Fault Tolerant Authentication in Mobile Computing *

Bharat Bhargava Sarat Babu Kamisetty Sanjay Kumar Madria

Center For Education and Research in Information Assurance and Security
and
Department of Computer Sciences
Purdue University
West Lafayette, IN, U.S.A.
{bb, kamisesb, skm}@cs.purdue.edu

Abstract *Survivability and secure communications are essential in a mobile computing environment. In a secured network, all the hosts in the local network must be authenticated before they communicate with the hosts outside the network. The failure of the nodes/agents that authenticate the hosts may completely detach the hosts from the rest of the network. In this paper, we describe two techniques to eliminate such a single point of failure. Both of these techniques make use of backup servers, but they are architecturally different. We address the scalability issue of the techniques proposed by a cluster-based scheme where the front-end directs the requests to a group of back-end machines.*

Keywords: authentication, fault-tolerance, mobile

1 Introduction

Providing security services in the mobile computing environment is challenging because it is more vulnerable for intrusion and eavesdropping. Most of the existing wireless network models assume the presence of stationary base stations which is not quite true in all scenarios. For example, in the tactical mobile networks, base stations also move from one network to another network. Typically, a Base Station(BS) serves all the hosts in a Local Area Network. To get the service, the hosts have to authenticate themselves with BS. Therefore, each packet contains authentication information apart from the actual data. Once the authentication is successful, the packet-forwarding is done. All the hosts in network have a default route to BS in their routing tables i.e. all the packets originating from the hosts in the LAN will go to BS no matter where they are destined. Assume that the base station BS moves to a foreign network. As the hosts in LAN are not aware of BS's movement, they still keep sending their packets to BS. Since BS is not in the home network currently and there is no other base station that could forward packets destined to BS to the foreign network where BS is present currently, all the packets that originate from any host in the LAN are dropped. Essentially, now all the hosts in the LAN are isolated from the Internet. This disruption of service is caused by the movement of base station which the traditional networking protocols cannot handle. Two simple approaches to handle the above problem are as follows. First approach is to set up proxy base station in the network and change the default route in all the hosts to point to this proxy base station. This approach is practically unacceptable because routing tables of all the hosts should be updated manually. This becomes tedious if the number

*Portions of this work were supported by sponsors of the Center For Education and Research in Information Assurance and Security, NSF under CCR-9901712 grant and IBM grant

of hosts in the network are large. Moreover, manual configuration is error prone. Secondly, the currently running applications need to be restarted. Since the tables are updated manually, it is a time consuming process and hence provision of service is disrupted. Ideally, applications should be unaware of the base station's movement.

Another approach is to have another base station that forwards all the packets that originated within the LAN to BS when BS is visiting a foreign network. However, the communication delays that are introduced by this solution are totally inacceptable. Consider a packet originated from a host in N1 that is destined to a host in network N2. As shown in figure 1, when BS is in the home network, the route R1 is taken to reach the destination. If BS is visiting a foreign network then first the packet is forwarded by the new base station BS1 to BS via route R1 and from there, the packet is sent to the destination via route R2. In most of the cases, the path taken in the second scenario is longer than the one in the first case. The problem here is that there is a single point of failure (in our example, it is BS).

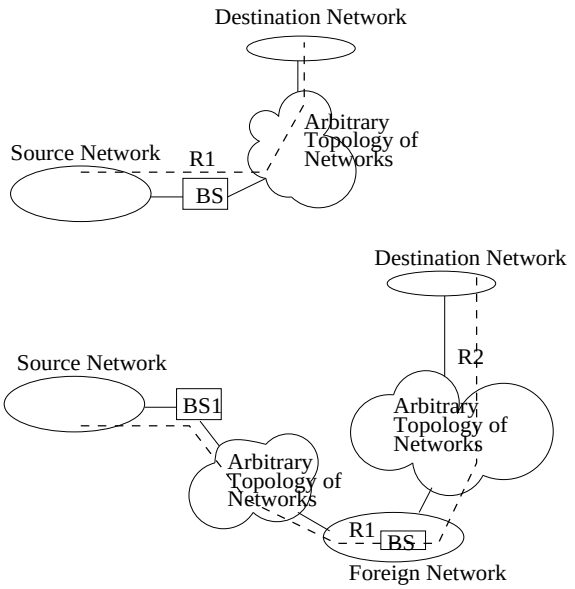

Figure 1: Solution using another base station

We need an architecture that eliminates this point of failure and provides smooth (i.e. without any disruptions) service to the hosts while still allowing the mobile hosts to move in and out of the home network. In the above example, we discussed about only one kind of service namely authentication. In general, it could be any form of service like secure database access.

In this paper, we propose techniques for providing uninterrupted service to the mobile hosts while still allowing the service providing agents to move or fail. Our scheme eliminates the manual configuration, does not add any communication delays and does not impose any security threats as the packets are never allowed to leave the local network. This scheme is "smooth" since the applications can be totally ignorant of failures.

2 Mobile IP Security

In this section, we briefly describe the services that Mobile IP protocol provides and identify some drawbacks of Mobile IP from security point of view. Mobile IP[6][9] enables hosts to move from one IP network to another. It is suitable for mobility across homogeneous media (for example, ethernet to ethernet) as well as mobility across heterogeneous media (for example, ethernet to wireless LAN). A **Mobile Host(MH)** changes its point of attachment from one network to another. A **Home Agent(HA)** is a router on the MH's home network which tunnels datagrams for delivery to the MH when it is away from home, and maintains current location information for the MH. A **Foreign Agent(FA)** is a router on a MH's visited network which provides routing services to the MH while registered.

While Mobile IP promises un-interrupted IP connectivity as MHs move in the Internet, it also increases the risk of causing remote redirection of traffic[2] by simply introducing bogus registration and binding update messages. Moreover, the presence of MHs in the foreign networks may cause security problems to both the home and foreign networks.

The two goals of Mobile IP security protection are to allow a MH enjoy similar internet connectivity and safety when it visits a foreign network as it is in its home network and to protect both the home and the foreign networks from passive and active attacks In order to frustrate the remote traffic redirection attack mentioned above, registration messages include 64-bit identification tag for detecting replay attacks and one or more authentication extensions [4][5] to provide message integrity and strong authentication using a Message Authentication Code(MAC). Although the use of MAC and an anti-replay tag addresses the security services cited above, the current Mobile IP lacks a scalable key management scheme for dispatching cryptographic keys needed to support these services. To protect registration messages, keys must be shared atleast among the MHs and their HA.

3 Fault-tolerance

Authentication is the mechanism by which the receiver of a message can ascertain its origin [8]; an intruder should not be able to masquerade as someone else. Most of the authentication protocols proposed till now require a trusted third party which generates the secrets keys for the communicating parties. There are some drawbacks with this approach. For example, if the number of communicating parties are more, then the third party becomes a bottleneck. It also becomes an attractive spot for attackers. If a malicious guy breaks into the trusted party's secret database, all the keys are compromised.

In mobile networks, when a MH wants to securely communicate with other hosts, it has to be first authenticated by the HA. When there is a single HA, this becomes a single point of failure i.e, when the HA fails, all the MHs in the home network cannot communicate with the outside world. A simple but powerful solution to this problem is to have back-up HA(s), which assumes the responsibility of a Master when the current Master fails. Failure of a HA can be detected by listening to the agent advertisements. If the Master is not responding or advertising since a fixed amount of time, then it can be declared to be dead and one of the backups take up the responsibility of the Master Home Agent. This requires that the secret key database is fully replicated on all the backups too and introduces a potential security threat as there are several sites that could be attacked now In section 4.1, we propose a refined technique using this idea that achieves controlled and smooth transitions between the Master and Backup. This is an extension to the idea specified in[7]. In section 4.2, we propose a scheme to address issue using an entirely different approach. The idea is to logically arrange these back up servers in a hierarchy that represents the communication flow. In case of a Home Agent failure, a node at higher level in the hierarchy can authenticate the MHs of that network and provide network services uninterruptedly.

4 Proposed Schemes

In this section, we propose two different schemes for achieving fault-tolerant authentication. The first approach uses an abstract entity called Virtual Home Agent and the second approach requires that different Mobile Agents be arranged logically in a tree structure. Fault-Tolerant Authentication is essential in tactical mobile military networks where the base stations are subject to failure.

4.1 Virtual Home Agent Scheme

We define the following entities that we use in the scheme. A **Virtual Home Agent(VHA)** is an abstract or virtual agent that is identified by a network address (eg: IP address). All the hosts send their authenticating requests to the VHA's network address. VHA acts as a default HA for the MHs in the LAN. VHA's responsibilities include authenticating the MHs by using a **Shared Secrets Database**. A **Master Home Agent(MHA)** is a HA that is currently assuming the responsibilities of a VHA.

For a VHA, at any given point of time, there will be only one MHA assuming that VHA's responsibilities. A MHA intercepts and processes all the packets destined to VHA's network address. A **Backup Home Agent(BHA)** is a HA that backsup a given VHA. There could be more than one BHA for a given VHA, in which case, each BHA will be assigned a priority. In the case of failure of the MHA, BHA having the highest priority becomes the MHA. Each of the MHs share a secret key with the VHA. Typically, a secret could be a password or a symmetric key like DES key. All these secret keys are stored in a separate database called **Shared Secrets Database**. A **Shared Secrets Database Server** is a server that protects and processes the queries and updates to the Shared Secrets Database is called Shared Secrets Database Server. The VHA will send requests to this server while authenticating a MH or when a new shared secret is issued to a MH. To frustrate the impersonating attacks by malicious hosts, the VHA has to authenticate itself with this server. Figure 2 illustrates the above discussed entities in a typical scenario. In the figure, the MHA is the Master Agent for

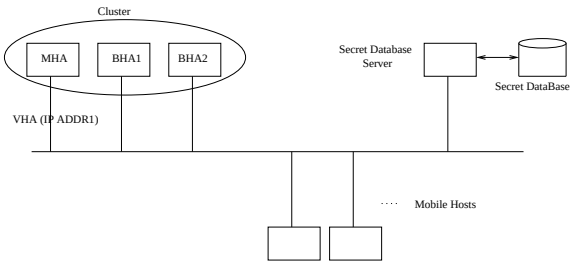

Figure 2: A Sample LAN environment

VHA identified by the IP address IP ADDR1. BHA1 and BHA2 are the Backup Agents for the VHA. Only the MHA contacts the Shared Secrets Database Server. The protocol functionality is described below.

Periodically MHA sends advertisements on the network to a pre-configured multicast address. All the BHAs and MHA join this multicast group. Each BHA is assigned a priority which indicates the administrator's preference for a BHA to become MHA if the current MHA fails. The MHA has the lowest value for the priority than all the BHAs. Each advertisement is a packet that contains the following items - VHAs IP Address, MHAs Priority and Authentication information. This advertisement is transmitted periodically every few seconds and this time period is called Advertisement Interval. All the BHAs listen to these advertisements. If the advertisements are not heard for some period of time, then the election of a new Master starts.

Typical election protocols require the backups exchange their priorities and then elect the new Master. But they introduce more traffic and do not provide smooth transition from Backup to the Master because of communication delays. Another drawback is that the priority values could be manipulated while exchanging by malicious nodes biasing the election result. The following scheme overcomes all these disadvantages. Each BHA sets the Down Interval Timer as described below. When the Down Interval Timer expires, the BHA transitions to the Master state.

```
Down Interval Time = 5 * Adv. Interval
+ (BHA Priority / MHA Priority)
```

Each of the BHAs reset the Down Interval Timer whenever an advertisement is received on the multicast channel. There are two things that need special attention regarding the Down Time Interval. First of all, it's value is atleast five times the advertisement interval, so the election process will not start until the MHA fails to send five consecutive advertisements. A BHA might not receive some advertisements even though the MHA is alive due to packet losses, but five or more consecutive losses of the same packet is very unlikely. Secondly, the Down Time Interval is a function of BHA's configured priority. The BHA having the lowest priority value will have the lowest value for the Down Interval Time and hence it fires earlier than others.

In this election scheme there is no communication between the BHAs once the MHA goes

down eliminating security threats and time delays. It does not use extra bandwidth and it is guaranteed that only the Down Interval Timer of the BHA having the lowest priority value fires earliest and hence there is no confusion and no additional computations are required. The downside of this algorithm is that there is a possibility of partitions. Consider the following scenario. Once a BHA's Down Time Interval fires, it sends an advertisement to the multicast address announcing its presence as the Master. But the packet reaches only a subset of BHAs which are now aware of the new Master. The other subset of the BHAs haven't received the advertisement and one of them declares itself to be a Master. Now there are two Masters leading to chaos. But this is not such a serious problem in a LAN as packet loss is very less and especially broadcast networks support multicast in a natural way - only a single copy of a packet will be transmitted in a LAN eventhough it is a multicast packet. So, if a packet is lost, none of the hosts receive it.

4.1.1 Enhancements

Eventhough the scheme proposed is sufficient for most of the common scenarios, optimizations are possible by utilizing the Backup Home Agents appropriately. In this section, we describe some modifications to the scheme described earlier.

Doing encryption and decryption on every packet is very expensive. In the scheme described in the previous section, if the network is busy MHA becomes a bottleneck and it starts dropping the packets worsening the congestion. Moreover, the BHAs are just listening to the MHA's advertisements and donot service any of the requests. Hence, their processing capacity is not utilized properly. Eventhough, the default BS has been replaced by a virtual entity to eliminate single point of failure, the central database, might still be an attractive target to the attackers. We can replicate the database fully on all the BHAs and the MHA, eliminating the central database server, but this requires additional storage capacity on each of the BHAs and the MHA. Also, any updates to the database have to be carried out on all the BHAs and MHA.

To overcome above mentioned problems, we extend the scheme by forming a cluster consisting of MHA and the BHAs. A cluster is a group of servers acting as a single server which gives the effect of a multiprocessor machine. A cluster is identified by a single IP address and it consists of a front-end machine and one or more backends. Only the IP address of the front end machine is well-known. When a client sends a request to the front-end, the front-end forwards the request to one of the back-ends which services the request. Front-end does not process any of the requests, but just routes the request to an appropriate back-end. So, the front-end will not be a bottleneck. Typically, there will be more than one back-end and hence the throughput of the system increases dramatically. In our modified scheme, MHA acts as front-end and BHAs become the back-ends. So, BHAs are now used to process the requests instead of just listening to the MHA's advertisements. This increases scalability and efficiency of the system, especially when the backends are dedicated systems. The front-end has to do load balancing to avoid overloading a particular backend, has to keep track of which back-ends are active at any instant. Note that the back-ends in this scheme remain anonymous like in the previous scheme. Request redirection can be used instead of request forwarding. In request redirection, when a client contacts a front-end, the front-end chooses a back-end and redirects the client to contact that back-end. The downside of this approach is that the client gets to know the back-end's identities, not desirable from security point of view and it also places additional burdens on clients, increases communication delays and message complexity. The request forwarding decision by the front-end depends on different factors like the request contents, current load on the back-ends or current cache contents of the back-ends. A detailed description and a prototype implementation of this scheme can be found in [3].

4.2 Hierarchical Authentication

In this section, we propose another technique for achieving fault-tolerant authentication by tree based organization of HAs in the LAN. For the purpose of discussion we, use figures 3 and 4. In figure 4, we show two LANs that use this tree-based approach. Figure 3 shows logical organization of the Agents in LAN1. The dotted lines show the communication link between the two LANs. This could be wired or wireless media. In LAN1, Mobile Hosts D, E, F and G are MHs that are not routers. A, B and C are the Agents that could provide service to the MHs. One could think of large number of services that Agents could provide, for example, access to weather reports is another popular service: the hosts have to authenticate themselves with the Agents to gain access to the weather reports. Subscribed clients will be given a key-list so that they can get authorized access to the services. In LAN1, the Agents

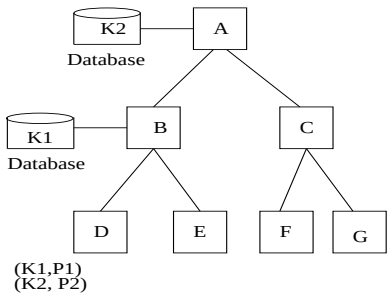

Figure 3: Tree-based organisation of Agents

are logically arranged in a hierarchy forming a tree like structure. The hosts (i.e. clients) are at leaf level. Intermediate levels are occupied by Agents. A leaf node shares a secret with each of the Agents that lie in the path from itself to the root of the tree. In the figure, host D shares a secret key with every host in the path from D to A i.e with B and A; with B it shares K1 and with A it shares K2. Each of the secret keys have a priority associated with them. The key having the highest priority will be used for authentication before the key with the next highest priority is used. Here, K1 has priority P1 and K2 has a priority P2. These priorities are assigned based on various factors like communication delays, processing speed, frequency of usage of the key, key's life time etc. A given key's priority should be a func-

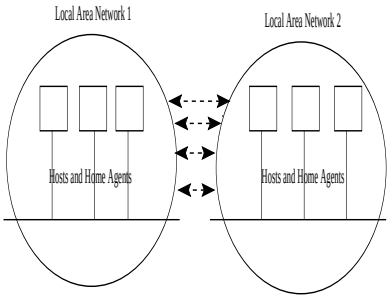

Figure 4: LANs using tree-based approach

tion of all these factors. Each factor could be assigned a weight factor W_{f_i}, where f_i represents the i^{th} factor. So, if P_{K_j} represents the priority of Key K_j, then P_{K_j} is computed as $P_{K_j} = \sum_{i=1}^{n} W_{f_i} * P_{f_i}$. In the formula, P_{f_i} is the priority of the key computed taking only factor f_i into consideration. For the sake of discusssion, we assign priorities according to communication delays assuming that the delay is proportional to the distance between the nodes in the tree. So, P1 is greater than P2 as B is nearer than A to D. Now, when D wants to communicate with any host in LAN2, it has to first authenticate itself with B or A. Since P1 is greater than P2, key K1 and Agent B is chosen. D sends a authentication request packet to B sending the secret K1 either directly or indirectly. For this any of the well known authentication schemes can be used. B sends back either a positive or negative acknowledgement based upon whether the authentication is successful or not. Once authentication with B is successful, D can communicate with any outside host as B provides the packet-forwarding service. For tighter security, every packet has to be authenticated. If B fails, then D will not get any acknowledgement from B. After transmitting the requests for fixed number of times, it now uses the secret key K2 to authenticate

itself to A. D can either discard K1 assuming that node B has gone down permanently (for example, in battle field) or reduce the priority of K1 to a value less than P2 and any other keys it has, assuming that B's failure is only temporary (for example in commercial networks or in mobile environment where the agent itself is mobile). Once D authenticates itself to A, it now will be able to communicate with the outside world unlike when there was a single point of failure. The problem becomes more challenging when the authenticating agent itself is mobile. In such a case, the current authenticating agent should handover its responsibilities to any existing backup authenticating agents.

5 Related Work

Eventhough numerous authentication protocols have been proposed and are widely in use on the Internet, the issue of fault tolerance and scalability has not received much attention. With the existing systems, if the server goes down, client will not be able to get services from the server. Our paper addresses the issue of fault-tolerance on LANs and scalability of the system. Our architecture is flexible to accomodate any authentication scheme. Virtual Router Redundancy Protocol [7] addresses the issue of eliminating single point of failure on LANs but the scheme is not scalable for very busy networks. Our contribution towards this issue is the proposal of the use of a cluster of nodes rather than a single node. Web servers based on clusters of workstations are being widely used in the coporate networks [1] to service the HTTP requests. We extend this scheme and provide fault-tolerance to provide un-interrupted services to clients.

6 Conclusions

In this paper, we discussed schemes for achieving fault-tolerant authentication in mobile environments. The two techniques presented for fault-tolerant authentication problem rely on same basic philosophy to handle failures – using backups, but architecturally, they are different; one is a flat model and other one is a tree-based model. To improve the system performance, a cluster based enchancement to the discussed model is proposed. Eventhough the proposed schemes solve the problem, some issues need further study. In tree-based model, key priorities need to be computed based on various factors like communication delays, processing speeds etc. Further experiments need to be conducted to discover the parameters that effect the performance of the system and study how the priorities depend on these factors. Another issue that needs to be addressed is, how should the secret key database be partitioned, so that system performance is optimal.

References

[1] S. D. G. A. Fox and et al. Cluster-based scalable network services. In *In Proceedings of the Sixteenth ACM Symposium on Operating System Principles*, October 1997.

[2] S. Bellovin. Security problems in tcp/ip protocol suite. *ACM Computer Communications Review*, Mar. 1989.

[3] B. Bhargava, S. Kamisetty, and S. K. Madria. Fault tolerant authentication and group key management in mobile computing. In *CERIAS technical report*, April 2000.

[4] S. Kent and R.Atkinson. Ip authentication header. *Internet Draft, draft-ietf-ipsec-auth-header-07.txt*, July 1988.

[5] S. Kent and R.Atkinson. Ip encapsulating security payload. *Internet Draft, draft-ietf-ipsec-esp-v2-06.txt*, July 1988.

[6] C. Perkins. IP mobility support. *RFC 2002*, Oct. 1996.

[7] D. W. S. Knight and et al. Virtual router redundancy protocol. *RFC 2388*, Apr. 1998.

[8] B. Schneier. *Applied Cryptography*. John Wiley and Sons, 1995.

[9] J. D. Solomon. *Mobile IP :The Internet Unplugged*. Prentice Hall Series in Computer Networking and Distributed Systems, 1997.