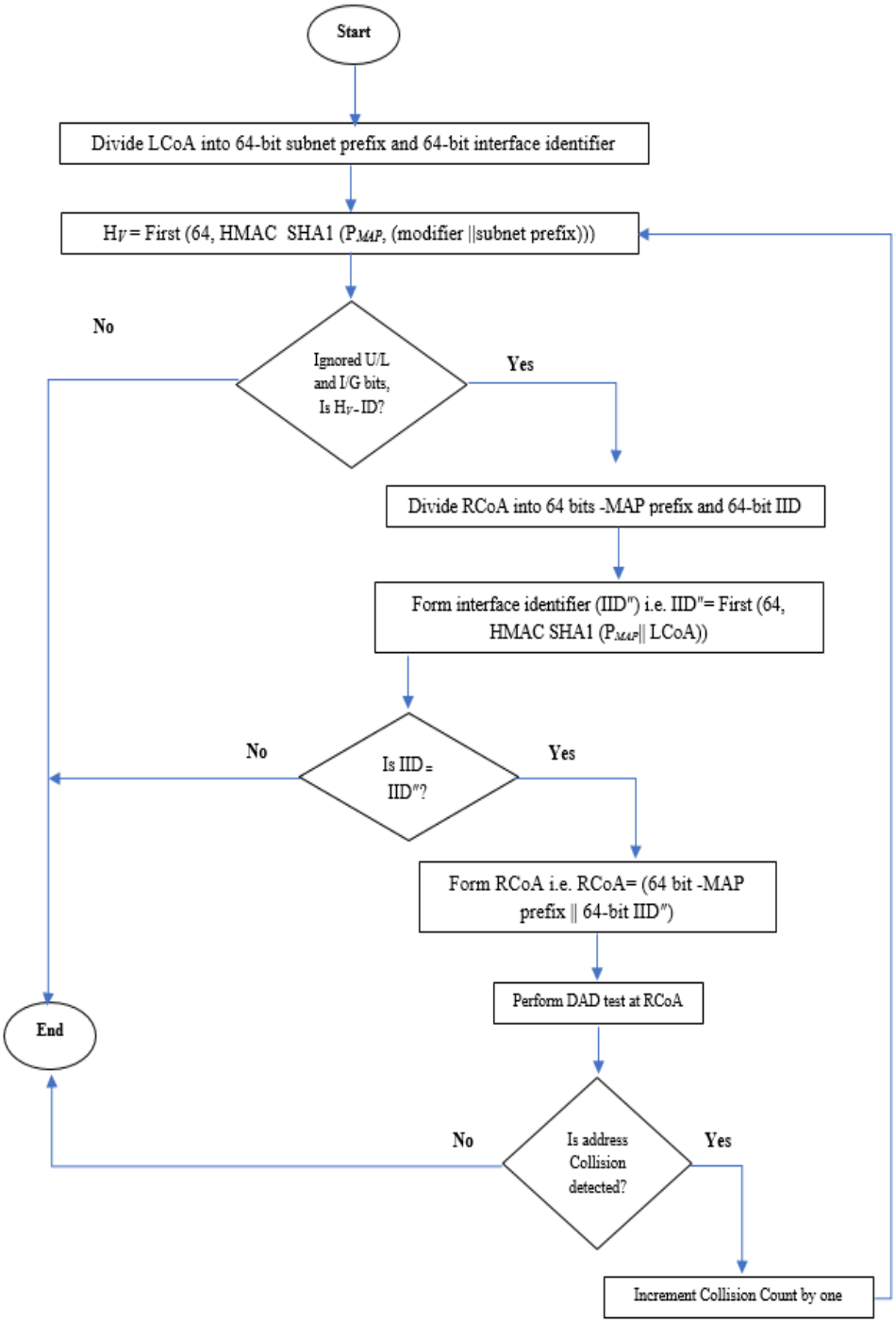

Fig. 2. LCoA and RCoA Verification Process.

B. A Light-Weight LCoA Reachability Test Method

The aim of the light-weight reachability test method to allow the MAP to confirm the reachability of the MN at the claimed LCoA. To doing so, the AR assists the MAP to ensure that the MN is indeed located at the LCoA. The AR generates a fresh care-of keygen token (CoT) value based on the secret key ($K_{AR\text{-}MAP}$) that is shared with the MAP, and the AR then securely sends the CoT value to both the MN and the MAP.

The reachability test is initiated as soon as the MAP receives a valid LBU message from the MN. The MN includes the received CoT value from the AR to show its presence at the claimed LCoA; in other words, the MN sends an LBU message containing the CoT value to the MAP. If the MAP successfully verifies the CoT value, it can then ensure that the MN is indeed located at the claimed LCoA. This method prevents a malicious MN from launching a flooding attack.

The LCoA reachability test is new in the HMIPv6 protocol context; this is the first test that incorporates the CoT's value to enable the MAP to verify the reachability of the MN at the claimed LCoA, and it does not affect the E-MAP registration protocol performance in terms of registration delay and signalling overhead because it does not require an extra message between the MN and the MAP to confirm the LCoA reachability.

V. E-MAP Registration Protocol in Detail

This section presents a detailed description of the E-MAP registration protocol. As shown in Fig. 3, the E-MAP registration protocol executes five messages to perform the MAP registration process. Each message has a specific name, as follows: Router Solicitation (RtSol); Pre-binding Update (PBU); Router Acknowledgement (RtAck); Local Binding Update (LBU); and Binding Acknowledgement (BA).

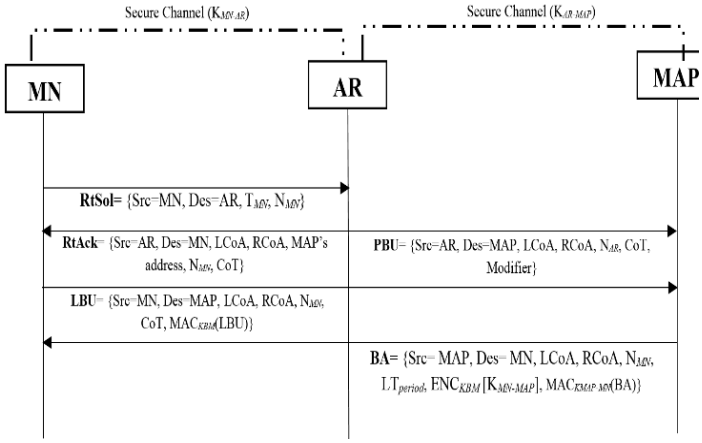

Fig. 3. E-MAP Registration Protocol Sequence Messages.

The E-MAP registration protocol is detailed in the following procedures:

- When the MN roam into the WMN domain, it registers with the MAP in the domain by using the E-MAP registration protocol. The MN initiates the E-MAP registration protocol by securely sending the RtSol message – as shown in (1) – to AR requesting to generate LCoA and RCoA. The MN includes a fresh timestamp (T_{MN}) and fresh nonce (N_{MN}). The T_{MN} is used to protect the AR against replay attack, and the N_{MN} is used to protect the MN against replay attack when it finds a response from the AR.

$$RtSol = \{T_{MN}, N_{MN}\} \quad (1)$$

- Upon the AR received message RtSol message, it checks the value of T_{MN} to confirm the freshness of the message. Upon successful verification, the AR runs the public key certificate CGA-based generation to configure LCoA and RCoA for MN as stated in Section (IV). The AR then generates a CoT – as shown in (2) – based on the secret key (K_{AR-MAP}) shared with the MAP and the fresh nonce N_{AR}.

$$CoT = First\ (64, HMAC\_SHA1\ (K_{AR-MAP}, (LCoA || RCoA || N_{AR}))) \quad (2)$$

The AR then sends the PBU message – as shown in (3) – to the MAP. At the same time, the AR sends the RtAck message – as shown in (4) – to the MN via an IPSec ESP secure tunnel.

$$PBU = \{LCoA, RCoA, Modifier, N_{AR}, CoT, MAC_{KAR-MAP}(PBU)\} \quad (3)$$

$$RtAck = \{LCoA, RCoA, MAP_{address}, N_{MN}\} \quad (4)$$

- When the MAP received PBU message, it checks the value of CoT to verify the freshness of message. Upon successful verification, the MAP verifies the integrity and authenticity of the received message using the key K_{AR-MAP} shared with AR. If any of these verifications fails, the MAP will discard the message without any further action. Otherwise, the MAP creates a binding cache entry for the MN, in which it stores the LCoA, RCoA, CoT and Modifier carried by the PBU message. Once the binding cache entry is created, the MAP waits for a limited amount of time for the owner of those addresses to send the LBU message. If no valid LBU message is received from the MN during the binding cache entry's preconfigured lifetime, then the MAP will delete it.

- When the MN received an RtAck message, it decrypts the message using the (K_{MN-AR}) key, and it checks the value of N_{MN} to verify the freshness of the message. If the verification fails, the MN will discard the message without any further action. Otherwise, the MN creates a binding list entry for the MAP and sets the status to Binding_Pending, indicating that it is waiting for acknowledgements from the MAP. The MN stores the LCoA and the RCoA values in its binding list entry and then generates fresh sequence numbers (Seq_{MN}) that it will use to detect any replay attack when it finds a response from the MAP. The MN then hashes a CoT value enclosed in the RtAck message to generate a binding management key (K_{BM}), as shown in (5). The MN then uses the MAP's address enclosed in the RtAck message to send an LBU message – as shown in (6) –to the MAP.

$$K_{BM} = SHA1\ (CoT) \quad (5)$$

$$LBU = \{LCoA, RCoA, Seq_{MN}, CoT, Ack, MAC_{KBM}(LBU)\} \quad (6)$$

- Upon the MAP received an LBU message, it checks the value of CoT to verify the freshness of the message and the reachability of MN the at claimed LCoA (as stated in Section IV). If the verification fails, the MAP will discard the message without any further action. A positive outcome from this verification check assures the MAP that the LBU message is fresh and comes from a node that is reachable in the LCoA, which provides some assurance of the MN's honesty before heavy computations are performed. This verification protects the MAP against replay attacks and resource exhaustion from denial-of-service attacks. Otherwise, The MAP verifies the ownership of the claimed LCoA and the RCoA, using a public key certificate CGA-based technique (as stated in Section IV). Aftre sucessful the ownership verfication, the MAP hashes

the value of CoT to generate the binding key management (K_{BM}). The MAP then verifies that MAC_{KBM} (LBU) value. A positive result from this verification check assures the MAP that the LBU message is indeed from the MN and has not changed in transit. If any of above verifications fails, the MAP will discard the LBU message with no further action. Otherwise, the MAP updates the values of the MN's LCoA and RCoA and stores the Seq_{MN} value of the MN that was carried in the LBU message in a binding cache entry for the MN. The MAP then generates a fresh session key (K_{MN-MAP}), and sets a lifetime period between the LCoA and RCoA of MN in the binding cache entry to the maxium lifetime value to reduce the number of redundant binding refreshes and by extension, signalling overheads. Finally, the MAP sends the BA message – as shown in (7) – to the MN for an acknowledgement of the binding of the LCoA and the RCoA.

$$BA = \{LCoA, RCoA, Seq_{MN}, LT_{period}, ENC_{KBM}[K_{MN-MAP}], MAC_{KMN-MAP}(BA)\} \quad (7)$$

- When the MN receives a BA message, it will use the MAP's address as an index to search its Binding Update list. If a list entry is found with a Binding_Pending status, the MN will verify the freshness of the message. Upon successful verification, the MN decrypts the code session key K_{MN-MAP} using the K_{BM}. The MN then verifies integrity and authenticity of the BA message. If any of these verifications fail, the MN will discard the message without any further action. Otherwise, the MN will story the values of LCoA, RCoA, lifetime period and K_{MN-MAP}, and change the status of binding Update list to Binding_Complete at match list entry. In this case, the E-MAP registration protocol is completed.

VI. Formal Security Analyses

This section formally verifies the accuracy of the E-MAP registration protocol presented in Section V. To doing so, the Scyther tool [22] has been used. According to [23], the Scyther tool is one of the fastest tools which stills finding attacks efficiently. To model the E-MAP registration protocol in the Scyther tool, the security protocol description language (SPDL) has been used.

Fig.4, summarises the experimental outcomes of the automatic formal verification of the E-MAP registration protocol. The figure shows that the results of the verification can confirm the security properties (i.e., the secrecy of the exchanged values) to guarantee that the information has not been stolen but exchanged safely. In addition, it shows that each role in the E-MAP registration protocol (i.e., MN, AR and MAP) meets the four major authentication forms those defined in the Scyther tool [24],[25] namely: Aliveness (*Alive*), Non-injective agreement (*Niagree*), Non-injective synchronisation (*Nisynch*) and Weak agreement (*Weakagree*).

To conclude, the outcome means that no attacks are found on Scyther's automatic security claim verifications for the E-MAP registration protocol.

Claim				Status		Comments
E_MAPR	MN	E_MAPR,MN1	Secret k(MN,AR)	Ok	Verified	No attacks.
		E_MAPR,MN2	Secret CoT	Ok	Verified	No attacks.
		E_MAPR,MN3	Secret Kmapmn	Ok	Verified	No attacks.
		E_MAPR,MN4	Secret Kbm	Ok	Verified	No attacks.
		E_MAPR,MN5	Secret SeqMN	Ok	Verified	No attacks.
		E_MAPR,MN6	Alive	Ok	Verified	No attacks.
		E_MAPR,MN7	Weakagree	Ok	Verified	No attacks.
		E_MAPR,MN8	Niagree	Ok	Verified	No attacks.
		E_MAPR,MN9	Nisynch	Ok	Verified	No attacks.
	AR	E_MAPR,AR1	Secret k(MN,AR)	Ok	Verified	No attacks.
		E_MAPR,AR2	Secret k(AR,MAP)	Ok	Verified	No attacks.
		E_MAPR,AR3	Secret CoT	Ok	Verified	No attacks.
		E_MAPR,AR4	Secret modifier	Ok	Verified	No attacks.
		E_MAPR,AR5	Alive	Ok	Verified	No attacks.
		E_MAPR,AR6	Weakagree	Ok	Verified	No attacks.
		E_MAPR,AR7	Niagree	Ok	Verified	No attacks.
		E_MAPR,AR8	Nisynch	Ok	Verified	No attacks.
	MAP	E_MAPR,MAP1	Secret k(AR,MAP)	Ok	Verified	No attacks.
		E_MAPR,MAP2	Secret Kmapmn	Ok	Verified	No attacks.
		E_MAPR,MAP3	Secret CoT	Ok	Verified	No attacks.
		E_MAPR,MAP4	Secret modifier	Ok	Verified	No attacks.
		E_MAPR,MAP5	Secret SeqMN	Ok	Verified	No attacks.
		E_MAPR,MAP6	Alive	Ok	Verified	No attacks.
		E_MAPR,MAP7	Weakagree	Ok	Verified	No attacks.
		E_MAPR,MAP8	Niagree	Ok	Verified	No attacks.
		E_MAPR,MAP9	Nisynch	Ok	Verified	No attacks.

Done.

Fig. 4. Automatic Security Claims from Scyther Tool.

VII. Simulation Setup and Performance Evaluation

This section evaluates the performance of the E-MAP registration protocol. The performance is measured in terms of the MAP registration delay and signalling overhead. The MAP registration delay is defined as the total time taken to complete the registration with the MAP in the WMN domain, measured in seconds. The signalling overhead is defined as the total amount of HMIPv6 protocol signalling traffic sent and received by all involved entities in each proposed protocol. The control signalling overhead is measured in bits/second. The OPNET^{TM} modeller [26] (version 14.5) has been chosen to simulate the performance of the E-MAP registration protocol under varying network conditions.

Depicted in Fig.5, the chosen scenario involves one MAP that connects the IPv6 internet cloud via wired uplinks and five ARs that connect with each other through a multi-hop wireless backbone. A MAP has two separate interfaces for providing wired connectivity to the internet and wireless connectivity to form the mesh backbone. Every AR also has two interfaces, and both are for wireless connectivity. These interfaces support the operation of a router in two separate channels. One channel

forms a wireless mesh backbone to route packets for the MNs. For the other channel for MN access, the AR uses an ordinary IEEE 802.11g MAC protocol with a data rate of 54 Mbps and a transmission power of 0.1 W with a power threshold of 20 dBm, and it operates within a 100-m range to simulate the AR so that Wi-Fi-compatible devices can easily join the mesh domain. This IEEE 802.11g MAC protocol is used in the WMN [1]. The figure also shows that the three movement trajectories by the white lines of the MN have been made and represent the three different scenarios of handoff of the MN with the AR. The scenarios dependent on the number of hops between the MN and MAP, e.g. one, three and five hops. When the MN enter the domain, the MN performs the link layer (L2) handoff with the AR and then initiates to perform the network layer (L3) handoff by operating the registration process with MAP.

The following subsection presents the analyses of the results obtained from the simulation study of the MAP registration delay and the control signalling overhead. It compares the results of the E-MAP registration protocol, the B-MAP registration protocol [4] and the HMIPv6sec protocol [16].

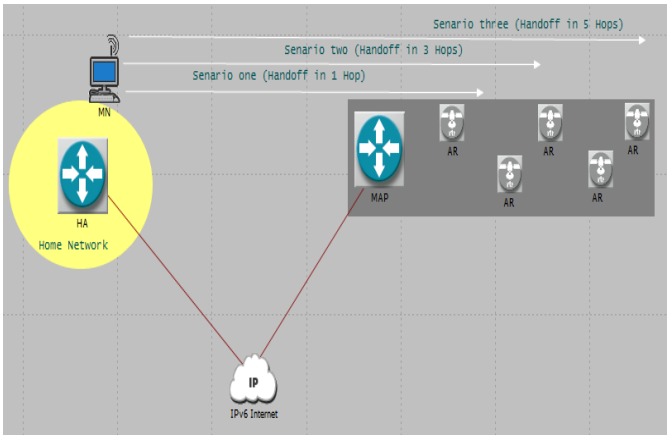

Fig. 5. MAP Registration Process Simulation Model.

A. MAP Registration Delay

This subsection presents an analysis of the MAP registration delay simulation results. A selection of the simulation results is shown in Figures 5-8. In these results, the time spent to examine the uniqueness of the LCoA and the RCoA using the duplicate address detection test is set to zero. The rationale for this setting is that during the implementation of the E-MAP registration protocol, the B-MAP registration protocol and the HMIPv6sec protocol, the duplicate address detection test obtained varying values ranging from 1 second to 1.4 seconds, which could mean that these values affect the accuracy of the collected results.

Fig.6 shows the MAP registration delay at varying numbers of hops (i.e., one, three and five). The figure shows that the MAP registration delay slightly increases as the number of hops increases. This is because the intermediate devices in the WMN domain require time to examine a packet header to determine where to direct it, causing the MAP registration delay to decrease.

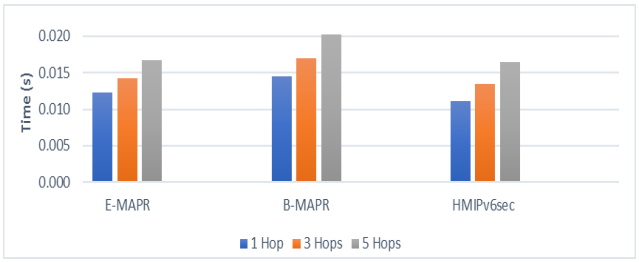

Fig. 6. MAP Registration Delay for E-MAP, B-MAP and HMIPv6sec Protocols Versus Varying Numbers of hops (one, three and five).

For further comparative analysis, Fig.7 shows the average MAP registration delay under different levels of background traffic among the ARs when there are three hops between the MN and the MAP. This figure illustrates that as the background traffic volume increases, ARs become more congested, and the MAP registration delay increases significantly under all protocols. Fig. 8 also shows that on average, the B-MAP registration protocol is around 17% and 19% higher than the E-MAP registration protocol and the HMIPv6sec registration protocol, respectively. This is because the B-MAP registration protocol requires eight messages (i.e., one message to receive the RA message includes the MAP option from the AR plus six messages in the IKEv2 protocol, and two additional messages to bind the LCoA and the RCoA at the MAP. It is also shown that the MN spends around 4% to register its LCoA and RCoA at the MAP in the E-MAP registration protocol, which is higher than in the HMIPv6sec protocol. This is because of (1) the additional two HMAC_SHA1 operations performed by the MAP to verify the reachability of the MN at the LCoA and (2) the increase in the size of the PBU and the LBU messages exchanged in the E-MAP registration protocol compared with the size of these messages in the HMIPv6sec protocol.

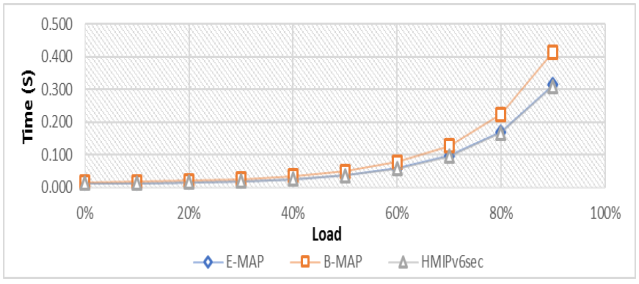

Fig. 7. Average MAP Registration Delay for E-MAP, B-MAP and HMIPv6sec Protocols Versus Load (one MN at three hops).

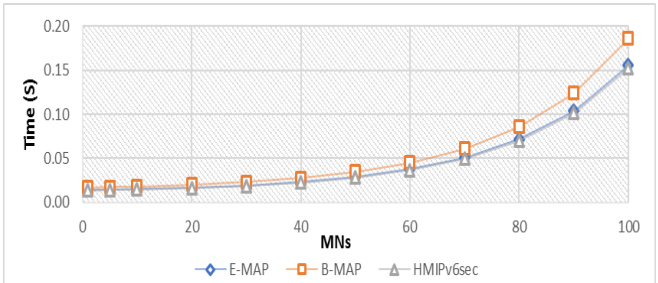

Fig. 8. Average MAP Registration Delay for the E-MAP, B-MAP and HMIPv6sec Protocols Versus the number of MNs.

Fig.8 compares the performance of the E-MAP, B-MAP and HMIPv6sec protocols based on the number of simultaneously visiting MNs served by MAP, which increases when those MNs are three hops away from the MAP, and the background traffic is zero. As shown in this figure, the channel contention at the MAP slightly increases when the number of simultaneously roaming MNs increases from one MN to 50 MNs, but when the number increases to more than 60 MNs, the network load and the channel congestion at the MAP increase exponentially. As also shown in the figures, the E-MAP registration protocol offers a lower rate of MAP registration delay than the B-MAP registration protocol. The reason is that the MAP in the E-MAP registration protocol is required to perform fewer operations during registration than the B-MAP registration protocol. Consequently, the queuing time on the MAP side of the E-MAP registration protocol increases faster than in the B-MAP registration protocol. On the other hand, the E-MAP registration protocol produces slightly more operations than the HMIPv6sec protocol. Therefore, as the number of simultaneously roaming MNs increases, the performance gap between the E-MAP registration protocol and the HMIPv6sec protocol grows slightly.

B. Signalling Overhead

This section analyses the control signalling overhead results for the E-MAP, B-MAP and HMIPv6sec protocols. Table 2 shows the numerical results for the control traffic received by and sent from the MN, AR and MAP sides.

The following observations can be made from Table 2: (1) Generally, the E-MAP registration protocol receives and sends a high amount of control traffic compared with the B-MAP and HMIPv6sec protocols because the E-MAP registration protocol requires an extra length of signalling messages received and/or sent at all that entities (2) The amount of control traffic received at the MN in the E-MAP registration protocol is around 19.4% and 16.3% higher than in the B-MAP and HMIPv6sec protocols, respectively; additionally, the amounts of control traffic sent at the MN in the E-MAP and B-MAP registration protocols are identical. (3) The amounts of control traffic received and sent at the AR in the E-MAP and HMIPv6sec protocols are somewhat identical, but zero bit/second is noted in the B-MAP registration protocol. (4) The amount of control traffic received and sent at the MAP in the E-MAP protocol is higher than in the B-MAP and HMIPv6sec protocols.

TABLE II. CONTROL TRAFFIC RECEIVED AND SENT ON THE MN, AR AND MAP SIDES IN E-MAP, B-MAP AND HMIPV6SEC PROTOCOLS

Protocol	Control traffic (bit/second)	MN	AR	MAP
E-MAP	Received	4.90	1.83	5.19
	Sent	3.66	2.38	2.83
B-MAP	Received	4.03	0	2.01
	Sent	3.66	0	1.83
HMIPv6sec	Received	4.16	1.83	4.43
	Sent	3.13	2.12	2.54

C. Discussion

The simulation study involving the E-MAP registration protocol reveals the following findings: (1) Increasing the number of hops between the MN and the MAP has an insignificant effect on the performance of the E-MAP, B-MAP and HMIPv6sec protocols in terms of the MAP registration delay. (2) Increasing the background traffic volume among the ARs in the WMN domain has a lower impact on the performance of the E-MAP registration protocol than that of the B-MAP registration protocol but has a slightly higher effect compared with the HMIPv6sec protocol. (3) The impact of increasing the number of simultaneously roaming MNs served by the same MAP on the performance of the E-MAP registration protocol is lower compared with the B-MAP registration protocol and higher compared with the HMIPv6sec protocol. (4) The E-MAP registration protocol receives and sends more control traffic at the MN, the AR and the MAP as a cost to pay for supporting the location authentication of the MN to the MAP, allow the MN to use the WMN domain services and compute the shared secret key between the MN and the MAP.

To conclude, if the E-MAP, B-MAP and HMIPv6sec protocols are compared based on efficiency and security, the E-MAP registration protocol emerges as the superior option.

VIII. CONCLUSION AND FUTURE WORK

This paper has presented the design of the E-MAP registration protocol that allows the MAP to verify the MN's ownership of the claimed LCoA and RCoA, as well as the reachability of the MN at the LCoA, and to securely identify and compute the shared secret key with the MN in the WMN domain. This study has combined two ideas. First, it has generated the MN's LCoA and RCoA using a public key certificate CGA-based technique. Second, it has applied a novel lightweight method to confirm the reachability of MN at claimed LCoA. In addition, the MAP computes the shared secret key with MN. Via these actions, the E-MAP registration protocol reduces the likelihood that a malicious MN can successfully steal a third party's node addresses (i.e., the LCoA and the RCoA), prevents a malicious MN from launching a flooding attack and protects any future BU against a false BU attack. The formal security analysis using the Scyther tool has demonstrated that no attacks haven found in the E-MAP registration protocol. Additionally, the performance evaluation has proven that increasing the number of hops in the WMN domain has little effect on the performance of the MAP registration process in the E-MAP, B-MAP and HMIPv6Sec protocols. The E-MAP registration protocol also offers a lower registration delay than the B-MAP protocol but higher than the HMIPv6sec protocol. Moreover, the E-MAP registration protocol introduces a higher signalling overhead compared with the B-MAP and HMIPv6sec protocols as a cost to support the MN location at the WMN domain, allow the MN to use the services of the WMN domain and configure the shared secret key between the MN and the MAP. Further research is recommended to extend the E-MAP registration protocol to securely handle MNs roaming within the WMN domain.

ACKNOWLEDGMENT

Sultan Alkhliwi gratefully acknowledges the Northern Border University, 'KSA' for financial support

REFERENCES

[1] I. F. A. and Xudong, Wireless Mesh Networks (Advanced Texts in Communications and Networking). John Wiley & Sons, 2009.

[2] C. K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall PTR, 2002.

[3] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775 (Proposed Standard), 2004.

[4] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," RFC 5380 (Proposed Standard), 2008.

[5] T. Aura, "Mobile IPv6 Security," in Security Protocols, 2004, pp. 215–234.

[6] A. Moravejosharieh, H. Modares, and R. Salleh, "Overview of Mobile IPv6 security," Proc. - 3rd Int. Conf. Intell. Syst. Model. Simulation, ISMS 2012, pp. 584–587, 2012.

[7] G. O'Shea and M. Roe, "Child-proof authentication for MIPv6 (CAM)," Acm Sigcomm, vol. 31, no. 2, p. 4, 2001.

[8] T. Aura, "Cryptographically Generated Addresses (CGA)," RFC 3927 (Proposed Standard), 2005.

[9] E. Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," RFC 4306 (Proposed Standard), 2005.

[10] S. K. and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301 (Proposed Standard), 2005.

[11] S. Bridglal et al., The TCP / IP Guide. .

[12] R. Kandikattu and L. Jacob, "Comparative analysis of different cryptosystems for hierarchical mobile IPv6-based wireless mesh network," Int. J. Netw. Secur., vol. 10, no. 3, pp. 190–203, 2010.

[13] H.-S. Kang and C.-S. Park, "Authenticated Fast Handover Scheme in the Hierarchical Mobile IPv6," Wisa, vol. 4298, p. 211-224--, 2007.

[14] C. Perkins and P. Calhoun, "Authentication, Authorization, and Accounting (AAA) Registration Keys for Mobile IPv4," 2005.

[15] I. You, J. H. Lee, K. Sakurai, and Y. Hori, "ESS-FH: Enhanced security scheme for fast handover in hierarchical mobile IPv6," IEICE Trans. Inf. Syst., vol. E93–D, no. 5, pp. 1096–1105, 2010.

[16] W. Haddad, S. Krishnan, and H. Soliman, "Using Cryptographically Generated Addresses (CGA) to secure HMIPv6 Protocol (HMIPv6sec)," Expired Internet-Draft: draft-haddad-mipshop-hmipv6-security-06, 2006.

[17] C. Bauer, Secure and Efficient IP Mobility Support for Aeronautical Communications. KIT Scientific Publishing, 2013.

[18] C. Vogt, R. Bless, M. Doll, and T. Kuefner, "Early binding updates for Mobile IPv6," IEEE Wirel. Commun. Netw. Conf. WCNC, vol. 3, no. C, pp. 1440–1445, 2005.

[19] W. Haddad, L. Madour, J. Arkko, and F. Dupont, "Applying Cryptographically Generated Addresses to Optimize MIPv6 (CGA-OMIPv6)," Expired Internet-Draft: draft-haddad-mip6-cga-omipv6-04.txt, 2005.

[20] O. Elshakankiry, A. Carpenter, and N. Zhang, "A Novel Scheme for Supporting Location Authentication of Mobile Nodes," in In Proceeding of the Second International ICST Conference on Security and Privacy in Mobile Information and Communication Systems (MobiSec2010), 2010, pp. 91–102.

[21] T. J. S. Thomson, T. Narten, "IPv6 Stateless Address Autoconfiguration," RFC 4862 (Draft Standard), 2007.

[22] C. J. F. Cremers, "Scyther : Unbounded Verification of Security Protocols," no. 572, pp. 1–18, 2006.

[23] C. J. F. Cremers, P. Lafourcade, and P. Nadeau, "Comparing State Spaces in Automatic Security Protocol Analysis," in Formal to Practical Security: Papers Issued from the 2005-2008 French-Japanese Collaboration, V. Cortier, C. Kirchner, M. Okada, and H. Sakurada, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 70–94.

[24] G. Lowe, "A hierarchy of authentication specifications," 10th Comput. Secur. Found. Work., pp. 31–43, 1997.

[25] C. J. F. Cremers, S. Mauw, and E. P. de Vink, "Injective synchronisation: An extension of the authentication hierarchy," Theor. Comput. Sci., vol. 367, no. 1–2, pp. 139–161, 2006.

[26] "Riverbed Modeler." [Online]. Available: https://www.riverbed.com/gb/products/steelcentral/opnet.html. [Accessed: 24-Jan-2016].