JOURNAL OF TELECOMMUNICATIONS, VOL. 27, ISSUE 2, 2014, PP. 43-49; ISSN: 2042-8839

Multi-Service Group Key Establishment for Secure Wireless Mobile Multicast Networks

Trust T. Mapoka, Yousef. A.S. Dama, Haider M. AlSabbagh, Simon J. Shepherd, Raed A. Abd-Alhameed

Abstract—Recently there is high demand in distributing multimedia services over the internet to ubiquitous and computational intelligent mobile subscribers by the service providers (SPs). In this instance, provision of those services must be restricted to authorized subscribers via integration of authentication and group key management (GKM). GKM with diverse group services subscribed dynamically by moving subscribers in wireless networks has been omitted in conventional approaches. However it is expected that significant key management overhead will arise in them due to multi-services co-existing in the same network. In this paper, we propose a scalable decentralized multi-service GKM scheme considering host mobility in wireless environment. In the scheme, authentication of mobile subscribers and key management phases are delegated from the trusted domain key distributor (DKD) to the subgroup controllers known as area key distributors (AKD). The trusted intermediate AKDs can then establish and distribute the service group keys to valid subscribers in a distributed manner using identity-based encryption without involving the domain key distributor (DKD). This alleviates unnecessary delays and possible bottlenecks at the DKD. We show by simulation that the proposed scheme has some unique scalability properties over known schemes in terms of optimized rekeying communication and storage overheads. The security performance studies have shown resilience to various attacks.

Index Terms—Multicast communication; multi-service group key management, wireless mobile multicast networks

- Trust T. Mapoka, Simon J. Shepherd and Raed A. Abd-Alhameed, School of Engineering and Informatics, Bradford University, Bradford, BD7 1DP, UK
- Yousef A.S. Dama; An-Najah National University, Nablus, Palestinian
- Haider M. AlSabbagh; Department of Electrical Engineering, University of Basra, Basra, Iraq

1 INTRODUCTION

The existing GKM protocols for wired approach as in [1] focus on generating keys and rekeying with dynamic group members. They are divided into centralized, decentralized and contributory [1]. Centralized schemes rely on the centralized server known as the DKD which is a single point of failure for key generation and distribution. Contributory scheme allow group members to cooperate for group key establishment without the DKD involvement. Decentralized schemes partition the group into subgroups each controlled by subgroup controllers to equally distribute the key management tasks hence scalability. Work in [2] further categorizes the GKM as common TEK and independent TEK per subgroup approaches depending on the TEK distribution in the framework. Common TEK approaches such as in [3-5] utilize one TEK for all group members and commonly suffer from 1-affect-n phenomenon; thus rekeying of the new TEK disturbs members in the entire network whenever a membership change occurs. Independent TEK per subgroup alleviate the 1-affect-n phenomenon caused by common TEK approaches such as in [6], by enabling each subgroup to manage its own TEK, thus rekeying of the new TEK is localized within the affected subgroup during membership change. However the GKM protocols do not consider rekeying on host mobility on their implementation which is the focus of this paper.

On the other hand, the existing GKM protocols for wireless mobile approach such as [7-10] focus on generating keys and rekeying with dynamic movements of subscribers. The protocols adopt decentralized framework for scalability. Work in [11] also categorized them according to common TEK [7-9] and independent TEK per subgroup [10] approaches as described in [2] to address similar rekeying issues. However, both the GKM approaches address access control in a single service. In these approaches, all the subscribers have same level of access privilege which enables them full access to the subscribed service if the decryption key is valid or deny access for an invalid decryption key. However, multi-service oriented GKM schemes may focus on multilevel access privileges which could complicate key management. Thus mobile subscribers may subscribe to various multiple services while moving and decrypt them with their keys. Several group oriented applications such as video conferencing, pay-per-view sports channels, and multi-stream mobile TV events may co-exist in the same evolving wireless networks. This would require an efficient GKM scheme for securing those service streams.

In this paper, we introduce a new scalable session key distribution list (SKDL) concept to our earlier multi-service GKM scheme for wireless mobile networks introduced in [12]. The new concept offers the following benefits over the previously proposed schemes;

- Move the authentication of individual mobile receivers from the DKD to the area intermediate trusted key distributor AKD. This alleviates un-

necessary delays and possible bottlenecks at the DKD in large distributed multicast network involving several services.
- Reduce the rekeying traffic between the AKDs and DKDs which is replaced by the traffic to the SP, which needs to be notified of each rekeying.
- Allow the intermediate trusted AKDs to establish and securely distribute the service encryption keys for affected services due to host mobility/handoff. This is done in a scalable manner without involving the trusted DKD. In fact rekeying is handled locally to alleviate 1-affect-n phenomenon and reduce communication overheads.

The rest of this paper is organized as follows: Section 2 details the new scalable multi-service concept while Section 3 details the performance and security analysis of the new scheme against the related work. A conclusion is drawn in Section 4.

2 A New Scalable Multi-Service GKM Scheme

In this section, a decentralized multi-service GKM scheme known as scalable multi-service GKM (SMGKM) scheme is presented. The authentication and key management phases of the system are both delegated securely from the trusted DKD to the intermediate AKDs.

2.1 Multi-service system Description

Suppose the mobile subscribers are subscribed to n services provided by the SP denoted as (s_1, s_2… s_n). For simplicity, assume the SP is collocated with the DKD and can act as the DKD. We also assume that the SP has secure channels already established to distribute the services securely to the subscribers.. All mobile subscribers subscribing to the same set of services form a service group (SG) denoted as (G_1, G_2, …G_K) where K is the number of SGs. This means that there can be $K \leq 2^{n-1}$ possible SGs due to overlapping memberships. For example, if the system consists of various related services such as voice (s_1), sports (s_2), movie (s_3), music (s_4), stock quote (s_5) and email (s_6) as illustrated in Fig 1.

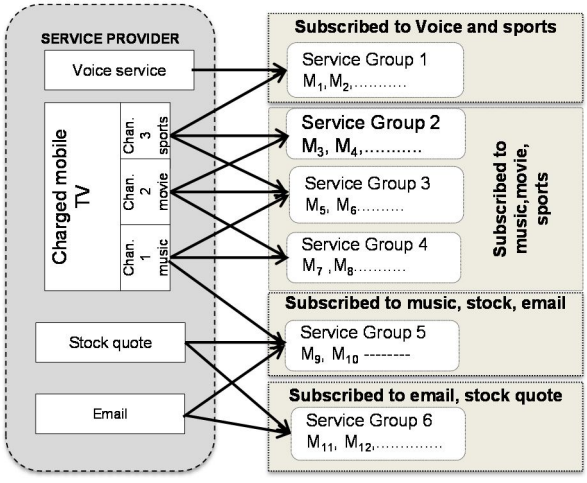


Fig 1 Radiation pattern at yz-plane in the on-on state

The corresponding SGs are G_1 access (s_1, s_2), G_2 access (s_2, s_3), G_3 access (s_2, s_3, s_4), G_4 access (s_3, s_4), G_5 access (s_4, s_5, s_6) and G_6 access (s_5, s_6). In this case subscribers may join, leave or switch service groups dynamically or may dynamically handoff between different administrative domains while maintaining their service groups. This also requires authentication of mobile subscribers whenever they cross different domains.

2.2 System Initial Setup

The entire system assumes trusted computing. It is a decentralized framework similar to [7, 8] with AKDs operating under the jurisdiction of the trusted DKD. The mobile subscribers M_i which are under the jurisdiction of the trusted AKDs access their subscribed services wirelessly. Initially, the DKD setup the necessary system parameters that are delegated to the intermediate AKDs for use during the lifetime of our multi-group service oriented approach. Thus the DKD selects the following parameters:

- Large prime $p = 2q+1$, where q is also primitive,
- An additive cyclic group $\mathbb{G}_1$ and multiplicative cyclic group $\mathbb{G}_2$ both with order p.
- Numbers $P, Q \in \mathbb{G}_1$
- Two strong one way hash functions $H_1 : \mathbb{G}_2 \times \mathbb{Z}_p \rightarrow \mathbb{Z}_p$ and $H_2 : \mathbb{G}_2 \times \mathbb{Z}_p \rightarrow \{0,1\}^*$and $H_3 : \{0,1\}^* \rightarrow \{0,1\}^*$.
- Master secret parameter $s \in \mathbb{Z}_p$ and master public authentication key sP, denoted as (AK_i).

Generally given the cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$, let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map if $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ $\forall$ $P, Q \in \mathbb{G}_1$ and $\forall$ $a, b \in \mathbb{Z}_p$ and non-degenerative if $\hat{e}(P, P) \neq 1$ for the generator $P \in \mathbb{G}_1$. Therefore the DKD uses any efficient and computable non-degenerate bilinear map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ between the two cyclic groups, $\mathbb{G}_1$ and $\mathbb{G}_2$, provided the Diffie Hellman Problem (DHP) in $\mathbb{G}_1$ is infeasible. Other possible practically efficient and computable non degenerate bilinear maps include Weil pairing [13] and Tate pairing [14] on elliptic curve. During the initial registration phase every mobile M_i in the network receives the master secret parameter s and the long term public authentication key (AK_i) common to all registered subscribers. The DKD also derives the M_i private session keys (SK_{Mi_AKDi}) per AKD_i from s and AKD_i identity $ID_{AKDi} \in \mathbb{G}_2$ as sID_{AKDi}. DKD then generate the list called session key distribution list (SKDL) as shown in Fig 2.

#	AKD_ID$_{AKDi}$	$E_{SAi}\{SK_{Mi\_AKDi}\}$	MAC_{SAi}	$E_{SAi}\{$Security parameters$\}$
1	ID_{AKD1}	$E_{SA1}\{SK_{Mi\_AKD1}\}$	$MAC_{SA1}\{ ID_{AKD1} \| \# \| E_{SA1}\{SK_{Mi\_AKD1}\}\}$	$E_{SAi}\{p, s, P, Q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{Z}_p, H_1, H_2, H_3\}$
2	ID_{AKD2}	$E_{SA2}\{SK_{Mi\_AKD2}\}$	$MAC_{SA2}\{ ID_{AKD2} \| \# \| E_{SA2}\{SK_{Mi\_AKD2}\}\}$	
3	ID_{AKD3}	$E_{SA3}\{SK_{Mi\_AKD3}\}$	$MAC_{SA3}\{ ID_{AKD3} \| \# \| E_{SA3}\{SK_{Mi\_AKD3}\}\}$	
n	ID_{AKDi}	$E_{SAi}\{SK_{Mi\_AKDi}\}$	$MAC_{SAi}\{ ID_{AKDi} \| \# \| E_{SAi}\{SK_{Mi\_AKDi}\}\}$	

Fig. 2. Session Key Distribution List

The list has the following new features:

- It is specific to the AKD_i and its 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 MAC to prevent replay attacks
- It include 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 DKD which are also delegated securely to the AKDs for group key establishment
- It can be used to securely store accounting information at the cluster level before it is securely sent to the SP for further processing.

We assume that the AKD_i-DKD communication link is also protected using shared pairwise security association key (SA_i). The SA_i is used by the trusted DKD to securely push the $SKDL_i$ rows to the corresponding AKD_i which the AKD_i verifies. Finally the AKD_i securely retrieve the delegated DKD chosen security parameters and M_i derived SK_{Mi_AKDi} to proceed with the authentication and group key establishment phases without involving the DKD hence DKD scalability and no bottlenecks. After the system initial setup phase, each cluster AKD_i manages the service group keys independently without the centralized DKD to alleviate 1-affect-n phenomenon [2]. Henceforth, every joining M_i is assumed to be tamper proof and receives AK_i at offline prior to joining the network. M_i can also subscribe to any service group G_K they wish after successful registration.

2.3 Service Group Key distribution

To provide access control to multi-service groups is more complicated than in a single service. Therefore the AKD_i needs to provide an efficient access control mechanism to manage multi-service groups. After successful registration of subscribers, the AKD_i generate $m \times n$ accessibility matrix ($A_{i,j}$) such that:

$$A_{i,j} = \begin{cases} 1, \text{ if } M_i \text{ access service } s_n \\ 0, \quad \text{otherwise} \end{cases}$$

where i denote the rows for subscribers in G_K ($1 \le i \le m$) and j denote the columns for the subscribed service j, ($1 \le j \le n$).

The AKD_i securely retrieve the delegated chosen security parameters from the distributed $SKDL_i$ to proceed with service group keys establishment which are then distributed (broadcasted) to the registered M_i. Let the service group key share be $TEK_{i,j}$ for encrypting service j at cluster i (C_i). The $TEK_{i,j}$ share is shared among the SP and $M_i \in G_K$ subscribed to service j under AKD_i. Let K_{GK} denote the service group keyset containing the aggregated group key shares $\{TEK_{i,1}, TEK_{i,2}, \ldots, TEK_{i,j}\}$ for services in G_K. Thus subscribers under the same G_K would receive the aggregated $TEK_{i,j}$ shares for their subscribed services. To distribute the $TEK_{i,j}$ shares, the AKD_i initially adopt the qualified bilinear pair approach described fully in [15]. The pair $\left(x_k^i, x_k\right)$ is said to be a qualified pair if $x_k^i x_k = x_k^i \bmod p$.

Proof:

Let the received prime parameter p be decomposed as: $p = p_1^{a_1} + p_2^{a_2} + p_3^{a_3} \ldots p_n^{a_n}$, where p_k and a_k are decomposed primes and integers respectively for k=[1,n], $p_k \neq p^i$ and $k \neq k^i$. If $x_k^i = p_k^{a_k}$ and $x_k = \prod_{j \neq k} p_j^{a_j} + 1$, then pair (x_k, x^i) satisfies to be a qualified pair.

The AKD_i can now distribute $TEK_{i,j}$ shares as follows:

1. $\forall M_i \in C_i \cap G_K$, AKD_i choose m qualified pairs (x_k, x^i),
2. Compute $x = \prod_k x_k^i$ where $m \in G_K \cap C_i$ according to the accessibility matrix $A_{i,j}$, for ($1 \le i \le m$).
3. Compute the public parameter xP and the private key $x_k sID_{AKDi}$ $\forall M_i \in SKDL_i \cap G_K$ and verify if the received SK_{Mi_AKDi} matches the private key.
4. Choose random $\rho \in \mathbb{G}_2$ and compute $r = H_1(\rho, K_{G_K})$
5. Broadcast the ciphertext $\langle U, V, W \rangle$ to $\forall M_i \in G_K$ as
6. $\left\langle rxP, K_{G_K} \oplus H_2(\rho), \rho \hat{e}(xP, sID_{AKD_i})^r \right\rangle$

When $\forall M_i \in C_i \cap G_K$ receives the ciphertext $\langle U,V,W \rangle$, it can obtain the K_{GK} as follows

1. Compute $\hat{e}$ (U, $x_k sID_{AKDi}$) = $\hat{e}$ (rxP, $x_k sID_{AKDi}$) which reduces to $(\hat{e}(P, ID_{AKD_i})^{rxx_k s} = (\hat{e}(P, ID_{AKD_i})^{rxs}$.
2. Compute $W \times (\hat{e}(P, ID_{AKD_i})^{-rxs} = \rho$.
3. Compute $V \oplus H_2(\rho) = K_{G_K} = \{TEK_{i,1}, \ldots, TEK_{i,j}\}$.
4. Compute $r' = H_1(\rho, K_{G_K})$ to verify U. $M_i \in G_K$ will accept K_{G_K} if $r'xP = U$, otherwise deny.

Note that $M_z \notin G_K$ cannot obtain the broadcasted message from the AKD_i since it is restricted from the qualified pair, i.e. $xx_z \neq x \bmod p$. Therefore the K_{GK} is delivered to the subscribers using qualified pairs of legitimate members M_i in G_K. Offline $M_i \in G_K$ can still receive the broadcasted message anytime during reconnection provided it is still a valid subscriber. Most of the computation is squeezed into the AKDs with reliable power provision and subscribers only perform 4 computations on initial service group keys establishment. However during their subscription period in the network, they perform fewer computations to preserve power consumption.

2.4 Member handoff with backward secrecy

During the lifetime of system, subscribers may dynamically move between homogeneous/heterogeneous clusters. Handover subscribers are considered as leaving the old cluster i (C_i) controlled by AKD_i followed by join at the target cluster v (C_v) controlled by AKD_v. This requires backward secrecy [8] to be guaranteed for affected services at C_v. Certainly forward secrecy [8] is pointless at C_i because subscribers maintain active sessions on handoff. We assume the context transfer (CXTP) [16] and media independent handover MIH [17] protocols are already installed at the AKDs for accelerating handoff. In this case a moving subscriber undergoes authentication phase on reconnection at the target C_v before the AKD_v generate and distribute the affected service group keys. This implies secure and seamless connectivity on handoff. However, both the phases are handled at the cluster level without involving the DKD to prevent bottlenecks, rejoin and rekeying latencies. Our system adopt a multi-service

slot based rekeying strategy of [18]. The strategy uses key update slots (KUS) of size *l*-bits corresponding to the number of services provided by the SP. Each slot can accommodate a maximum of 2^l subscribers.

On handoff, a moving subscriber M_i detects low power signal strength (P_i) from AKD_i and high power signal strength (P_v) from the target AKD_v, (i.e. $P_i \ll P_v$). The M_i sends a *move_notify* message encrypted under its derived SK_{Mi_AKDi} to the current AKD_i. The AKD_i authenticates $M_i \in G_K$ against its SK_{Mi_AKDi} stored in $SKDL_i$. If the M_i derived SK_{Mi_AKDi} = DKD derived SK_{Mi_AKDi} stored in the $SKDL_i$ then the AKD_i decrypt the notification message successfully. The AKD_i determines a set of affected services subscribed by M_i in G_K hence requiring key update. The AKD_i generate the KUS notifier for the affected services to initiate handover process. Both the KUS notifier and the M_i $SKDL_v$ rows are forwarded to the target AKD_v via AKD_i-AKD_v link using CXTP protocol. Note that the AKDi deletes the M_i rows in $SKDL_i$ after complete handoff. The received KUS notifies the AKD_v about the service keys in K_{GK} requiring key update before M_i arrives. The AKD_v initially perform integrity checks on the received $SKDL_v$ row using MAC before adding it to its existing list ($SKDL_v$). Thus AKD_v now perform service group key update for the affected services progressively as follows:

1. Select a new qualified pair (x^i, x_i) for the joining $M_i \in G_K$.
2. Compute $x^i = (x^i, x_i)$ where $x = \prod_k x_k^i$ $\forall_{Old} M_v \in C_v \cap G_K$, and set the parameter x^iP public.
3. Update the affected service group keys in K_{G_K} by $K_{G_K}^i = H_3(K_{G_K})$, where $K_{G_K}^i = \{TEK_{v,1}^i,...,TEK_{v,j}^i\}$.
4. Wait until M_i join notification message is received.

Whenever M_i detects the ID_{AKDv} from the AKD_v strong signal coverage, the following steps occur progressively:

1. M_i derives the private key SK_{Mi_AKDv} specific to the target C_v as sID_{AKDv}. KDF such as SHA1[19] can be used for deriving the private key.
2. M_i automatically deletes the stored service group keys in K_{GK} and the private SK_{Mi_AKDi} previously used at C_i. This actually creates space for accommodating the security keys in C_v.
3. Send a move_notify message encrypted under the derived SK_{Mi_AKDv} to AKD_v.
4. AKD_v verify the derived SK_{Mi_AKDv} by comparing it with the stored $SK_{Mi\_AKDv} \in SKDL_v$ then decrypt the message if they match otherwise deny M_i into C_v.
5. AKD_v retrieve the service group keys in K_{GK} subscribed by the M_i then distribute the updated version $K_{G_K}^i$ to the $M_i \in G_K$ including $M_v \in G_K$.
6. The AKD_v also send the KUS notifier to the DKD/SP encrypted under SA_v for updating the affected services keys after T_{update}.

However the updated service group key versions in $K_{G_K}^i$ can be distributed to the subscribers using either *pairwise* or *LKH rekeying* approaches at the cluster level:

In *pairwise rekeying approach* [20], all the subscribers in G_K share common set of $TEK_{v,j}$ shares. Whenever M_i reconnect at the target C_v, the AKD_v multicast the new $TEK_{v,j}$ shares in $K_{G_K}^i$ for the affected services (encrypted with the old $TEK_{v,j}$ in K_{GK}) to the existing members $M_v \in C_v$ and unicast the new $TEK_{v,j}$ shares in $K_{G_K}^i$ along with the x_i-sID_{AKDv} to the new joining $M_i \in C_v \cap C_i$ from C_i encrypted by M_i derived secret key (SK_{Mi_AKDv}). Thus rekeying overhead (RO_T) at the target cluster v when M_i joins C_v gives two messages to ensure backward secrecy. However for N subscribers joining C_v with backward secrecy while participating in S services gives RO_T of $O(N)+S$.

In *LKH rekeying approach* [5], a full balanced key graph tree of degree d is used to approximate the rekeying overhead at any time. For join operation at the target C_v, the AKD_v where the join occurs update each key that is on the path from the leaf that represents the secret key/individual key associated to the new joining M_i to the root of the tree which denotes the $TEK_{v,j}$ share. Consequently, each key from the leaf of the new member to the root is transmitted twice, i.e. transmitted to the new joining M_i through unicast encrypted under the child key known to the new M_i, and via multicast to the members that share the node encrypted under its old version. Thus, the rekeying messages using the LKH with branching factor d under cluster v gives $O(\log_d(N))+S$.

The rekeying process of the multi-service group system is pictorially presented in [12]. It is important to note that the qualified pair selection for subscribers is maintained by the AKDs until subscriber M_i is revoked or handoff to another C_v. Subscribers M_i are restricted from accessing the service keys when it re-joins at each cluster if the private key (SK_{Mi_AKDi}) is invalid. Forward secrecy is not necessary because M_i deletes the security keys used in the previously visited clusters and M_i maintains its services on move.

3 Performance And Security Analysis

This section analyses our scheme in terms performance and security. In addition to our rekeying transmission overhead at the core network discussed in [12], the performance is also analyzed through numerical and simulation analyses in terms of rekeying communication overheads and memory consumption requirements in the introduced SKDL concept introduced. Finally the security analysis considers all types impossible attacks in our system with regard to the introduced SKDL.

3.1 Rekeying Communication Overhead

This overhead corresponds to the expected amount of rekeying messages from unicast or multicast transmissions at the cluster level when *N*-subscribers perform handoff while maintaining *S*-active multicast sessions. Table 1 and 2 compares the communication overheads of our scheme with the existing work for the two rekeying approaches (*pairwise* and *LKH*) discussed in section 2.4 respectively. Simulation was carried out for the corre-

sponding rekeying approaches in Fig. 2a) and b) to compare the rekeying communication overheads of the concerned schemes. The simulation parameters used include N=n= 1 and 10 for S=7 services.

It can clearly be seen that BR induces high number of rekeying signalling messages than others because the service keys and local area keys are updated independently in both clusters on member handoff. The IR reduces the need to rekey service keys but triggers local area key rekey only in both clusters. To further reduce communication overheads from IR, both the GKMF [8] and Kellil et al [9] schemes adopt DR [7] strategy by introducing the use of mobility list as to record handover members such that the previous C_i induces null rekeying overhead on M_i handoff.

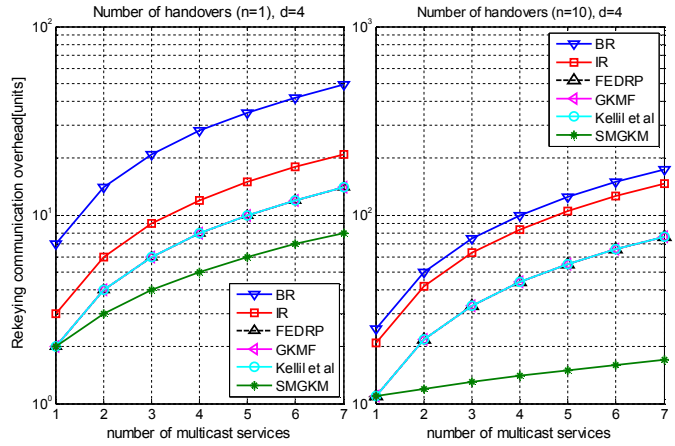
TABLE 1
PAIRWISE REKEYING APPROACH COMMUNICATION OVERHEADS

Reference frameworks	Communication overhead at		Total Overhead
	cluster i	cluster v	
BR [7]	$S\times$[O(N)+1+1]	$S\times$[O(N)+1+1+1]	2SO(N)+5S
IR [7]	$S\times$[O(N)]	$S\times$[O(N)+1]	2SO(N)+S
FEDRP [7]	0	$S\times$[O(N)+1]	SO(N)+S
GKMF [8]	0	$S\times$[O(N)+1]	SO(N)+S
Kellil et al[9]	0	$S\times$[O(N)+1]	SO(N)+S
SMGKM	0	O(N)+S	O(N)+S

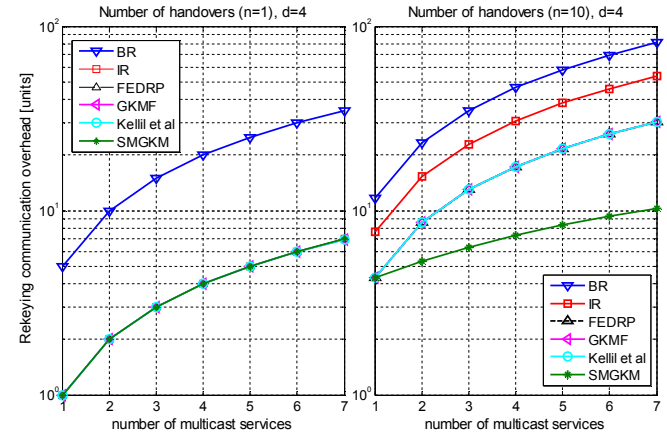
TABLE 2
LKH REKEYING APPROACH COMMUNICATION OVERHEADS

Reference frameworks		Total Communication Overhead
DeCleene et al[7]	BR	2SO($\log_d(N)$)+5S
	IR	2SO($\log_d(N)$)+S
	FEDRP	SO($\log_d(N)$)+S
GKMF[8]		SO($\log_d(N)$)+S
Kellil et al [9]		SO($\log_d(N)$)+S
SMGKM		O($\log_d(N)$)+S

This actually improves the bandwidth efficiency of the system while satisfying the backward secrecy requirement of services at the target C_v. However the rekeying communication overhead is reduced at the expense of storage complexity at the resource constraint mobile receiver as discussed in [12]. Thus a handover subscriber need to maintain the local area keys for the previously visited clusters to reduce the need for rekeying on return period provided no rekey has occurred on M_i. However whenever the high mobility subscribers decide to leave after visiting multiple clusters rekeying is triggered in all the previously visited clusters hence inducing significant communication overheads hence enormous bandwidth usage due to repeated rekeying. Since data transmission cannot resume until all visited clusters have been rekeyed with new service keys. Thus FEDRP, GKMF and Kellil et al stay offline longer than SMGKM. In this case for N multiple handoffs participating in S services; conventional protocols incur huge communication overhead and huge delays in obtaining the services. If the service key shares require key update, the rekeying overhead in conventional approaches becomes substantial due to 1-affect-n phenomenon. Therefore adopting independent service group keys per cluster is the ultimate solution to reduce rekeying overheads hence delays since rekeying gets localized. Additionally the SKDL concept introduced also prevents repeated rekeying caused by frequent handoffs which finally leave after visiting multiple clusters.



a) Using Pairwise rekeying approach



b) Using Pairiwise rekeying approach

Fig.3 Rekeyin communication overheads

3.2 SKDL Memory Consumption Requirement

The SKDL storage capacity in the AKD is determined by the number of M_i the AKD_i is currently serving and number of rows for the target AKD_v the M_i will visit. The format of the SKDL can be determined by the size of the *MAC*, SK_{Mi_AKDi} and the ID_{AKDi}. Assuming that the SKDL uses common initialization vector (IV) with non-repeating values in rows for semantic security and assuming encryption and security parameters does not supplement to the SKDL size. Therefore total SKDL storage requirement can be computed as

$$SKDL_i^{Memory} = sf(\text{IV}) + A*SK_{Mi\_AKDi}*M_i*(sf(SK_{Mi\_AKDi}) + sf(ID_{AKDi}))$$

where $SKDL_i^{Memory}$ denote the storage capacity of the SKDL in bytes; A denote the maximum number of SKDL rows which is usually equivalent to the number of M_i under AKDi; $sf()$ denote the function returning the size in bits of the parameter in brackets. Now specifying SMGKM to utilize the parameters, MAC (e.g SHA1)=160 bits, AES Key (SK_{Mi_AKDi}) = 128 bits, ID_{AKDi} for IPv4 and IPv6 networks= 32 bits and 128 bits respectively. It can be

deduced that one SKDL row size requires 40 bytes and 52 bytes for the network supporting IPv4 and IPv6 respectively.

The SKDL with 28 AKDs would require storage of 1120 bytes and 1456 bytes for IPv4 and IPv6 network respectively which is < 1500 bytes packet of the largest allowed Maximum Transmission Unit (MTU) by Ethernetv2 at the network layer without fragmentation [21]. The AKDi supporting IPv4 and IPv6 addresses with 50 keys and 25 M_i under it will occupy a total memory of 50kB and 65kB respectively for SKDL only. Thus the SKDL perspective is more applicable for distributing the SK_{Mi_AKDi} synchronously to the specific AKDs especially in congested areas like the city center where the AKDs and M_i are considerable huge in number without key request from the DKD for authentication and key distribution. Therefore this optimizes the signalling load between the AKD and DKD and authentication delays in the convectional schemes. As shown in Fig. 4, if SMGKM support IPv6 enabled network, this would increase SKDL memory consumption by 30% than IPv4 network due to its lengthy address.

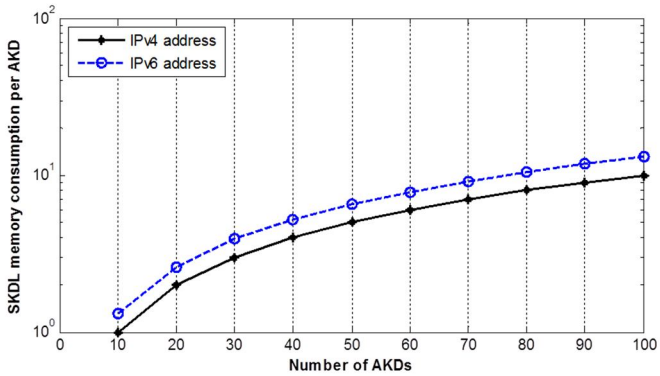


Fig. 4. SKDL memory consumption with increasing AKDs

However the SKDL with SK_{Mi_AKDi} keys for 400 target AKDs would need memory of about 16kB per M_i for IPv4 and 20.4kB per M_i for IPv6. The AKDs are only interested in obtaining their rows equivalent to the number of M_i they are currently serving. Since increasing the memory requirement in the AKD reduces signaling load due to key request from the DKD hence DKD scalability. Conclusively SKDL proposal is a signaling optimizer of the conventional key request schemes (DeCleene et al, GKMF and Kellil et al). Thus our scheme scales better when more target AKDs are included in the SKDL rows assuming the mobility pattern of the M_i and network conditions remains stable.

3.2 Security Analysis

The service group key distribution in our scheme uses identity based encryption which is secure against *outside probabilistic polynomial time attacks*. Thus to get the updated K_{GK} from the broadcasted ciphertext <*U*, *V*, *W*>, the outside adversaries must compute ρ without the secret key of the AKD_i. Due to cryptographically independent keys per cluster key compromises within the cluster get localized in our scheme. *Eavesdropping opportunities* are impossible in our system because backward secrecy requirement is achieved by updating the affected service keys in K_{GK} which are delivered to valid subscribers using their qualified pairs by the AKD_i. Thus the reconnecting subscriber cannot discover the preceding service group key shares in K_{GK} before it joins due to the one-way property of H_3. *Impersonation attacks* are also intolerable due to mutual authentication between M_i and the AKDi on every handoff. Each member is verified against the already DKD derived SK_{Mi_AKDi} in the $SKDL_i$ before accessing the fresh versions of the service group keys using the secret key SK_{Mi_AKDi} which is bonded to the specific AKD_i identity. Thus with mutual authentication over the air interface, M_i can check if a network can be trusted before entering hence preventing denial of service attacks (*DoS*). The AKD_i cannot forge to be legitimate because it is not aware of the M_i public key AK_i. It only receives the already derived SK_{Mi_AKDi} for M_i via secure $SKDL_i$ rows from the trusted DKD which it uses to mutually authenticate the subscribers at joining. The security parameters for deriving the service keys are also securely stored and delivered through the $SKDL_i$. The $SKDL_i$ rows are kept in encrypted form and integrity protected using MAC to prevent any *replay attacks*. Encryption/decryption operations of the $SKDL_i$ rows are specific to the AKD_i to prevent *contributory attacks*. If the introduced $SKDL_i$ is implemented in an external secure hardware chip like the trusted module, it will be practically more infeasible for the adversary to obtain the SK_{Mi_AKDi} by just examining the $SKDL_i$ rows between the AKD_i-AKD_i links. The $SKDL_i$ concept also inhibits *location based attacks* and *redirection attacks* are also impossible due to the encrypted rows by the trusted DKD. *Physical node capture attacks* are impossible because the subscribers are assumed to be tamper proof and the adversary cannot derive the private key SK_{Mi_AKDi} without prior knowledge of the AK_i.

4 Conclusion

In this paper we presented a new decentralized multi-service GKM scheme for managing service group keys independently. The trusted DKD securely delegated the authentication and the key management security parameters to the intermediate trusted AKDs during initial registration setup using secure SKDL concept. The key distribution phase during rekeying on handoff was offloaded to the AKD level after successfully authenticating subscribers who dynamically change point of attachment to the network domains. This has massively reduced performance hurdles such as authentication delays, 1-affect-n phenomenon in the entire network and single point of failures faced by the DKD in the conventional schemes. The performance evaluation and simulation results showed that our scheme is adaptive to multi-services and outperforms the existing schemes in terms of communication overheads while providing resilience to various attacks. The SKDL concept introduced was found to be very useful for accelerating handoff and provided DKD scalability. The concept can be more applicable in congested mobile environments like the city center where multiple

handoffs may occur with reduced rekeying transmissions over the core network. Thus SMGKM can be used for securely transmitting several applications with different access privileges in the same network with less bandwidth requirement hence benefiting the future generation of communication technologies.

ACKNOWLEDGE MENT

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] S. Rafaeli and D. Hutchison, " A Survey of Key Management for Secure Group Communication," *ACM Computing Surveys,* vol. 35, pp. 309–329, September 2003 2003.

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

[3] H. Harney and C. Muckenhirn, " Group Key Management Protocol (GKMP) Specification," RFC 2094,July 1997.

[4] T. Hardjono, B. Cain, and I. Monga, "Intra-domain Group Key Management for Multicast Security," IETF Internet draft,September 2000.

[5] W. Chung Kei, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *Networking, IEEE/ACM Transactions on,* vol. 8, pp. 16-30, 2000.

[6] S. Mittra, "Iolus: a framework for scalable secure multicasting," *SIGCOMM Comput. Commun. Rev.*, vol. 27, pp. 277-288, 1997.

[7] B. DeCleene, L. Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, *et al.*, "Secure group communications for wireless networks," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, 2001, pp. 113-117 vol.1.

[8] 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.

[9] M. Kellil, Olivereau, J. C. A., and P. Janneteau, "Rekeying in secure mobile multicast communications," *United States Patent Application Publications,* US 2007/ 0143600 A1 2007.

[10] 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.

[11] 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.

[12] 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 *IEEE 10th International Conference on Wireless Communications and Mobile Computing (IWCMC)*, Nicosia, Cyprus, 2014, pp. 839-844.

[13] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology – CRYPTO 2001*, 2001, pp. 213-229.

[14] S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing," in *Algorithmic number theory*, ed: Springer, 2002, pp. 324-337.

[15] L. Wang and C.-K. Wu, "Efficient identity-based multicast scheme from bilinear pairing," *IEE Proceedings-Communications,* vol. 152, pp. 877-882, 2005.

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

[17] V. Sharma, A. Agarwal, and M. A. Qadeer, "Media Independent Handover (IEEE 802.21): Framework for Next Generation Vertical Handover Protocols," in *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, 2011, pp. 507-511.

[18] T. T. Mapoka;, S. Shepherd;, R. Abd-Alhameed;, and K. O. O. Anoh;, "Novel Rekeying approach for multiple multicast groups over wireless mobile networks," in *10th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)* 2014, p. (Accepted).

[19] S. K. and and S. Frankel, "RFC4868: Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec " *IETF RFC 4868,* May 2007.

[20] D. Wallner, E. Harder, and R. Agee, "Key Management for Multicast : Issues and Architecture," National Security Agency, RFC 2627,June 1999.

[21] D. Murray, T. Koziniec, K. Lee, and M. Dixon, "Large MTUs and internet performance," in *High Performance Switching and Routing (HPSR), 2012 IEEE 13th International Conference on*, 2012, pp. 82-87.