J Indian Inst Sci, Mar.–Apr 2000. **80**, 113–127

An approach for secure multicasting in mobile IP networks

VIJAY VARADHARAJAN, RAJAN SHANKARAN AND MICHAEL HITCHENS
Distributed System and Network Security Research, University of Western Sydney, Australia
vijay@cit nepean.uws edu.au

Abstract

There is a considerable interest in the area of mobility with the advent of powerful portable computing devices such as laptops and other information appliances These enable a user to access a service from anywhere at any time Such nomadic computing poses several challenges in multicasting and security. We first consider a framework that has been proposed by Acharya *et al.* [Acharya, A , Bakie, A. and Badrinath, B. R. *IP multicast extensions for mobile internet working*, Rutger DCS Technical Report, LCSR-TR_243.] for multicasting in mobile IP networks. In this paper, we extend this framework to support a secure multicasting service. We describe secure schemes for a mobile host to initiate, join and leave a multicast group We also discuss the secure movement of mobile hosts in intra and inter campus environments.

Keywords: Mobile IP networks, secure multicasting, nomadic computing.

1. Introduction

Multicast is becoming important as it acts as a key enabler for new applications over both private and public networks. This can be in the form of one to many or many to many communications. This approach contrasts with unicast communications where a message is transmitted to a single recipient and with broadcast communications where a message is transmitted to all the recipients. Multicasting networks imply that there are provisions in the network infrastructure to support multicast. Multicast applications use multicast network services as an underlying enabling service. Multicasting is significant because it enables applications to scale, thereby offering service to many users without overloading network and server resources. There are many applications (both real time and non-real time) that benefit from multicasting. These include real-time applications such as video conferencing and Internet audio and non-real time services such as software distribution and database replication.

Although IP multicast is the most common technology that is usually considered within the context of multicast networking, in general, two fundamental multicast networking techniques exist, namely, link- and network layer multicasts. LANs have been historically shared media with connectionless service with three forms of MAC (medium-access control) addresses for uni-, multi- and broadcast. Recently, provisions have been made for supporting multicast in frame relay and ATM networks. Perhaps the main issue with multicasting in wide area link layer has been to do with provisioning (except possibly in ATM). The multicast group is set up statically by the network provider and is changed only by a reconfiguration of the network.

Network-layer multicast solutions based on IP have a major advantage in that group setup and tear down are dynamic. In this case, the routers in the multicast network need to support multicast routing using a suitable protocol. There exist a variety of multicast routing protocols. Typically, a host informs the nearest router supporting multicast IP of its membership in the multicast groups. This is done using the Internet Group Management Protocol (IGMP).[1] IP multicast is a best-effort datagram service, which implies that such a service does not guarantee that its intended recipients will receive the datagram; furthermore, the ordering of the datagrams may change while in transit. IGMP provides a means by which a host can join or leave a group at any time in a dynamic way. There is no group owner at the IP level and joining the group is only a routing issue. Current versions of IGMP (versions 1 and 2) do not incorporate security features. Also, multicast applications run on UDP that is an unreliable, connectionless transport service. Of course, other protocols such as the Real-Time Transport Protocol (RTP) have been developed recently,[2] which support multicasting at the transport level. RTP facilitates end-to-end delivery service for real-time data. However, most of the multicast implementations run over UDP (as the common TCP only provides unicast services).

Though in a general sense, security threats in multicast communications are similar to unicast ones, there are certain security aspects that are unique to multicasting. In particular, the characteristics of multicasting can be very diverse implying that there is unlikely to be a unique security solution. A key aspect is the creation and deletion of groups and members in these groups, in a secure manner. The group size can vary from several tens of members in a small discussion group to thousands of members say in a virtual conference. In some cases, the group membership can be static and in other, members may join at different times whereas in others members may join and leave at different times. The lifetime of a group can also vary from several minutes to days or even an unbounded amount of time. The number of senders can also change from a single sender to several members to all members of the group. Finally, the type of traffic (e.g. real time) as well as the volume of traffic can be different for different multicasting applications.

Some of the main security concerns in multicast communications are concerned with the following:

- *Secure group communication*: This is concerned with ensuring that only legal members of the group have access to communication related to that group. In other words, non-members must not be able to eavesdrop on multicast traffic. This requires the presence of a confidentiality service for the multicast group. Integrity service is also required to assure that a message has not been illegally altered in transit. The members of the group should be able to detect such illegal modifications.
- *Group and member authentication*: The main issues here are concerned with the authentication of group and members of the group. These are often intricately linked with key distribution and management as the provision of authentication service is often based on possession of certain keys. Hence the problem of secure key distribution amongst the members of the group becomes a vital concern. It is necessary to provide mechanisms for revoking memberships of those who leave and for registering those who wish to join new. This in turn implies that that there must be some secure mechanisms for secure key generation, distribution and revocation as the membership of the

group changes. There may also be a need for periodically refreshing the key for the group. Another related issue is the need for individual member authentication in addition to the group authentication. For instance, under certain circumstances, it may be necessary to determine which member of the group performed a certain action (e.g. sent a message).

- *Multicast group management*: Here the main issues are concerned with who can join the group and who and how one decides who can join the group. This is part of access control. More generally for any operation that changes the group structure, these issues need to be addressed.

- There should also be appropriate mechanisms to counteract the denial of service attack. Maintaining service availability against malicious attack is ever more challenging in a multicast setting, as clogging attacks are easier to mount. We do not consider this service availability issue in this paper.

The introduction of mobility further aggravates these security problems. First, multicasting in a TCP/IP network with mobile hosts itself raises technical issues for which there are no widely accepted solutions. Secondly, the increasing demand by mobile users to access services and applications over the network from anywhere and at any time introduces further challenges. For instance, mobile users are increasingly using services such as weather reports, airline information and stock market details. Multimedia applications such as videoconferencing will also have their share of mobility in them. Therefore it is clear that there is a need to provide secure multicasting service for such applications involving mobile hosts. Over the recent years, there have been several proposals for providing mobility for hosts. These include the Columbia approach[3] and Sony's approach[4] and IETF Mobile IP.[5] In this paper, we only consider the Columbia scheme (and its extension, Acharya *et al.*[6]) and develop a secure multicasting service for mobile hosts. In another paper, currently under preparation, we address secure multicasting for IETF Mobile IP. The solution provided by the Columbia scheme supports mobility in a campus environment. It uses a group of cooperating routers known as mobile support routers. The design is interoperable with the TCP/IP protocol stack as it maintains the existing functionality of Internet services. The paper is organized as follows. Section 2 describes the Columbia approach and considers the multicast extensions to it. In section 3, we extend the Columbia model and consider the design and integration of multicast security services. Finally, Section 4 gives our conclusions.

2. Columbia approach

In this section, we shall first describe the Columbia mobile IP architecture. We then highlight some of the issues related to multicasting that arise in this environment. We describe the design choices and the solution used to provide transparent multicast services to mobile hosts. For further details regarding this architecture, the reader is referred to Acharya *et al.*[6]

2.1. *Mobile IP-based Columbia network architecture*

In this mobile IP based model, a virtual subnet is created using a set of cooperating mobile support routers called MSRs which act as gateways between the real wired subnet and the mo-

bile subnet. This set of MSRs is referred to as a campus. A campus is a collection of physical networks called cells that are under a single administrative domain. For instance, this could be a business or a university network. Each MSR supports one or more cells. Since any mobile host may be in any cell, all cells have the same subnet number. In this architecture the mobile subnet really comprises many unconnected physical segments called the cells. In order to make this partitioned network appear as a single subnet, the MSRs exchange information about which mobile hosts are where, and tunnel datagrams between them when they are required to route a datagram destined for a mobile host in another MSR's cell. Within a campus, a mobile host's (MH) address remains unchanged. This implies that mobile hosts move within a campus without changing their logical IP address that is based on the subnet address. This makes the movement of mobile hosts transparent within the campus. MSRs within a campus exchange information about the current location of a mobile host, and they tunnel datagrams between them for host-to-host communication. IP datagrams from one mobile host (MH) to another in the same cell are processed locally.

Unlike the IETF mobile IP[5] proposal, there is no concept of a specific home agent in the Columbia mobile IP proposal. Instead, a set of MSRs that cover the mobile subnet within a campus maintain the binding information. If binding information is not valid, search is used by MSRs to find an up-to-date binding. Also, the binding information stored at the MSRs periodically expires unless it is refreshed by a packet or by the explicit search mechanism. The movements of a mobile host can be of two types: intra- and inter-campus moves. In an intra-campus move, the logical IP address of the host remains unchanged and only the binding changes whereas in the inter-campus move the logical IP address of the mobile host changes and it needs to acquire a temporary address. In the intra-campus environment, IP datagrams from a non-mobile host are sent to the nearest MSR; if the target MH is served by that MSR, the datagram is simply forwarded to the MH. Otherwise, if the MSR knows which MSR is currently serving the concerned MH, it performs another IP encapsulation of the IP datagram and forwards it to the other MSR. The receiving MSR removes the outer encapsulation and delivers the inner original datagram to the MH. If an MSR receives a message for a target MH whose current location that it is unaware of, then it sends a query to all other MSRs within the campus. The MSR with whom this mobile host is currently registered with responds to this query. The sending MSR caches this information and uses it in the future datagrams (without the need for a campus-wide query). Therefore, routing a datagram to an MH requires first routing the datagram to the nearest MSR, which then tunnels the datagram to the MH's current MSR, which subsequently delivers the datagram to the target MH.

When a mobile host moves to a foreign campus, the MH needs to acquire a temporary address known as the nonce address from the network it is visiting. The equivalent terminology for nonce address in IETF mobile IP architecture is care of address. Such a mobile host is referred to as the 'popup'. It maintains two addresses, namely, its home and the nonce addresses. The MH then informs the home MSR of its foreign address. The home MSR subsequently tunnels all the packets meant for this mobile host to its current nonce address. Any packets that the mobile host transmits are tunneled back to the home MSR for forwarding. The MH has the option of using either of these addresses in the source field of the datagram that it sends while located in the foreign campus.

Within a campus, each MSR advertises its presence using beacon packets that the MSR transmits at regular intervals. Each non-MSR router within a campus points to some MSR as the shortest path to the mobile host. However, the shortest path from two non-MSR routers to the mobile subnet will not necessarily point to the same MSR as an MH is constantly moving between cells. This implies that the forward and the reverse paths are asymmetric. Additionally, since the wireless (mobile) interface of each MSR is directly attached to the mobile subnet, the route entry for the mobile subnet at an MSR, as seen by a protocol external to mobile-IP, points to the wireless interface.

2.2. *Multicast extensions*

The multicast extension proposed in Acharya *et al.*[6] for the mobile IP-based network described above uses the DVMRP multicast routing protocol[7], which forms the basis of MBONE.[8] MBONE is an example of a multicast network overlaid on top of the traditional unicast Internet. DVMRP constructs a source-rooted multicast delivery tree using variants of reverse-path broadcasting algorithm (RPB). The major difference between RIP and DVMRP is that RIP is concerned with calculating next hop to a destination, while DVMRP is concerned with computing the previous hop back to the source. Since mrouted 3.0, DVMRP has employed the RPM (reverse-path multicasting) algorithm. DVMRP forwards the packets away from a multicast source along a group's RPM tree. The general name for this technique is reverse-path forwarding.

Multicasting in a mobile network involves the following issues :

- If the source of a multicast datagram is a mobile host, then a copy of the datagram may not reach all hosts (static or mobile) that are members of the multicast group. Typically, in distance vector-routing protocols such as DVMRP, the source address of a multicast datagram plays a crucial role at an intermediate router. If the datagram did not arrive on the shortest reverse path (from the router) to the source, the datagram is not routed further and is silently discarded.
- A mobile host may experience a delay in receiving multicast datagrams when it enters a cell that has no other group member located in the same cell.
- Datagrams multicast from a static source may not reach some cells depending on the time-to-live (TTL) value used in the multicast datagrams; hence the same mobile host may receive datagrams from that source in some cells but not in others.

The crux of the problem in multicasting with mobile hosts is that even though all MHs and wireless interfaces of MSRs within a campus share a common subnet address, link-layer connectivity amongst MHs and an MSR is present only within a single cell. IP multicast, on the other hand, implicitly assumes that if there are multiple routers connected to a subnet, then a link-layer transmission from any host on the subnet reaches all routers and hosts on that subnet. This implicit assumption is not valid in the presence of a logical (mobile) subnet physically partitioned amongst multiple MSRs. Acharya *et al.*[6] consider a solution to this problem that consists of 'healing' the partition amongst the MSRs using a predefined multicast tunnel. In other words, the concept of unicast tunnels used in mobile IP and IP multicast is extended to form a multicast tunnel or MTUNNEL that links all MSRs within a campus. The MTUNNEL

provides an abstraction of link-layer connectivity among the MSRs. This abstraction along with appropriate modifications to IGMP guarantees reliable routing of datagrams from mobile hosts to all group members; it also ensures that an MH experiences no delay in receiving datagrams regardless of the mobility within the campus. This MTUNNEL uses a reserved multicast address for all MSR group and is used to forward multicast datagrams and IGMP messages from MSR to all other MSRs using IP within IP encapsulations. The encapsulating IP header for a packet sent on the MTUNNEL contains the all-MSR address in its destination field and the forwarding MSR address in its source field.

The multicasting approach presented in Acharya *et al.*[6] is scalable and addresses issues such as routing of datagrams, correct delivery of datagram to intended recipients and delay factor when receiving datagram in a different cell. However it does not address security and privacy issues. We now extend this model and consider the provision of secure multicasting service.

3. Secure multicast extensions

The following principles and assumptions are used in the formulation of our security model:

- We assume that a public key infrastructure is in place in the form of a certification authority (CA) or a hierarchy of certification authorities for the purpose of authentication and public key distribution. A certification authority is a trusted entity that verifies the identity of a participating entity, allocates a distinguished name to it and vouches for the identity by signing a public key certificate for that entity using its private key.
- Every host is initially registered in the campus. It receives a certificate that is signed by some CA that is local to the host.
- Each MSR has a public key and maintains a cache of the public keys of other MSRs in the campus.
- Each router in the wired network has a public key and maintains in its cache the public key of its neighbor.
- Each MSR and its next hop router on the wired network have each other's public keys cached within them.
- An MSR is essentially a router capable of handling mobility. It may use routing protocols such as RIP to exchange routing information with other MSRs periodically. If MSR does not exchange such information periodically and does not answer to query requests from other MSRs, then it is considered to be non-operational.
- All MSRs in the campus share a group key that is used to encrypt reserved multicast address (for the MSR group) and all multicast datagrams and IGMP messages from one MSR to another.
- If an MSR is found to be non-operational, other MSRs need to recompute the group key. This involves eliminating the non-operational MSR.
- Each multicast group has a designated member whose responsibility is to determine who has access to belong to this group. Normally, this entity is the initiator of the

group. If the initiator leaves the group, a new designated member can be chosen via an election algorithm or some other equivalent manner.

- Each MSR maintains a binding between the group identity and the identity of the designated member.
- Each MSR is trusted for authenticating a mobile user based on public key information and for maintaining the relevant multicast group information.
- The beacon message of an MSR includes its public key certificate.
- Our security model is based on IGMP v2.[9]

We describe the security extensions by considering the following stages:

- Stage 1 considers a mobile host joining an existing multicast group or initiating the creation of a multicast group or leaving a multicast group.
- Stage 2 considers the movement of the mobile host to a foreign campus.

3.2. *Stage 1: MH joining an existing multicast group*

Consider the situation where a mobile host MH wishes to join an existing multicast group G. The process of registering with a multicast group is as follows. MH first obtains the public key of the receiving MSR (MSRx) from the certificate in the beacon message that the MSR sends. MH then sends a join message. This join message is typically an IGMP v2 protocol message that is extended to include security-related information as follows:

MH → MSRx: CERT-MH, MH, T, N, MSRx, G, {T, N, MH, MSRx, G}SK-MH.[6]

where

- MH is the identity of the mobile host that wishes to join the multicast group.
- CERT-MH is the public key certificate of the mobile host.
- MSRx is the identity of the receiving MSR.
- T, N: Timestamp and Nonce generated by MH.
- {...}SK-MH: This notation implies that the contents within {..} are hashed and signed using the private key of MH, SK-MH. In this case, it includes the nonce, timestamp, identity of the MSR, and the address of the multicast group (G) that MH wishes to join. The signed element is referred to as the Token of MH.

The message contains a signed timestamp T and nonce N to prove its freshness and to protect against replay attacks. The signed element also includes the identity of MH and MSRx along with G that the mobile host wishes to join.

Upon receiving this message, the receiving MSRx verifies the certificate and uses the public key of the MH recovered from the certificate to verify the signed element. It checks the integrity of the message and whether MSRx itself is the intended recipient. It also checks the timestamp and nonce to establish whether the message is fresh. Then MSR*x* refers to its multicast table entries to identify the appropriate designated member for this group G. It then con-

structs the following message that is dispatched via the MTUNNEL and the wired interface of MSRx:

MSRx → DMH: CERT-MSRx, MSRx, Dest-Addr, T1, N1, N, MH, CERT-MH, {Dest-Addr, T1, N1, MH, MSRx}SK-MSRx, {T, N, MH, MSRx, G}SK-MH[10]

where

- CERT-MSRx is the certificate of the sending MSR.
- Dest-Addr is the destination address of the format: <RouterId, AllMSRGroup, (G, DMH)> where
  - RouterId is the address of the router at the appropriate interface.
  - AllMSRGroup is a multicast address for all MSRs.
  - G is the target group address that MH wishes to join.
  - DMH is the designated member of the group for the above-mentioned groupid.
- T1,N1: Timestamp and Nonce generated by MSRx.
- CERT-MH: Certificate of mobile host.

The message has two subsections. The first contains a timestamp T1 and a nonce N1 to prove its freshness and to protect against replay attacks. It also contains the identity of the sending MSR and the initiating mobile host and their certificates as well as the destination address (Dest-Addr). The Dest-Addr itself it divided into two sections: The outer section contains the broadcast address and the multicast address of the MSR group of the campus. The inner address indicates to whom this message is meant for, i.e. the designated member of the group G. The signed element includes the nonces and timestamp and the identities of the sender MSRx and the receiver Dest-Addr that contains G. This element is hashed and signed using the private key of MSRx. The second section of the message contains the MH token that was sent as part of message.[6] This indicates to the receiving designated host (DMH) that it was the MH which actually wished to be part of the multicast group G. The two signed components are linked via the identity of the MH, the group G, and the nonce N.

As mentioned before, by using a combination of MTUNNEL and wired interface routes, the message gets delivered to the intended designated host. The intended host responds back with an access accept or access-deny message which is delivered back to the sending mobile host.

The format of the access accept message is as follows:[11]

DMH→MSRx: CERT-DMH,DMH,<Id,Cert>,
Dest-Addr,N1,N,T2,N2,Graft(MH,G),[Ks]PK-MH,
{DestAddr,DMH,N1,N,<Id,Cert>,Ks,T2,N2, Graft(MH,G}SK-DMH

where

- CERT-DMH is the certificate of designated member host of the group.
- DMH is the identity of the sender who is the designated member or the initiator of this multicast group.

- <Id,Cert> is a list of identities and the corresponding certificates of other members of the group.
- Dest-Addr stands for destination address with the format: <RouterId, AllMSRgroup,(G, MH)>
- T2 and N2 are Timestamp and Nonce generated by DMH.
- Graft(MH,G). This message indicates that MH has been accepted into the multicast group. It is a message for MSRs and other routers to update their multicast tables.
- Ks : A fresh group session key computed using a secure lock technique described below and it is encrypted using the public key of MH.
- The signed section includes the identity of the designated host DMH, destination address, associated timestamp and nonces along with the Graft message. All these parameters are hashed and signed using the private key of DMH.

This message flow contains a timestamp and nonces of messages (6) and (10) to indicate its freshness and to protect from replay attacks (as well as to indicate that it is a response of message (10)). It also includes the certificate of the sender and its identity (DMH and CERT-DMH) along with <Id, Cert> that lists the identities and certificates of other members of the group. In practice, this list can be made to be optional for general services such as news and weather. However, for specific services such as a corporate meeting in an organization, one requires the information pertaining to other members of this group. In such situations, it is advisable to make this field mandatory. The destination address in this case includes the multicast address for all MSRs, the target group address that MH wishes to join and the identity of the mobile host. Instead of sending this message directly to the recipient of the message, it is sent to the destination address because the MSRs must be informed of the status of this request for them to cache this information. Also the recipient DMH must authenticate itself to the MSR and other members of this group G must be aware of MH's presence. The signed element includes the identity of the DMH, Dest-Addr and the associated timestamp and nonces for integrity checks. It also includes the Graft message that explicitly states that this mobile host is now part of group G. The encrypted section contains the session key required by the MH to communicate securely with the group.

Access-deny message is sent when an MH is not accepted into a particular group. This is determined by the local group policy that governs its access control list. Access-deny message has the following format:

DMH → MSRx: CERT-DMH, DMH, Dest-Addr, N, N1, T2, N2, MSG, {Dest-Addr, N, N1, T2, N2, MSG}SK-DMH[1]

Note that this message contains almost the same parameters as in the accept message except that it does not contain the Graft message and group member identities. This indicates that the mobile host MH has not been accepted into this group. Also it contains a message MSG that indicates the reason for not granting the MH the permission to be a part of this group.

The final step involves MSRx conveying DMH's message (Access-Accept message) to the concerned mobile host MH.

MSRx → MH: CERT-MSRx, MSRx, CERT-DMH, DMH, <Id,Cert>, T3, N, {Graft (MH, G)}SK-DMH, [Ks]PK-MH, {N,T3, MSRx, G, MH, <Id,Cert>}SK-MSRx[7]

This message includes the original timestamp and nonce used by mobile host in message[6] to bind this response to the request. The message format to convey Access–Deny message will have a similar syntax except that it will not contain the fields <Id, Cert> and the Graft message. Instead it will include a message MSG that might indicate the reason for not granting the MH the permission to be a part of this group G.

3.3. *Stage 1: MH initiating a multicast group*

A mobile host initiating a multicast group creates an access control list (ACL) and a security association (SA) for the session. It then announces this group by sending an advertisement message across the internetwork. The announcement may be advertised to potential members by directing it to particular multicast address reserved for receiving session announcements (SAP) or alternatively invitation protocols such as SIP (session initiation protocol) may be used to convey the announcement to a specific group.[12] Each valid recipient performs an authentication process (involving itself and its current MSR) using a process similar to that described in Section 3.2. The request is passed on to the initiating host, which then computes the group key. This group key is computed from the public keys of the participants using the scheme described below.

Several key management schemes and protocols[10] exist for securely distributing keys in a network environment. In this paper, we use the secure lock technique suggested by Chiou and Chen.[11] It uses the Chinese remainder theorem to generate a 'secure lock' to lock the deciphering group session key. The secure lock is transmitted with each encryption message. Only users in the secure group can unlock the session key. This scheme is only efficient for small groups; in a campus-type environment where a multicast group may not have a large number of participants, this can be sufficient. Initiator must store the public keys of each of the participants. Considering the fact that the computational and storage capacity of modern-day mobile systems is rapidly increasing, this does not seem to be an issue.

From the Chinese Reminder Theorem, for N1.... Nn positive, relatively large prime integers and R1.... Rn, positive integers, a set of congruous equations

X=R1 mod N1, ..., X=Rn mod Nn

has a common solution X in the range of [1,L-1} where L=N1*N2*N3*....Nn, where n is the number of participants in the group.

The Chinese Remainder Theorem is used to generate X where Ri = [Ks]PKi where the session key Ks is encrypted using the public key PKi. The common lock X is a function of each of the participants' public key. Therefore, only those participants whose public keys are included in the calculation of X can unlock d.

Dynamic addition and deletion of group can be carried out as follows. Every time there is a change in the group membership, the initiator can recreate the common X and modify the group to include or exclude certain participants from future communications. As far as the

storage requirements are concerned, the initiator who is the creator of the lock must store the public keys of each of the participants. Decipherment of the session key Ks for each participant is fairly efficient. The scheme is a centralized one, as the computation of X is restricted to a single entity who is the initiator, thereby offering better control; however it does not scale well to large groups.

3.4. *Stage 1: MH leaving a group*

When a mobile host leaves a particular multicast group it needs to send an explicit IGMP v2 exit message to its MSR. This message includes the identity of the mobile host along with its group membership details such as group address and the designated host address. Upon authenticating this request, this message is routed to the designated host. The designated host then recomputes X, which now does not include the public key of the host that has decided to leave.

3.5. *Stage 2: MH moving to a foreign campus*

The following occurs in an inter-campus movement of a mobile host.

1. Having identified that it is in a foreign domain, MH undergoes an authentication process with the local MSR before registering itself in the foreign campus.
2. The registration process involves the foreign MSR informing the MSR of the home campus of the current location of MH.
3. The MH also informs the foreign MSR of any multicast groups it currently belongs to. Alternatively, as soon as it detects a mobile host entering its cell, the MSR sends a membership list that consists of groups that have members local to the cell. If the current foreign campus is already registered with this multicast group, then all the MSRs within this foreign campus get the multicast datagrams and hence they get delivered to the MH straightaway. If not, there is an initial delay of MH getting registered in this campus. All MSRs update their multicast tables with this information and thereafter the MH gets datagrams delivered to it. All multicast datagrams are then tunneled to the current location of the MH from its home campus.

There are several issues that arise when a mobile host moves to a foreign campus and wishes to use certain services. In such a scenario, some trusted authority in the foreign campus (in this case, it could be an MSR) needs to authenticate the mobile host to verify its credentials. Having done this, this trusted authority needs to inform the mobile host's home campus of its current location in a secure way. This is necessary, as the home campus needs to forward the mobile host's incoming data to the foreign domain. An additional issue is that of anonymity, which is absent in the intra-campus moves. The anonymity problem addresses the issue of disclosing the identity and movements of a mobile host to the relevant entities. In an ideal situation, only the home campus is aware of the true identity, and the whereabouts and movements of a mobile host. In such a case, the true identity of the mobile host is anonymous with respect to the foreign campus where it is currently located. However, at the same time, the foreign campus needs to know some information of the mobile host to verify its credentials and perhaps for billing purposes. If the true identity of the mobile host is revealed to the foreign cam-

pus, then it becomes relatively easy for the foreign authorities to keep track of mobile host's movements. Clearly, in some cases, this is not desirable.

The intuitive first step solution to this problem is to assign a traveling alias to every mobile host when it is away from home campus. Then the key question is: should the alias be fixed or should it be continually changed? If the alias is fixed, then an attacker, who is closely monitoring the host's movements, may still be able to associate the alias with the true identity of the host. If the alias is constantly changing, then it becomes difficult for the attacker to associate these different aliases to its true identity. This also makes it almost impossible for a set of foreign campuses to link the entire set of movements of the mobile host. However, in such situations how does a home-MSR associate each of these aliases of the mobile host to its true identity? Also, the use of aliases prohibits the mobile host from using its certificate in the authentication process as certificates normally vouch for the true identity of the user. The crux of this discussion leads to the following requirement and design principle: there should be a set of procedures or mechanisms in place by means of which a mobile host can pursue its nomadic movements by using different aliases and yet be able to authenticate itself to the foreign campuses by remaining under the jurisdiction of its home campus. In this section, we present a scheme for facilitating anonymity, which aims to fulfill this requirement.

The MSR in a home campus generates a random number R. R is a constant. Using R and the identity of the mobile host (MH), MSR computes the first alias (A1) in the following way:

$$A1 = h(MH \oplus R) \tag{6}$$

h is assumed to be a strong one-way hash function such as the Secure Hash Standard (SHA).

Both the MSR and the mobile host know R. This alias is generated by the MSR soon after the mobile host registers in its home campus. The MSR maintains an entry for this mobile host and its alias in its directory. It also generates a token for this mobile host as defined below. This token enables the mobile host to authenticate itself to a foreign campus. The token format is as follows:

$$\text{Token} = \langle A1,\ PK\text{–}A1,\ Home\text{–}MSR,\ CERT\text{–}Home\text{–}MSR,\ Time,\ Validity \rangle,\ \{< >\}SK\text{–}Home\text{–}MSR \tag{7}$$

where:

- {< >}SK–Home–MSR denotes that the contents of token are signed using the private key of Home–MSR.
- A1is the alias that corresponds to a mobile host identity.
- PK–A1 is the public key of A1 (public key of mobile host MH).
- Home–MSR: is the identity of the home MSR that generated the token.
- CERT–Home–MSR is the certificate of the home MSR.
- Time, Validity: Validity is the time period for which the token is valid. Time denotes the time the token has been generated

Home–MSR → MH: Home–MSR, MH, time, Token, {MH, time, token}SK–Home–MSR

The token sent (from definition 7) serves as a pseudo certificate for a mobile host. It can present this token in its first visit to a foreign campus. This token proves that a mobile host A1 is under the jurisdiction of an MSR representing a campus. Some certification authority vouches the credentials of this Home–MSR. This is indicated by the certificate of Home–MSR. Validity period allocates a lifetime for this token. It serves the same purpose as a certificate lifetime; however, the time span for this period is comparatively short. Since the token is signed using the private key of the MSR, it binds this token to its originator (Home–MSR).

Let us now consider the authentication process in a foreign domain.

1. The mobile host (MH) recognises that it is in a foreign domain by means of the beacon message sent by some foreign MSR known as F–MSR. Having identified that it is in a foreign campus, MH undergoes an authentication process with this F–MSR before registering itself in the foreign campus. This authentication process is as follows:

   A1 (MH) → F–MSR: A1, T, N, F–MSR, Token, Msg, {T, N, F–MSR, Token, Msg, A1}SK–A1 (8)

   The use of message field Msg is host specific. In this case, the host may use this field to convey to MSR the multicast group(s) that it belongs to.

2. F–MSR can verify the signature using MH's public key obtained from the token. This is necessary to prove that message actually originated from A1 and not from anyone masquerading as A1. F–MSR verifies the credentials of A1 by verifying the signature of its home MSR using the home MSR's public key procured through the certificate embedded in the token.

3. The registration process involves the foreign MSR informing the home MSR of the current location of MH. The foreign MSR constructs the following message for home MSR of the mobile host.

   F–MSR → Home–MSR: CERT–F–MSR, F–MSR, Home–MSR, A1, T1, N1, Token, {Home–MSR, A1, T1, N1, Token}SK–F–MSR (9)

   Authentication token of A1 is signed by the foreign MSR. This is necessary for foreign MSR to prove to Home MSR that A1 is actually in its campus. This is because alias A1 is being used for the first time by the mobile host MH. Only MH could have supplied this credential as itself and Home–MSR are the only entities who are aware of this identity.

4. Upon receiving this message, the home MSR verifies the credentials of foreign MSR and the presence of MH as A1 in its domain. It then creates a new alias A2, which will be used by the mobile host when it moves to the next foreign campus. It completes the registration process by sending this new alias as part of a new token (Token') to the F–MSR.

   Home–MSR → F–MSR: Home–MSR, N1, T2, N2, A1, [Token']PK–MH {F–MSR, N1, T2, N2, A1}SK–Home-MSR [10]

   Upon receiving this message, the foreign MSR verifies the signature of home MSR using the public key of home MSR obtained from its certificate. Note that the contents of the new

token Token' (including the new alias A2) is not visible to the current F–MSR as it is encrypted with the public key of the mobile host MH.

5. Finally, the registration process is completed by F–MSR forwarding the new token to the mobile host.

   F–MSR → A1 (MH): F–MSR, A1, T3, N, [Token']PK–MH, {A1, T3, N, [Token']PK–MH}SK–F–MSR (11)

6. The new alias A2 in the token Token', for use in the next signed foreign campus, is generated as follows:

   $A2 = h(A1 \oplus MH \oplus R)$.

   The alias A2 is a function of alias A1 and the identity of mobile host. The forthcoming aliases would be computed as follows:

   $A3 = h(A2 \oplus MH \oplus R)$

   $A4 = h(A3 \oplus MH \oplus R)$ and so on.

   Thus the mobile host acquires a new alias An to be used in the next campus by using its current alias An-1. This is done in the registration phase to maintain a strong binding between the mobile host and its home MSR. The same procedure is repeated when moving to the next foreign campus. This approach makes the aliases independent of the foreign domain and maintains a tight synchronization between the mobile host and its home authority.

This scheme conceals the identity of the mobile host from the outside attackers and foreign domains. But it is not intended to conceal the identity from the home agent of the mobile host and from the members of the multicast groups to which the host belongs. Such an approach is realistic and satisfies the general spirit of mobile IP architectures.

Consider the following two cases.

- In the first case, the mobile host registers in the multicast group G when it is in its home domain. The identity of the mobile host is disclosed at the time of registration. When the host moves to a foreign domain, say, it acquires an alias A1. It uses this alias to access the services offered by G. Even though it has acquired this new alias, it can still avail the services of G as the multicast group key remains the same as the host's keys remain the same. Hence there is a possibility that the DMH and other members of the group may be able to identify the mobile host by mapping the aliases of the mobile host to its public key. However we envisage an environment where the public key to user name mapping is tightly controlled. In other words, if an entity requests the user name or the certificate of a user to the concerned authority, his request is declined straightway.
- In the second case, the mobile host joins a multicast group when it is in a foreign domain. In this case, the true identity of the host is not disclosed as it is under an alias say A1. Here the members of the group may be able to map the various aliases to A1 but cannot map them to the true identity of the mobile host.

4. Conclusions

A variety of services with different levels of quality will use multicasting both in a wired and wireless mobile environment. The area of mobile IP multicasting is relatively new and there is not a widely accepted method for multicasting in such environments. The framework proposed in Acharya *et al.*[6] seems to provide a good basis for considering multicasting in mobile IP networks. In this paper, we have extended this framework by developing a security model that can be used to provide a secure multicasting service. In particular, we have considered the various phases of a mobile host joining, initiating and leaving a multicast group and have proposed appropriate security protocols. The paper also considered secure group key generation and distribution. Finally, the paper discusses the movement of mobile hosts between campuses and describes an alias-based authentication scheme in such an inter-domain environment.

References

1. DEERING, S. — *Host extensions for IP multicasting*. RFC 1112, August 1989.
2. SCHULZRINNE, H., CASNER, S., FREDERICK, R AND JACOBSON, V — RTP A transport protocol for real time applications, RFC 1889, Jan. 1996.
3. IOANNIDIS, J. AND MAGUIRE, G. — The design and implementation of a mobile internetworking architecture, USENIX, San Diego, CA, 1993.
4. TERAOKA, F. — Sony VIP: IP extensions for host migration transparency, Draft RFC, 1992
5. PERKINS, C — *Mobile IP: Principles and practices*, Prentice-Hall, 1998.
6. ACHARYA, A., BAKRE, A. AND BADRINATH, B R. — IP multicast extensions for mobile internetworking, Rutgers DCS TechReport LCSR-TR_243, 1995.
7. DEERING, S , PATRIDGE, C. AND WALTZMAN, D — *Distance vector multicast routing protocol.* RFC 1075, November 1988.
8. ERIKSSON, H. — MBone. The multicast backbone, *Commun. ACM*, 1994, **37**, 54–60.
9. FENNER, W — Internet Group Management Protocol, Version 2 (IGMP v2), RFC 2236, Nov. 1997.
10. BALLARDIE, A. — Scalable multicast key distribution, RFC 1949, May 1996.
11. CHIOU, G. H. AND CHEN, W. T. — Secure broadcasting using secure lock, *IEEE Trans.*, 1989, **SE-15**, 929–934.
12. KRUUS, P. S. — A survey of multicast security issues and architectures, 21*st Natn. Information Systems Security Conf.* (NISSC), Arlington, VA, 1998.