Improving authentication function in wireless mobile multicast communication

CONFERENCE PAPER · SEPTEMBER 2015
DOI: 10.1109/ITechA.2015.7317435
Improving Authentication function in wireless mobile multicast communication

T.T. Mapoka¹, 1,2, Y.A.S Dama, 3 H.M. Alsabbagh, S.J. Shepherd¹, K.O. Anoh¹ and R.A. Abd-Alhameed¹
¹Electrical Engineering and Computer Science, University of Bradford, Bradford, BD7 1DP, United Kingdom
²Electrical Engineering, An-Najah National University, Nablus, Palestinian
³Electrical Engineering, Basra University, Basra, Iraq
¹{T.T.Mapoka; R.A.A.Abd}@bradford.ac.uk

Abstract—In this paper a distributed authentication scheme based on independent session key per access network (HOISKA) is proposed for the decentralized multi-service group key management scheme in a wireless multicast environment. It enables a handover user M_i involved in multiple multicast service subscriptions to establish the long term credential from the trusted authentication server (As) during initial registration. The M_i then securely reuses the long term credential established to derive unique session keys per access network during handover across diverse access networks. The distributed nature HOISKA enables offloading the authentication function to the area network controllers (AKDs) such that As does not participate during handover authentication signalling. This simplifies handover by reducing handover exchange signalling constituting to less handover delays. Two scenarios for HOISKA, initial handover access (IAA) and Handover Access authentication (HAA) are presented then analyzed using the delay analytical model. The HOISKA model proves efficacy in both scenarios by inducing less transmission delays with comparable level of security compared to the widely deployed authentication scheme.

Keywords—user handover authentication; security; multicast communication; mobility management; group key management; wireless networks

I. INTRODUCTION

Wireless mobile access networks are emerging in recent years. This has provokes the emergence of portable mobile devices such as laptop and smartphones with the ability to access diverse multimedia applications such Video on Demand, Pay per view services over the internet ubiquitously. The operators’ investments in fixed wireless network infrastructures are significant while users need security for the services that are exchanged in wireless multicast networks. Consequently these operators deny users access to their networks unless they have paid for a particular service and users hesitate to send their personal information over the internet without intense protection. Additionally users do not want to pay for services not intended for them. On the other hand, due to the open nature of wireless multicast networks, malicious users also want to freely access the network anonymously hence masquerading as other users. However they are multiple motivations to provide access control and service protection in wireless mobile access networks hence minimizing the effects of a compromised system especially at the edge of the network where wireless access agents reside in public domains. Thus without the network being secured, a mobile user can forge to be illegitimate to access the network services not intended for them, redirect the traffic packets and launch various attacks deeply in the network domain. Therefore it is crucial to integrate user authentication and group key management for providing network access control and service access control in the wireless network.

However the convectional group key management (GKM) schemes for service access control over wireless multicast networks [1] only protect a single service at a time using symmetric group key (Traffic Encryption Key (TEK)). The key is used by the service provider to encrypt a particular service which the legitimate group receivers use to decrypt the received service. Due to the dynamic nature of the multicast network, group users may join, leave or perform frequent handoffs especially in mobility environments hence triggering update of the TEK on every join or leave to achieve backward or forward confidentiality respectively [2]. This process is known as rekeying which has been extensively studied to add significant key management overhead in the presence of multi-service subscriptions [3], [4]. Our recent work in [5] proposes a novel rekeying strategy for efficiently addressing multi-service subscription problem in the convectional wireless GKM schemes [1]. To achieve resilient security the work propose integrating both group key management and authentication functions which are distributed efficiently in a decentralized wireless framework to achieve less exchange signalling, scalability and less storage overheads in the presence of multi-services and multi-frequent handoffs. However it is well known that in wireless environment the bandwidth is limited and need to be preserved to improve coverage hence preventing packet loss leading to service disruptions. The mobile user devices are resource limited and require light storage and light computations of the security keys. Therefore it is expected that integrating user authentication with key management in the presence of multi-service subscription may negatively affect the handoff performance, add overhead to the underlying system in terms of multiple key exchange signalling, user authentication and key distribution. Previous wireless mobile multicast GKM schemes such as GKM [6] implicitly assume user authentication method similar to the widely deployed EAP-TLS [7]. Though EAP-TLS is widely used, it has a drawback of synchronizing with the main authenticating AAA server (As) during frequent handoffs. The As may be across the globe hence constituting to significant delays and service packet loss. This is a crucial aspect to study in this paper to reduce the exchange signalling caused by authentication
during initial authentication at registration and handover authentication during handover. The multi-service key distribution technique that uses efficient authentication mechanism based on secure session distribution list (SKDL) [8] is adopted. SKDL is used to securely distribute the authentication functions to the edge of the network for accelerating handoffs with improved exchange signalling performance.

The rest of the paper is organized as follows: In section I the overview of the SMGKM scheme and SKDL concept are provided. The scenarios for initial authentication and handover authentication in the new concept are described in section III. In section IV and V the numerical analysis for both scenarios and security analysis is given respectively while section VI concludes the paper.

II. OVERVIEW OF THE SMGKM MULTI-SERVICE SYSTEM

A scalable two tier decentralized multi-service GKM scheme known as multi-service GKM (SMGKM) scheme [4], [5] was proposed. It consists of the DKD for initial registration of subscribers, initial generation of cryptographic key parameters for authentication and key management. The DKD for generating the key management parameters, AAA server for generating the authentication parameters and the service provider (SP) for providing the services subscribed are assumed to be collocated for network simplicity. The framework also consists of intermediate cluster controllers called the AKDs which operate under the jurisdiction of the DKD for securely establishing and distributing the group key management keys to valid mobile subscribers over a bandwidth limited wireless domain during the subscription period. The mobile subscribers use portable devices like smartphones, Ipad, etc. to wirelessly access their subscribed multimedia services over the air. Each AKD manage key management and authentication keys independently per cluster in order to localize key management and authentication functions. However these functions are delegated securely from the trusted DKD to the intermediate AKDs using a novel Session Key Distribution List (SKDL) to achieve DKD scalability, prevent bottlenecks and unnecessary delays constituting to service disruptions during the system lifetime [8].

A. SKDL Concept Overview

As specified in, the Session Key Distribution List (SKDL) [8] consists of rows specific to the AKD, and the rows are in encrypted form to securely store the private keys SKM AkDi corresponding to the number of registered Mi under AKD, for authentication purpose. The AKD specific rows are also integrity protected using Message Authentication Code (MAC)[9] to prevent replay attacks. Nonces or timestamps may also be used in this case. It includes the rows for the target AKD, where the Mi will visit and each AKD can modify its own rows without affecting the rows of its neighbors. It securely stores the system security parameters initially setup by the trusted DKD which are also delegated securely to the AKDs for group key establishment

However in this paper we focus on adding one novel feature to our SMGKM system known as Handover Optimized Independent Session Key based Authentication (HOISKA) which allows the mobile node Mi to utilize unique session key (SKM_AKD) per cluster during initial access authentication and as the Mi crosses boundaries of various networks during handover it is modified to a new one to ensure key separation per cluster. The SKM_AKD is derived by the As then delivered and stored at the intermediate AKDs using the SKDL concept [8].

III. HOISKA NEW CONCEPT

In HOISKA context of authentication during mobility management, two authentication scenarios are measured: Initial Access Authentication (IAA) of which the Mi initially accesses the currently serving network by providing the necessary credentials to the network during Mi boots up or connect to the network: Handover Access Authentication (HAA) of which Mi detach from the previous network cluster then attach to the target network cluster during as it changes point of attachment to the network (handoff) hence providing the necessary credentials to the target network in order to seamless access the network services. From these scenarios, it is obvious that the IAA latency does not have much impact on the Mi experiences rather than HAA latency which critically contributes to increased overall handover latency hence high likelihood of service disruptions. Before proceeding further, Table I give a definition of the notations used.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>A message X or statement</td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>E(K, X)</td>
<td>Function encrypting X with key K</td>
</tr>
<tr>
<td>K_A,B</td>
<td>Symmetric Key shared between A and B</td>
</tr>
<tr>
<td>A -&gt; B : X</td>
<td>Direction of X sent from A to B</td>
</tr>
<tr>
<td>ID_i</td>
<td>Identifier that uniquely distinguishes A</td>
</tr>
<tr>
<td>T_s</td>
<td>Timestamp generated by A</td>
</tr>
<tr>
<td>T_F</td>
<td>Start and End times for the Timestamp</td>
</tr>
<tr>
<td>N^f_A</td>
<td>An f^th nonce generated by A</td>
</tr>
<tr>
<td>ID_G</td>
<td>Identity of the multicast service group</td>
</tr>
</tbody>
</table>

The following assumptions for SMGKM are maintained: The mobility service provision network agents are timely synchronized, mobility service provisioning network agents have pre-established security association keys amongst them, each network entity is distinguished from each other by using unique identity such as MAC address, Network Access Identifier (NAI), the network agents also have high computation power capability, The multicast routing
protocols such as DVMRP are already enabled and the multicast Internet Group Membership Protocols such as IGMPv2 for IPv4 and MLDv1 for IPv6 are already setup to allow users to join various multicasts groups. Thus with the above notations and assumptions, each scenario of the HOISKA is explained in detail from each subsection.

A. Initial Access Authentication

The IAA consists of initial join access request/response (\(IJ{-}\)AccReq / \(IJ{-}\)AccRes) and authentication joins request/response (\(J{-}\)AccReq/ \(J{-}\)AccRes). During the initial join access request/response, the long term authentication parameter \(AK_m\) and the session key \(SK_{i,m(AKD)}\) are both generated and distributed from the AAA Server (\(AS\)) such as Diameter or RADIUS to the \(Mi\) and the AKD, respectively. Instead, in \(J{-}\)AccReq/ \(J{-}\)AccRes, the \(Mi\) is authenticated by deriving the \(SK_{Mi(AKD)}\) using the obtained \(AK_m\) before entering the target access network.

Step 1: Suppose that the \(Mi\) under the serving access network controlled by AKD, initially registers with the network at boot up. The \(Mi\) randomly chooses a nonce \(N_M^{(i)}\) and sends the \(IJ{-}\)AccReq message to the current AKD, i.e. \(Mi \rightarrow AKD: IJ{-}\)AccReq(\(ID_M\) | \(ID_k\) | \(ID_{acc}\) | \(N_M^{(i)}\)).

Step 2: The AKD, after receiving the \(IJ{-}\)AccReq it also randomly choose its nonce \(N_{AKD}^{(i)}\) then add it to the \(IJ{-}\)AccReq in formation of the Join Access Request message to be sent to the \(As\) (\(IJ{-}\)AccReq\(As\)). Assuming the security association key \(K_{AKD-\allowbreak As}\) is already shared. Thus: \(AKD \rightarrow As: E(K_{\allowbreak AKD-\allowbreak As}, IJ{-}\)AccReq\(As(\allowbreak ID_M\) | \(ID_k\) | \(ID_{acc}\) | \(N_M^{(i)}\) | \(N_{AKD}^{(i)}\)).\) The AAA server (\(As\)) also generates the \(SK_{Mi(AKD)}\) for the corresponding \(Mi\) under AKD, as \(SK_{Mi(AKD)} = E(K_{SK_{Mi(AKD)}}, K_{\allowbreak AKD-\allowbreak Mi\allowbreak }, ID_M, ID_k, T_{i}T_{k}, T_iT_{j})\), where \(T_{i}T_{k}\) is the set start and end times of the corresponding session key, \(K_{\allowbreak AKD-\allowbreak Mi\allowbreak }\) is the key for generating the authenticator during registration at the cluster level. The \(As\) after detecting that \(Mi\) want to initially register in to the underlying system, it notifies the KD to generate the SKDL\(_{i}\) row specific for the \(Mi\) under AKD which securely store the generated \(SK_{Mi(AKD)}\) [8]. Note that here we use nonces to prevent replay attacks in the received SKDL instead of MAC and the format of the SKDL\(_{i}\) is maintained.

Step 3: The \(As\) respond with \(IJ{-}\)AccResAs to the \(AKD\) where \(Mi\) currently resides as \(As \rightarrow AKD: IJ{-}\)AccRes\(As(ID_M, ID_k, ID_{acc},\)
\(E(K_{\allowbreak AKD-\allowbreak As}, ID_M, ID_k, T_{i}T_{k}, N_{Mi}^{(i)}),\)
\(E(K_{\allowbreak AKD-\allowbreak As}, ID_M, ID_k, T_{i}T_{k}, N_{Mi}^{(i)}))).\)

Step 4: The AKD, after receiving the \(IJ{-}\)AccResAs message, it obtains the \(SK_{Mi(AKD)}\) from the SKDL\(_{i}\) row by decrypting part of the \(IJ{-}\)AccResAs message \(E(K_{\allowbreak AKD-\allowbreak As}, ID_M, ID_k, T_{i}T_{k}, N_{Mi}^{(i)}),\) with \(K_{\allowbreak AKD-\allowbreak As}).\)

Step 5: After verifying the \(N_{Mi}^{(i)}\), it forward part of the \(IJ{-}\)AccResAs received message intended for \(Mi\) as

\(AKD \rightarrow Mi: IJ{-}\)AccRes\(s(\allowbreak ID_M, ID_k, ID_{acc}, SK_{Mi(AKD)},\)
\(E(K_{\allowbreak Mi}, SK_{Mi(\allowbreak AKD)}, ID_{acc}, T_{i}T_{k}, N_{Mi}^{(i)})).\)

Step 6: The \(Mi\) also obtain its derived \(SK_{Mi(AKD)}\) then extract \(SK_{Mi(AKD)}\) by decrypting part of the \(IJ{-}\)AccRes with \(AKD\) after verifying its nonce \(N_{Mi}^{(i)}\). The \(AKD\) is pre-shared key between the \(Mi\) and the \(As\) before network access boot up. Note that the AKD, is not aware of the \(AKD\) derivation and the \(AKD\) is assumed to be stored securely in a tamper proof mobile device smartcard. The \(Mi\) now need to derive unique Main session key (\(mSK_{Mi(AKD)}\) specific to the access network \(i\) for communication between the \(Mi\) and the AKD, over the subscription period at the serving network \(i\). Thus the \(Mi\) derive its Main Session Key (\(mSK_{Mi(AKD)}\)) as \(mSK_{Mi(AKD)} = HMAC - SHA1(K_{\allowbreak Mi}, (N_{Mi}^{(i)}), ID_M, ID_{acc})).\) Then \(Mi\) perform the derivation of the authenticator \(\phi_{ij}\) such that \(\phi_{ij} = E(K_{\allowbreak Mi}, ID_M, ID_{acc}).\)

Step 7: Assuming the underlying multicast system is now in operation, the \(Mi\) and the AKD, should mutually authenticate each other. Thus the \(Mi\) now form the join authentication request (\(J{-}\)AuthReq) to be sent to the AKD, as

\(Mi \rightarrow AKD: J{-}\)AuthReq(\(Id_M, Id_k, Id_{acc},\)
\(K_{\allowbreak Mi(AKD)}, \phi_{ij})).\)

The AKD then obtains the \(K_{Mi(AKD)}\) from the \(J{-}\)AuthReq message by decrypting \(SK_{Mi(AKD)}\) with \(K_{SK_{Mi(AKD)}},\) stored in the SKDL\(_{i}\). \(K_{Mi(AKD)}\) is further used to decrypt the \(\phi_{ij}\) hence the AKD obtains the corresponding \(mSK_{Mi(AKD)}\) and \(N_{Mi}^{(i+2)}\) to successfully warrant \(Mi\) access to the network resources. In this context, the resources are the group communication session key shares \(TEK_{ij}\) for the multi-services subscribed by the \(Mi\) which are distributed during the group key distribution phase described in [4].

Step 8: Finally the AKD, acknowledge the \(Mi\) entirely in to the system by sending the join authentication response (\(J{-}\)AuthRes) as:

\(AKD \rightarrow Mi: J{-}\)AuthRes(\(E(\allowbreak K_{\allowbreak Mi}, ID_M, ID_{acc}, N_{Mi}^{(i+2)} + 1, TEK_{ij})).\) Note that the \(J{-}\)AuthRes message contains the fresh \(TEK_{ij}\) shares equivalent to the number of services subscribed by the \(Mi\) depending on which service group \(G_k\) the \(Mi\) belongs to [4]. The \(Mi\) on receiving the \(J{-}\)AuthRes message, it decrypts it using \(K_{\allowbreak Mi(AKD)}\) then verify the nonce value \(N_{Mi}^{(i+2)} + 1\). If valid, the \(Mi\) successfully authenticates the AKD, to obtain the group communication key shares securely. This concludes the IAA operation in HOISKA hence follows secure transmission of multicast services to the \(Mi\) from the SP which is not part of this paper.

B. Handover Access Authentication

HAA involves two common phases: Context Transfer phase enable secure transfer of the rows in SKDL\(_{i}\) containing the \(SK_{Mi(AKD)}\) for the \(Mi\) moving to the target AKD. The
rows correspond to the number of Mi performing handoff and this prepares the target AKDi for fast authentication and group key distribution without involving the domain As as addressed in [8], [5]. The assumption is that context transfer protocol (CTXT) [10] is already in place between the current AKDi and the target AKDi. Join authentication request/response phase which authenticate the handoff user Mi at the target access network v controlled by AKDi before obtaining its service group communication keys. Thus the handover Mi, provide the derived unique SKMi,kDi specific for the access network v and the new authenticator φv before it can be granted access at the target AKDi. Now suppose Mi reaches the cross boundaries of the AKDi and the target AKDi, where the signal coverage Pj for AKDi is lower than that of Pj for AKDv. Thus the Mi performs handoff from the access network by the AKDi to the access network by the AKDv. Assuming also that layer 2 (L2) switch is used to enable fast handover [11]. When the Mi prepares handover at the cross boundaries,

Step 1: The SKDLi, row containing the KSKMi,kDi for Mi is securely transferred within the Context Transfer Authentication message (CtxtTAuthReq) to the AKDi as AKDi ® AKDv: CtxtTAuthReq(E(KMi,kDi, KSKMi,kDi || IDMj || TAKDv || KUS)).

Note that the Key Update Slot (KUS) notified in CtxtTAuthReq message to notify the Target AKDv about the affected services requiring key update (rekeying) during the key management phase described fully in [4],[5].

Step 2: The AKDi, upon receiving the CtxtTAuthReq, it obtains the contents of the message by decrypting with KSKMi,kDi. The AKDi checks the validity of the contents by verifying TAKDv. The AKDi similarly securely notify the AKDv using context Transfer acknowledgement response (CtxtTAuthRes) for successfully receiving the content of CtxtTAuthReq hence prepares for authentication of the Mi in advance before Mi reaches its coverage. If there are any non-real time packets destined for Mi, the AKDi also forward them securely to the AKDv, so that they are buffered until Mi rejoin the target access network to prevent further packet loss. We assume that the AKDi has sufficient memory to store the packets until the Mi arrive.

Step 3: As the Mi is reaches the target access network controlled by AKDv, it undergoes handover authentication in order to be granted access. Thus the Mi notifies the previous AKDi about leaving the access network i by sending Leave authentication request message (L_AuthReq) as Mi ® AKDi: L_AuthReq(SKMi,kDi || φi). The AKDv verifies SKMi,kDi stored in SKDLi, N'Mi(Nj) and the end time for φi. The AKDv finally removes the Mi rows in SKDLi to reserve more storage space for other joining users. Similarly the Mi removes the key parameters used while residing in the previous AKDi. The Mi randomly choose a nonce N'Mi(Nj+1) to generate the new mSKMi,kDi as

\[ mSKMi,kDi = HMAC-SHA1(AMi(N'Mi(Nj+1)) || IDMj || IDAKDi) \]

hence generating the new authenticator φi as

\[ φi = E(KMi,kDi, IDMj || mSKMi,kDi || TDi || N'Mi(Nj+1)) \]

Notice that both new mSKMi,kDi and the new φi are unique per access network to ensure key separation per access network as specified in [5]. Thus the Mi does not maintain history of the local keys used in the previously visited networks. This prevents compromises while enhancing Mi key storage. This is what makes HOISKA very unique.

Step 4: After successful derivation of both new mSKMi,kDi and φi provided Pj ≪Pv, the Mi then send a J_AuthReq to the target AKDi as:

\[ Mi ® AKDi : J_AuthReq(SKMi,kDi || φi) \]

Step 5: After receiving the J_AuthReq from the Mi, AKDi decrypt the message using KkmMi,kDi in SKDLi, then check the validity of the nonce of N'Mi(Nj+1)+1. If valid, the AKDi authenticate Mi successfully. The assumption is that the AKDi has performed key update process using the received KUSi from the AKDv, hence key distribution of the TEKjv for the affected services [4,5] along with the J_AuthRes in step 6.

Step 6: The AKDi sends an acknowledgement response message J_AuthRes containing the group communication keys for the services the Mi is subscribed to as Mi ® AKDi : J_AuthRes(KMi,kDi, IDMj || N'Mi(Nj+1)+1 || TEKjv). The Mi after receiving the J_AuthRes message from the AKDi, the Mi decrypts it using KkmMi,kDi it verifies the nonce value N'Mi(Nj+1)+1. If valid the Mi also authenticates the AKDi hence obtaining the message contents successfully before entering the target access network v. If they are any non-real time packets buffered at the target AKDi, the Mi receives them along with the services from the SP during the service distribution phase not covered in the paper.

IV. AUTHENTICATION DELAY ANALYSIS

In this section, we measure the performance of HOISKA scheme in terms of authentication latencies in comparison to the widely deployed EAP-TLS scheme. The authentication delay induced by authentication exchange signalling during handoff is the main performance impact factor since it negatively constitute to increasing network delays which adversely disrupt services. The assumption is that Mi and other participating network agents have sufficient capacity to operate cryptographic operations.

A. Authentication Delay Analysis

The delays induced in HOISKA and EAP-TLS are studied. In EAP-TLS, the EAP signalling exchange is fully synchronized between the Mi and the AAA server (As). Thus the transmission delay induced by executing full EAP-TLS exchange during the IAA phase is given as

\[ D_{EAP-TLS} = 3T_{M_i,AKD} + T_{AKD,As} + T(n, T_{M_i,As}) \]  

(1)
where $T_{M_1\cdot AKD_1}$ and $T_{(AKD)_\cdot As}$ denote the average transmission delays induced between the $M_1$ and the $AKD_1$ and between the $AKD_1$ and the $As$ respectively. $T(n,T_{M_1\cdot AKD_1})$ denote the function computing the EAP authentication method delay, where $n$ is the number of messages required for executing the EAP method. $3T_{M_1\cdot AKD_1}$ represent the time necessary to complete two EAP starting (EAP-Request/Identity and EAP-Response/Identity) notification messages and EAP finish (EAP-Success) message. Consequently the EAP-TLS method undergoes full exchange when the $M_1$ initially boots up or during handoff [7]. This accordingly induces HAA latency equivalent to the IAA latency in (1) such that

$$D_{IAA}^{(EAP-TLS)} = D_{IAA}^{(EAP-TLS)} + 2T_{M_1\cdot AKD_1},$$

(2)

However in HOISKA, the IAA phase consists of initial join access request/response and authentication joins request/response as already justified. Thus during the initial join access request/response, the $SK_{M_1\cdot AKD_1}$ for the particular $M_1$ is generated and distributed from the $As$ via the serving $AKD_1$. Then the $M_1$ is authenticated by providing both unique $SK_{M_1\cdot AKD_1}$ and the authenticator $\phi_i$ at the serving $AKD_1$. Therefore the delay induced by the IAA phase is

$$D_{IAA}^{(HOISKA)} = D_{IAA}^{(EAP-TLS)} + 2T_{M_1\cdot AKD_1} + 2T_{M_1\cdot AKD_1},$$

(3)

where $D_{IAA}^{(EAP-TLS)}$ and $D_{IAA}^{(HOISKA)}$ are the overall times to execute join Access request/response and authentication joins request/response respectively. However during the HAA phase whenever $M_1$ perform handoff the context transfer and the join authentication request/ response phases are executed. But the $As$ does not synchronize with the $AKD_1$, hence $As$ scalability and massive reduction in authentication exchange signalling at the wireline. Accordingly the delay induced during the HAA phase is expressed as

$$D_{HAA}^{(HOISKA)} = D_{HAA}^{(EAP-TLS)} + 2T_{M_1\cdot AKD_1} + 2T_{M_1\cdot AKD_1},$$

(4)

where $D_{HAA}^{(EAP-TLS)}$, $D_{HAA}^{(EAP-TLS)}$, $D_{HAA}^{(EAP-TLS)}$, $D_{HAA}^{(EAP-TLS)}$, and $D_{HAA}^{(EAP-TLS)}$ are the required times induced during context transfer phase and during leave and join authentication phases at handoff respectively. The assumption is that context transfer phase is proactively performed between the serving $AKD_1$ and the target $AKD_1$. As the $M_1$ reconnect with the target $AKD_1$ both the $M_1$ and $AKD_1$ perform mutual authentication.

B. Numerical and Simulation Analysis

In this section we analyze the performance of HOISKA and EAP-TLS based on the derived delay analysis. We use $n$ (message transmission delay on one hop $T_p=20$ ms) [12], $T_{M_1\cdot AKD_1} = T_p + T_{(AKD)_\cdot As} = T_{(AKD)_\cdot As} = xT_p$, where $x$ denote the number of hops between the $AKD_1$ and the $As/DKD_1$, $x\in[2,4]$, $T_{M_1\cdot AKD_1} = T_{M_1\cdot AKD_1} + 2T_{M_1\cdot AKD_1}$

$T_{n\cdot AKD_1} = T_{AKD_1\cdot AKD_1} - \sqrt{T_{AKD_1\cdot AKD_1} - 4}$

assuming AKD and DKD/As represent the MAG and LMA in [13]. The value of $n$ in EAP-TLS is 4 [7].

The IAA and HAA delays for the underlying schemes are investigated for increasing $n$. The value $n$ increases as the network domain expand. As shown in Fig 1, both the IAA and HAA latencies in EAP-TLS are hugely influenced by increasing $n$ compared to HOISKA. Thus EAP-TLS induces huge authentication exchange signalling at the wireline and wireless parts of the system by synchronizing with the $As$ during the IAA and HAA phases. The $As$ may be intercontinentally far hence adding the more delay factor. EAP-TLS requires fast $As$ and fast link between the $AKD_1$ and the $As$. This impacts the network framework and deployment scenarios. Instead in HOISKA, authentication exchange signalling is less affected by increasing $n$. Thus the $As$ does not participate during the HAA phase hence reducing authentication delay significantly compared to the related art. However transaction with the $As$ is required only during the IAA phase when $M_1$ initially registers with the access network to obtain the root keys for deriving unique session keys at the $AKD_1$ level without involving the $As$. The distributed nature of authentication function in HOISKA enables fault tolerance with $As$ scalability.

V. Security Analysis

The presence of mutual authentication between the $M_1$ and $AKD_1$ prevents impersonation attacks such as Denial of Service attacks. Thus malicious users have no prior knowledge of the long term credential, $AKM_1$, so it cannot derive its unique session key even if it captures the $SK_{AKD_1}$. Therefore malicious users are prevented from penetrating deeply in to the network hence overwhelming the network performance becomes improbable. At each handoff, the $M_1$ derives the new session key per access network along with its new authenticator. This is beneficial because key compromise get localized. The corresponding $AKD_1$ cannot forge to be legitimate for authenticating users because it uses the already derived session key from the trusted $As$ to verify the handoff users. The $AKD_1$ without prior
knowledge of the AKM1, it cannot derive the corresponding session key in SKDL1, hence preventing forgery attacks, man in the middle attacks and redirection attacks. The message transactions in HOISKA include nonces and timestamps which are verified to prevent any alteration attacks. Other interesting factors such as location based authentication using GPS can be incorporated in HOISKA for accurate traceability and identification of the M, since the IAKDi can easily be compromised with a sensor device.

VI. CONCLUSION

In this paper a novel fast and secure handover authentication scheme known as HOISKA is proposed. It takes advantage of decentralizing the authentication functions to the intermediate AKDs such that the domain authenticating server As which may be across the globe does not participate in authentication exchange signalling during user handover. The distributed nature of HOISKA provides user network access control to the security keys for protecting the multicast contents in SMGKM. The presented numerical results showed that HOISKA can provide acceptable performance for real time multicast applications by inducing minimal delays with same level of security compared to the widely deployed EAP-TLS. Thus it can be the suitable authentication method in high speed future wireless networks such as 5G. The verification of the authentication protocol against various attacks using BAN logic is the future work of this paper.

ACKNOWLEDGEMENT

The authors would like to thank MoESD-DTEF (Ministry of Education Skills & Development Planning-Department of Tertiary Education &Financing), Gaborone, Botswana and BIUST (Botswana International University of Science & Technology) for the grant that supported this work.

REFERENCES