Improving Authentication function in wireless mobile multicast communication

T. T. Mapoka, S. J. Shepherd, K. O. Anoh, R. A. Abd-Alhameed
Electrical Engineering and Computer Science
University of Bradford,
Bradford, UK
{T.T.Mapoka; R.A.A.Abd}@bradford.ac.uk

Y. A. S. Dama
Electrical Engineering
An-Najah National University
Nablus, Palestinian

H. M. AlSabbagh
Electrical Engineering
Basra University
Basra, Iraq

***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 Ipad 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 GKMF [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_i and the rows are in encrypted form to securely store the private keys SK_{Mi_AKDi} corresponding to the number of registered M_i under AKD_i for authentication purpose. The AKD_i 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_v where the M_i will visit and each AKD_i 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 M_i to utilize unique session key (SK_{Mi_AKDi}) per cluster during initial access authentication and as the M_i crosses boundaries of various networks during handover it is modified to a new one to ensure key separation per cluster. The SK_{Mi_AKDi} 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 M_i 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 I. Notations and Definitions

Notation	Definition
X	A message X or statement
$X \| \| Y$	Concatenation of X and Y
$E(K, X)$	Function encrypting X with key K
K_{A-B}	Symmetric Key shared between A and B
$A \rightarrow B : X$	Direction of X sent from A to B
ID_A	Identifier that uniquely distinguishes A
T_A	Timestamp generated by A
T_sT_e	Start and End times for the Timestamp
$N_A^{(j)}$	An j^{th} nonce generated by A
ID_{G_i}	Identity of the multicast service group

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_{Mi} and the session key SK_{Mi_AKDi} are both generated and distributed from the AAA Server (As) such as Diameter or RADIUS to the M_i and the AKD_i respectively. Instead, in *J_AccReq/ J_AccRes*, the M_i is authenticated by deriving the SK_{Mi_AKDi} using the obtained AK_{Mi} before entering the target access network.

Step 1: Suppose that the M_i under the serving access network controlled by AKD_i initially registers with the network at boots up. The M_i randomly chooses a nonce $N_{M_i}^{(i)}$ and sends the *IJ_AccReq* message to the current AKD_i, i.e. $M_i \rightarrow AKD_i : IJ\_Acc\mathrm{Re}q(ID_{M_i} \| ID_{G_i} \| ID_{AKD_i} \| N_{M_i}^{(j)})$.

Step 2: The AKD_i after receiving the *IJ_AccReq* it also randomly choose it's nonce $N_{M_i}^{(j)}$ then add it to the *IJ_AccReq* in formation of the Join Access Request message to be sent to the As (*IJ_AccReqAs*). Assuming the security association key $K_{AKDi\text{-}As}$ is already shared. Thus: $AKD_i \rightarrow As : E(K_{AKD_i-As}, IJ\_Acc\mathrm{Re}qAs(ID_{M_i} \| ID_{Gi} \| ID_{AKD_i} \| N_{M_i}^{(j)} \| N_{M_i}^{(k)})$. The AAA server (*As*) also generates the SK_{Mi_AKDi} for the corresponding M_i under AKD_i as $SK_{M_i\text{-}AKD_i} = E(K_{SK_{M_i\text{-}AKD_i}}, K_{M_i\text{-}AKD_i} \| ID_{M_i} \| ID_{G_i} \| T_sT_e)$, where T_sT_e is the set start and end times of the corresponding session key, $K_{M_i\text{-}AKD_i}$ is the key for generating the authenticator during registration at the cluster level. The *As* after detecting that M_i want to initially register in to the underlying system, it notifies the DKD to generates the $SKDL_i$ row specific for the M_i under AKD_i which securely store the generated $SK_{Mi\text{-}AKDi}$ [8]. Note that here we use nonces to prevent replay attacks in the received $SKDL_i$ instead of MAC and the format of the $SKDL_i$ is maintained.

Step 3: The *As* respond with *IJ_AccResAs* to the AKD_i where M_i currently resides as

$$As \rightarrow AKDi : IJ\_Acc\mathrm{Re}sAs(ID_{M_i} \| ID_{G_i} \| SK_{M_i-AKD_i} \| E(AK_{M_i}, K_{M_i-AKD_i} \| ID_{AKDi} \| T_sT_e \| N_{M_i}^{(j)}) \| E(K_{AKDi-As}, K_{SK_{M_i-AKD_i}} \| T_sT_e \| N_{AKD_i}^{(k)})).$$

Step 4: The AKD_i after receiving the *IJ_AccResAs* message, it obtains the $SK_{Mi\text{-}AKDi}$ from the $SKDL_i$ row by decrypting part of the *IJ_AccResAs* message $E(K_{AKDi\text{-}As}, K_{SK_{Mi\text{-}AKDi}} \| T_sT_e \| N_{AKD_i}^{(k)})$ with $K_{AKDi\text{-}As}$.

Step 5: After verifying the $N_{AKD_i}^{(k)}$, it forward part of the *IJ_AccResAs* received message intended for M_i as

$$AKD_i \rightarrow M_i : IJ\_Acc\mathrm{Re}s(ID_{M_i} \| ID_{G_i} \| SK_{M_i-AKD_i} \| E(AK_{M_i}, K_{M_i-AKD_i} \| ID_{AKDi} \| T_sT_e \| N_{M_i}^{(j)})).$$

Step 6: The M_i also obtain its derived $SK_{Mi\text{-}AKDi}$ then extract $K_{Mi\text{-}AKDi}$ by decrypting part of the *IJ_AccRes* with AK_{Mi} after verifying its nonce $N_{M_i}^{(j)}$. The AK_{Mi} is pre-shared key between the M_i and the *As* before network access boot up. Note that the AKD_i is not aware of the AK_{Mi} derivation and the AK_{Mi} is assumed to be stored securely in a tamper proof mobile device smartcard. The M_i now need to derive unique Main session key (**mSK_{Mi_AKDi}**) specific to the access network *i* for communication between the M_i and the AKD_i over the subscription period at the serving network *i*. Thus the M_i derive its Main Session Key (**$mSK_{Mi\text{-}AKDi}$**) as $mSK_{M_i\text{-}AKD_i} = HMAC\text{-}SHA1(AK_{Mi}, (N_{M_i}^{(j+1)} \| ID_{M_i} \| ID_{AKD_i}))$. Then M_i perform the derivation of the authenticator φ_i such that $\varphi_i = E(K_{M_i\text{-}AKD_i}, ID_{M_i} \| mSK_{M_i\text{-}AKD_i} \| T_sT_e \| N_{M_i}^{(j+2)})$.

Step 7: Assuming the underlying multicast system is now in operation, the M_i and the AKD_i should mutually authenticate each other. Thus the M_i now form the join authentication request (*J_AuthReq*) to be sent to the AKD_i as

$Mi ® AKD_i : J\_Auth\mathrm{Re}q(SK_{M_i\text{-}AKD_i} \| \varphi_i$. The AKD_i then obtains the $K_{Mi\text{-}AKDi}$ from the *J_AuthReq* message by decrypting $SK_{Mi\text{-}AKDi}$ with $K_{SK_{M_i\text{-}AKD_i}}$ stored in the $SKDL_i$. $K_{Mi\text{-}AKDi}$ is further used to decrypt the φ_i hence the AKD_i obtains the corresponding **$mSK_{Mi\text{-}AKDi}$** and $N_{M_i}^{(j+2)}$ to successfully warranty M_i access to the network resources. In this context, the resources are the group communication session key shares ($TEK_{i,j}$) for the multi-services subscribed by the M_i which are distributed during the group key distribution phase described in [4].

Step 8: Finally the AKD_i acknowledge the M_i entirely in to the system by sending the join authentication response (*J_AuthRes*) as:

$AKD_i \rightarrow M_i : J\_Auth\mathrm{Re}s(E(K_{M_i-AKD_i}, ID_{Mi} \| N_{Mi}^{(j+2)} + 1 \| TEK_{i,j}))$. Note that the *J_AuthRes* message contains the fresh $TEK_{i,j}$ shares equivalent to the number of services subscribed by the M_i depending on which service group (G_K) the M_i belongs [4]. The M_i on receiving the *J_AuthRes* message, it decrypts it using

$K_{Mi\text{-}AKDi}$ then verify the nonce value $N_{M_i}^{j+2}+1$. If valid, the M_i successfully authenticates the AKD_i to obtain the group communication key shares securely. This concludes the IAA operation in HOISKA hence follows secure transmission of multicast services to the M_i 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 $K_{SKMi\text{-}AKDv}$ for the M_i moving to the target AKD_v. The rows correspond to the number of M_i performing handoff and this prepares the target AKD_i 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 AKD_i and the target AKD_v. *Join authentication request/response phase* which authenticate the handoff user M_i at the target access network *v* controlled by AKD_v before obtaining its service group communication keys. Thus the handover M_i provide the derived unique $SK_{Mi\text{-}AKDv}$ specific for the access network *v* and the new authenticator φ_v before it can be granted access at the target AKD_v. Now suppose M_i reaches the cross boundaries of the AKD_i and the target AKD_v where the signal coverage P_i for AKD_i is lower than that of P_v for AKD_v. Thus the M_i performs handoff from the access network by the AKD_i to the access network by the AKD_v. Assuming also that layer 2 (L2) switch is used to enable fast handover [11]. When the M_i prepares handover at the cross boundaries,

Step 1: The $SKDL_i$ row containing the $K_{SKMi\text{-}AKDv}$ for M_i is securely transferred within the Context Transfer Authentication message (*CtxtTAuthReq*) to the AKD_v as

$$AKD_i ® AKD_v : CtxtTAuth\text{Re}q(E(K_{AKD_i\text{-}AKD_v}, K_{SKM_i\text{-}AKDv} \| IDMi \| T_sT_e \| T_{AKDv} \| KUS_i)).$$

Note that the Key Update Slot (KUS_i) notifier included in C*txtTAuthReq* 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 AKD_v upon receiving the *CtxtTAuthReq*, it obtains the contents of the message by decrypting with $K_{AKDi\text{-}AKDv}$. The AKD_v checks the validity of the contents by verifying T_{AKDv}. The AKD_v similarly securely notify the AKD_i using context Transfer acknowledgement response (*CtxtTAuthRes*) for successfully receiving the content of *CtxtTAuthReq* hence prepares for authentication of the M_i in advance before M_i reaches its coverage. If there are any non-real time packets destined for M_i, the AKD_i also forward them securely to the AKD_v so that they are buffered until M_i rejoins the target access network to prevent further packet loss. We assume that the AKD_v has sufficient memory to store the packets until the M_i arrive.

Step 3: As the M_i is reaches the target access network controlled by AKD_v, it undergoes handover authentication in order to be granted access. Thus the M_i notifies the previous AKD_i about leaving the access network *i* by sending *Leave authentication request* message (*L_AuthReq*) as $M_i ® AKD_i : L\_Auth\text{Re}q(SK_{M_i\text{-}AKD_i} \| \varphi_i)$. The AKD_i verifies $SK_{Mi\text{-}AKDi}$ stored in $SKDL_i$, $N_{M_i}^{(j+2)}$ and the end time for φ_i. The AKD_i finally removes the M_i rows in $SKDL_i$ to reserve more storage space for other joining users. Similarly the M_i removes the key parameters used while residing in the previous AKD_i. The M_i randomly choose a nonce $N_{M_i}^{(j+3)}$ to generate the new $mSK_{M_i\text{-}AKD_v}$ as $mSK_{M_i\text{-}AKD_v} = HMAC\text{-}SHA1(AK_{Mi}, (N_{M_i}^{(j+3)} \| ID_{M_i} \| ID_{AKD_v}))$. hence generating the new authenticator φ_v as

$\varphi_v = E(K_{M_i\text{-}AKD_i}, ID_{M_i} \| mSK_{Mi\text{-}AKDv} \| T_sT_e \| N_{M_i}^{(j+4)})$. Notice that both new $mSK_{M_i\text{-}AKD_v}$ and the new φ_v are unique per access network to ensure key separation per access network as specified in [5]. Thus the M_i does not maintain history of the local keys used in the previously visited networks. This prevents compromises while enhancing M_i key storage. This is what makes HOISKA very unique.

Step 4: After successful derivation of both new $mSK_{M_i\text{-}AKD_v}$ and φ_v provided $P_i \ll P_v$, the M_i then send a *J_AuthReq* to the target AKD_v as: $M_i ® AKD_v : J\_Auth\text{Re}q(SK_{M_i\text{-}AKD_v} \| \varphi_v)$.

Step 5: After receiving the *J_AuthReq* from the M_i, AKD_v decrypt the message using $K_{Mi\text{-}AKDv}$ in $SKDL_v$ then check the validity of the nonce of $N_{M_i}^{(j+4)}+1$. If valid, the AKD_v authenticate M_i successfully. The assumption is that the AKD_v has performed key update process using the received KUS_i from the AKD_i hence key distribution of the $TEK_{v,j}$ for the affected services [4, 5] along with the *J_AuthRes* in step 6.

Step 6: The AKD_v send an acknowledgement response message *J_AuthRes* containing the group communication keys for the services the M_i is subscribed to as $M_i ® AKD_v : J\_Auth\text{Re}s(K_{M_i\text{-}AKD_v}, ID_{Mi} \| N_{M_i}^{(j+4)}+1 \| TEK_{v,j})$. The M_i after receiving the *J_AuthRes* message from the AKD_v, the M_i decrypts it using $K_{Mi\text{-}AKDv}$ it verifies the nonce value $N_{M_i}^{(j+4)}+1$. If valid the M_i also authenticates the AKD_v 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 AKD_v, the M_i receives them along with the services from the SP during the service distribution phase not covered in the paper.

IV. AUTHETICATION 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 M_i and other participating network agents have sufficient capacity to operate cryptographic operations.

A. Authetication Delay Analysis

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

$$D_{IAA}^{(EAP\text{-}TLS)} = 3T_{M_i\text{-}AKD_i} + T_{AKD_i\text{-}As} + T(n, T_{M_i\text{-}As}), \quad (1)$$

where $T_{M_i\text{-}AKD_i}$ and $T_{AKD_i\text{-}As}$ denote the average transmission delays induced between the M_i and the AKD_i and between the AKD_i and the As respectively. $T(n, T_{M_i\text{-}As})$ denote the function computing the EAP authentication method delay, where n is the number of messages required for executing the EAP method. $3T_{M_i\text{-}AKD_i}$ 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_i initially boots up or during handoff [7]. This accordingly induces HAA latency equivalent to the IAA latency in (1) such that

$$D_{IAA}^{(EAP\text{-}TLS)} = D_{HAA}^{(EAP\text{-}TLS)}. \quad (2)$$

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 $SK_{Mi\text{-}AKDi}$ for the particular M_i is generated and distributed from the As via the serving AKD_i. Then the M_i is authenticated by providing both unique $SK_{Mi\text{-}AKDi}$ and the authenticator φ_i at the serving AKD_i. Therefore the delay induced by the IAA phase is

$$D_{IAA}^{(HOISKA)} = D_{AcRR}^{HOISKA} + D_{Auth}^{HOISKA} = 2T_{M_i\text{-}As} + 2T_{M_i\text{-}AKD_i}, \quad (3)$$

where D_{AcRR}^{HOISKA} and $D_{J\_Auth}^{HOISKA}$ are the overall times to execute join Access request/response and join authentication request/response respectively. However during the HAA phase whenever M_i perform handoff the context transfer and the join authentication request/ response phases are executed. But the As does not synchronize with the AKD_i hence As scalability and massive reduction in authentication exchange signalling at the wireline. Accordingly the delay induced during the HAA phase is expressed as

$$D_{HAA}^{(HOISKA)} = D_{CtxtTAuth}^{(HOISKA)} + D_{L\_Auth}^{(HOISKA)} + D_{J\_Auth}^{(HOISKA)} \quad (4)$$
$$= 2T_{AKD_v\text{-}AKD_v} + T_{M_i\text{-}AKD_i} + 2T_{M_i\text{-}AKD_v},$$

where $D_{CtxtAuth}^{HOISKA}$, $D_{L\_Auth}^{HOISKA}$ and $D_{J\_Auth}^{HOISKA}$ 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_i and the target AKD_v. As the M_i reconnect with the target AKDv both the M_i and AKD_i 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_{tr}=20 ms) [12], $T_{M_i\text{-}AKD_i} = T_{tr}, T_{AKD_i\text{-}As} = T_{AKD_i\text{-}As} = xT_{tr}$, where x denote the number of hops between the AKD_i and the As/DKD, x=[2,4], $T_{M_i\text{-}As} = T_{M_i\text{-}AKD_i} + T_{AKD_i\text{-}As}$,

$T_{As\text{-}DKD} = T_{AKD_i\text{-}AKD_v} = \sqrt{T_{AKD_i\text{-}DKD/As}}$ 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 8, 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_i 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_i initially registers with the access network to obtain the root keys for deriving unique session keys at the AKD level without involving the As. The distributed nature of authentication function in HOISKA enables fault tolerance with As scalability.

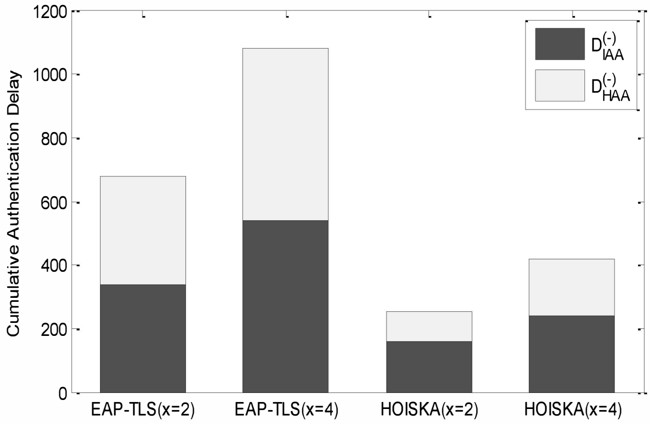

Fig 8.Variances of IAA and HAA delays in each scheme

V. SECURITY ANALYSIS

The presence of mutual authentication between the M_i and AKD_i prevents *impersonation attacks* such as *Denial of Service* attacks. Thus malicious users have no prior knowledge of the long term credential, AK_{Mi}, so it cannot derive its unique session key even if it captures the I_{AKDi}. Therefore malicious users are prevented from penetrating deeply in to the network hence overwhelming the network performance becomes improbable. At each handoff, the M_i derives the new session key per access network along with its new authenticator. This is beneficial because key compromise get localized. The corresponding AKD_i 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_i without prior knowledge of the AK_{Mi} it cannot derive the corresponding session key in $SKDL_i$ 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_i since the I_{AKDi} 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

[1] T. T. Mapoka, "Group Key Management Protocols for Secure Mobile Multicast Communication: A Comprehensive Survey," *International Journal of Computer Applications,* vol. 84, pp. 28-38, December 2013.

[2] Y. Challal and H. Seba, "Group Key Management Protocols: A Novel Taxonomy. ," *Enformatika, International Journal of Information technology,* vol. 2, 2005.

[3] T. T. Mapoka, Y. A. S. Dama, H. M. AlSabbagh, S. J. Shepherd, and R. A. A.-. Alhameed, "Multi-Service Group Key Establishment for Secure Wireless Mobile Multicast Networks " *Journal of Telecommunications* vol. 27, October 2014.

[4] T. Mapoka, T., S. Shepherd, and R. Abd-Alhameed, "A new multiple service key management scheme for secure wireless mobile multicast," *Mobile Computing, IEEE Transactions on,* vol. PP, pp. 1-1, 2014.

[5] T. T. Mapoka, S. Shepherd, R. Abd-Alhameed, and K. O. O. Anoh, "Novel rekeying approach for secure multiple multicast groups over wireless mobile networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*, 2014, pp. 839-844.

[6] L. M. Kiah and K. M. Martin, "Host Mobility Protocol for Secure Group Communication in Wireless Mobile Environments," in *Future Generation Communication and Networking (FGCN 2007)*, 2007, pp. 100-107.

[7] T. Clancy, M. Nakhjiri, V. Narayanan, and L. Dondeti, "Handover key management and re-authentication problem statement," ed: March, 2008.

[8] T. T. Mapoka, S. J. Shepherd, R. Abd-Alhameed, and K. O. O. Anoh, "Efficient authenticated multi-service group key management for secure wireless mobile multicast," in *Future Generation Communication Technology (FGCT), 2014 Third International Conference on*, 2014, pp. 66-71.

[9] S. Gharout, A. Bouabdallah, M. Kellil, and Y. Challal, "Key management with host mobility in dynamic groups," presented at the Proceedings of the 3rd international conference on Security of information and networks, Taganrog, Rostov-on-Don, Russian Federation, 2010.

[10] J. Loughney, M. Nakhjiri, C. Perkin, and R. Koodli, "Context Transfer Protocol (CXTP)," *IETF RFC 4067,* 2005.

[11] H. Yokota, K. Chowdhury, and R. Koodli, "RFC 5949 fast handovers for proxy mobile IPv6," ed: Reston, VA, USA, Internet Society, 2010.

[12] H. Zhou, H. Zhang, and Y. Qin, "An authentication method for proxy mobile IPv6 and performance analysis," *Security and Communication Networks,* vol. 2, pp. 445-454, 2009.

[13] J.-H. Lee, S. Pack, I. You, and T.-M. Chung, "Enabling a paging mechanism in network-based localized mobility management networks," *Journal of Internet Technology,* vol. 10, pp. 463-472, 2009.