Distributed Authentication for Low-Cost Wireless Networks

Sridhar Machiraju
Sprint Applied Research
Machiraju@sprint.com

Hao Chen
University of California, Davis
hchen@cs.ucdavis.edu

Jean Bolot
Sprint Applied Research
Bolot@sprint.com

ABSTRACT

Cost is one of the key challenges facing the deployment of wireless networks. Though 802.11-based networks have shown that costly, licensed spectrum is not always necessary, the costs of other components especially backhaul and network equipment continue to impede the growth of mobile wireless networks. In this paper, we provide some insights into how such costs can be reduced by designing a novel, low-cost authentication infrastructure for wireless networks. Our authentication scheme relies on base stations to collectively store authentication information. Thus, it eliminates the need to maintain costly infrastructure required by the traditional centralized scheme. Moreover, our scheme is optimized for mobility-induced handover ”re-authentication” and, hence, reduces the authentication overhead.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

General Terms

Algorithms, Design, Security

Keywords

Distributed, Authentication, Token transfer

1. INTRODUCTION

In recent years, we have witnessed a rapid increase in the availability of low-cost wireless-capable devices such as 802.11-based cards and data-capable cellphones. However, the task of deploying and managing wireless networks still remains challenging due to high costs associated with backhaul, network equipment and spectrum. Various efforts involving 802.11-based networks have shown that it is possible to offset the spectrum costs by operating in free, unlicensed spectrum while providing “acceptable” performance. Yet, all managed wireless networks including 802.11-based networks continue to encounter high costs of backhaul provisioning, network hardware, etc. [7, 3, 5], e.g., 100K USD per square mile over 5 years for municipal Wi-Fi networks [3].

In this paper, we address the issue of reducing the cost of managed wireless networks by focusing on an often-overlooked but vital component of these networks - user authentication. The prevailing scheme for wireless user authentication, which is widely used, relies on a costly, centralized infrastructure. This scheme is costly due to two reasons:

- **Highly Available Authentication Servers**: The authentication infrastructure in any network needs to have high levels of temporal availability. In a wireless network, users are mobile and can access the network at any part of a geographically-distributed infrastructure. Thus, spatial availability of the authentication infrastructure is also necessary in these networks.
- **Highly Reliable Backhaul**: The authentication process generates additional control traffic that is usually prioritized over normal data traffic for performance reasons, thereby taking up valuable backhaul resources [3, 5], which may cost up to 400 USD per month per (1.5Mbps) T1 line [5]. Since users are mobile and can appear anywhere, the backhaul of all base stations needs to be provisioned for worst-case scenarios.

When wireless users are handed off between base stations due to mobility, some form of “re-authentication” is required. Thus, the authentication costs are exacerbated as users become more mobile [7].

In this paper, we develop a novel, distributed authentication scheme to replace the prevailing, high-cost centralized scheme. Our scheme leverages the base stations to collectively store authentication information. Thus, it eliminates the need to maintain a highly available, centralized infrastructure. Moreover, whenever possible, the authentication traffic between base stations in our scheme can be transmitted via the mobile user instead of the backhaul[1]. Thus, our scheme can successfully combat mobility-induced authentication costs. We also find that challenges related to fault-tolerance and access revocation, which are typical in distributed schemes such as ours, can likely be addressed with

[1] We use the term “user” to refer to the “device”. Unless specified otherwise, we do not assume any human involvement.

relatively little complexity. In summary, we believe that our scheme provides a good alternative to centralized authentication schemes in current wireless networks and would make these networks easier to deploy and manage. Though we do not explore inter-domain authentication [10], triggered by vertical handover, our scheme has many potential advantages in that context, too.

This paper is organized as follows. In Section 2, we provide an overview of the prevailing, centralized authentication schemes and develop the two key goals of wireless authentication. In Section 3, we develop our distributed scheme for authentication that achieves the two goals of authentication without using a separate centralized, authentication infrastructure. In Section 4, we discuss additional mechanisms that can improve the performance and robustness of the distributed scheme.

2. PREVAILING AUTHENTICATION SCHEMES

In this section, we provide an overview of the authentication scheme used in managed wireless networks including Wi-Fi (single hop as well as mesh), WiMax and cellular data networks. In these networks, mobile users connect to a base station that provides Internet access via a wired or wireless *backhaul*. All mobile users are authenticated before connecting. In general, there are two goals of authentication.

1. **Access Control**: Only authorized users can connect.
2. **Single Point of Access**: No single user can simultaneously connect to multiple base stations. This is a natural goal since more system resources are likely to be used with two connections than a single connection.

Consider GSM-based cellular networks for example. They achieve the first goal using physical authentication based on SIM cards. They ensure the second goal implicitly by making it hard to clone SIM cards.

There are a variety of authentication systems used in wireless networks today. Wi-Fi networks with an open policy or those that use a universally shared key such as a WEP-key offer limited authentication guarantees and are not the focus of this paper. Most other managed wireless networks do achieve the above two goals. For instance, some Wi-Fi networks use the IEEE 802.1x authentication framework (and WEP replacements such as IEEE 802.11i) using front-end interfaces based on SSL-based captive portals [1]. The specific authentication scheme used in a cellular network depends on the generation (2G/3G) and technology (GSM/CDMA) used in that network.

However, there are fundamental characteristics common to the authentication schemes used in most managed wireless systems. They all use a centralized, authoritative server to store/query information related to authentication, billing, and service usage. AAA [2] servers are often used for this purpose along with protocols such as RADIUS and DIAMETER. In large systems, multiple servers may share the load. To join the network, a user starts a session with a base station or an upstream aggregation point, which queries the main authentication server. These sessions are usually encrypted to thwart sniffing attacks. Users may need to be handed over from one base station to another due to mobility. Since this would cause repeated querying of the main server, many cellular data networks use an “authentication cache” at base stations or upstream aggregation points such as Packed Data Serving Nodes (PDSNs).

The high costs of wireless authentication have been quantified in prior work [7]. Alternative schemes to improve authentication performance have also been previously proposed based on initializing authentication state at base stations using predictive mobility models [8] and local caching [6, 9]. Unlike the distributed scheme we describe in the next section, none of these schemes eliminate the centralized architecture and may not always work since they rely on being able to successfully predict mobility patterns, etc. Our scheme is also different from other distributed authentication schemes such as Kerberos [4] because they do not need to enforce the second goal (single point of access).

3. DISTRIBUTED AUTHENTICATION

In this section, we develop our distributed authentication scheme to achieve the two goals - access control and single point of access.

Threat model:. We require that all base stations be trusted and under the same administrative authority, can establish secure channels among themselves, and have dependable storage. Our protocol tolerates intermittent unavailability of and unreliable message delivery between the base stations. By contrast, we completely distrust mobile users – they may forge and drop packets as they wish.

3.1 Single Base Station Scenario

We start by considering a network of only one base station. In this case, authentication only needs to verify that a user is authorized. The second goal, service from at most one base station, is trivially satisfied.

The base station has pair of private key K^{-1} and public key K. When a user signs up for wireless access, the provider issues the user with an ID id and the signature on the ID $[id]_{K^{-1}}$ using the key K^{-1}. This signature becomes the shared secret between the user and the base station. Additionally, the user also carries the base station's public key K. The following protocol authenticates a mobile user (M) to a base station (B):

1. M → B: request to join
2. B → M: n
3. M → B: $\{id, [id]_{K^{-1}}, k, n\}_K$
4. B: decrypts the message to recover id and k, and to verify $[id]_{K^{-1}}$ and n.

In the above protocol, n is a nonce, i.e., a *fresh* and unpredictable value that should be used only once, and k is a session key chosen by M. Recall also that id is the identity of M and $[id]_{K^{-1}}$ is the signature on id provided when M signs up for service. $\{\cdot\}_K$ denotes encrypting a message using the key K, which was provided when the user signed up. After B authenticates M, B and M can encrypt their traffic using the session key k.

3.2 Multiple Base Stations

Now, consider a wireless network with multiple base stations. The above scheme does ensure that only authorized users can connect. However, we need additional mechanisms to ensure our second goal, namely, single point of access. To achieve this goal, we introduce the notion of **tokens**. Each mobile user has exactly one token, which is stored at the base station where the mobile user is receiving service. The token contains the identity and other information (such as billing and usage) regarding the user. When the mobile user moves between base stations, its token moves along with the user.

3.2.1 Token Transfer

Requirements: We design our system for transferring tokens to satisfy the following requirements:

- Single point of service: a mobile user, however malicious, may get service from at most one base station at any time.
- Conservation of tokens: the token of a mobile user shall never disappear from the network, even when a token transfer is interrupted at any stage.
- Stateless base stations: a base station need not maintain any state about mobile users that have left.

States: Each mobile user M is in one of four states at each base station B:

- *noToken*: B has no information about M.
- *withToken*: B has successfully acquired the token of M and stored it in the attribute *token.* In this state, B can serve M, and can pass the token to another station at M's request.
- *sending*: B has M's token, which is stored in the attribute *token*, but is in the process of passing the token to another base station, whose identity is stored in the attribute *peer.* In this state, B cannot provide service to M.
- *receiving*: B has just acquired M's token, and is waiting for the sender, whose identity is stored in the attribute *peer*, to delete its token. In this state, B can provide service to M.

Messages: We design five messages sent between mobile users and base stations for transferring a user's token from one station (the sender station) to another (the receiver station). Each message has a *from* attribute to indicate its source, and some messages may have additional attributes as described below.

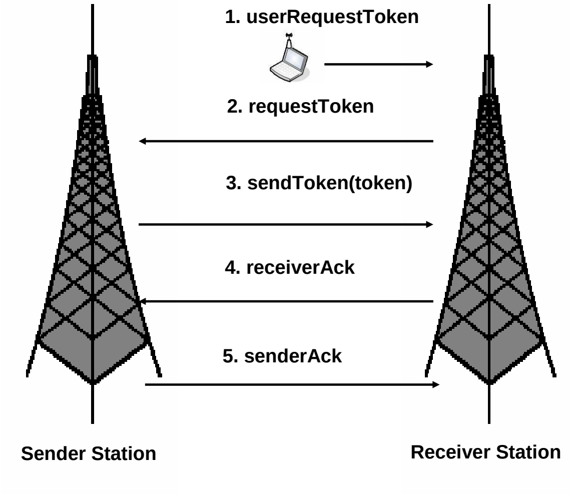

Figure 1: Protocol for token transfer under reliable message delivery.

1. *userRequestToken*: sent by the mobile user to the receiver station for requesting the token from the sender station. The attribute *peer* of this message stores the sender station's identity.
2. *requestToken*: sent by the receiver station to the sender station.
3. *sendToken*: sent by the sender station to the receiver station. The attribute *token* of this message stores the token.
4. *receiverAck*: sent by the receiver station to the sender station.
5. *senderAck*: sent by the sender station to the receiver station.

Protocol: Figure 1 shows the protocol for token transfer when message delivery is reliable. Algorithm 1 describes the complete protocol, which tolerates unreliable message delivery, at base stations. As stated earlier in the threat model, we assume that messages between base stations are sent through secure (confidential and authenticated) channels.

3.2.2 Properties

When a mobile user M signs up for service, the service stores its token at one and only one base station, where M's state is set to *withToken.* Thereafter, according to Algorithm 1, M's state at all the base stations must be in only one of the four following cases:

1. The state is *withToken* at one station, and is *noToken* at all the other stations.
2. The state is *sending* at one station, and is *noToken* at all the other stations.

```
Input: $message$, $message.from$
Optional Input: $message.peer$, $message.token$
State variables: $state,\ token,\ peer$
switch $message$ do
    case $userRequestToken$
        if $state = noToken$ then
            send $requestToken$ to $message.peer$
        else if $state = receiving$ then
            send $receiverAck$ to $message.peer$
    case $RequestToken$
        if $state = withToken$ then
            $state = sending$
            $peer = message.from$
            send $sendToken(token)$ to $message.from$
        else if $state = sending$ then
            if $message.from = peer$ then
                send $sendToken(token)$ to $message.from$
    case $sendToken$
        if $state = noToken$ then
            $state = receiving$
            $token = message.token$
        if $state = noToken$ or $state = receiving$ then
            send $receiverAck$ to $message.from$
    case $receiverAck$
        if $state = sending$ then
            $state = noToken$
        if $state = sending$ or $state = noToken$ then
            send $senderAck$ to $message.from$
    case $senderAck$
        if $state = receiving$ then
            $state = withToken$
end
```

Algorithm 1: Complete protocol for token transfer between base stations

3. The state is *receiving* at one station, and is *noToken* at all the other stations.
4. The state is *sending* at one station, is *receiving* at one station, and is *noToken* at all the other stations.

Algorithm 1 satisfies the three requirements in Section 3.2.1:

Single point of service:. Since only stations in the state *withToken* or *receiving* can serve M, M may get service from only one station in Cases 1, 3, and 4 above. Case 2 is either transient or due to message loss. Although the user gets no service in this case, he can recover it using the method described below.

Token conservation:. In each of the four cases, at least one station is in the state *withToken*, *sending*, or *receiving*, all of which have the token. Therefore, the token can never be lost no matter how the protocol is interrupted or if the mobile device is lost.

Stateless base stations:. Note that if M's state becomes *noToken* at a station B, B need not store any information about M. Therefore, in Cases 1, 2, and 3 above, only one base station needs to store M's state; in Case 4, when M is transiting from one station to another, only these two stations need to store M's state. In summary, a station maintains no state of mobile users who have left.

Robustness:. Since we do not require reliable message delivery, we designed our protocol to be recoverable from any message loss. To recover, the mobile user repeats the last message *userRequestToken* after a pre-determined timeout period. A potential criticism of this approach is that it increases the energy consumption of mobile devices. In the future, we intend to investigate this issue more thoroughly.

4. DISCUSSION

We are investigating the following additional mechanisms to make our scheme more efficient and robust.

Mobile-Assisted Token Transfer:. Our scheme eliminates the cost of maintaining authentication servers by distributing authentication information across base stations. It can also help reduce the communication overhead of authentication. Consider a mobile user handing over between two base stations. In most wireless networks, such neighboring base stations have overlapping coverage for seamless handover. Thus, the mobile user can transmit all token transfer messages, which are induced by handovers, between the two base stations. Such mobile-assisted token transfer has two key advantages. First, it eliminates the need to carry high-priority authentication traffic over costly backhaul links during handovers, which are the most common scenarios requiring user authentication. Second, since handovers inherently involve communications between the user and the base stations, the token transfer messages can be piggybacked over such communications. Hence, as users become more mobile, we expect little increase in authentication overhead.

Dirty Tokens:. Since we store tokens at base stations, a temporary base station failure would deny service to users whose tokens were stored at the failed base station, and would prevent the users from transferring their tokens to other functioning base stations. One solution is to have the user cache a copy of its token and present it to the receiver station along with the *userRequestToken* message. When the sender station fails and cannot be contacted, the receiver station stores the user's token and provides service; however, the *dirty* flag of the token is now set. The token also includes the identity of the failed base station. The dirty token is transferred in the same way as a normal token. However, when a base station receives a dirty token, it will attempt to contact all the failed base stations whose identities are stored in the token. At this time, if a failed base station has recovered, it will remove its copy of the token. Once all previous failed base stations have removed their copies, the dirty flag of the received token is cleared. This ensures that the authentication goals are satisfied before and after base station failures.

Revocation:. Our scheme can easily achieve predictable (time-based or usage-based) token revocation by including the expiration date of service or the remaining minutes/bytes of service. However, unpredictable revocation is one potential challenge in our scheme, as is typical in systems without centralized authority. We plan to mitigate this problem by periodically pushing revocation lists to base stations.

We believe that our work in this paper provides new insights into reducing the costs of wireless networks, especially due to authentication. We are currently implementing our scheme in an 802.11-based test network and intend to quantify its advantages over existing schemes.

5. REFERENCES

[1] Captive Portal. http://en.wikipedia.org/wiki/Captive_portal.

[2] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence. Generic AAA Architecture, 2000. RFC 2903.

[3] E. Griffith. Cost of Muni Wi-Fi is High, July 2006. http://www.wi-fiplanet.com/news/article.php/3518071.

[4] J.T. Kohl, B. Clifford Neuman, and T. Y. T'so. The Evolution of the Kerberos Authentication System. In *Proc. of Distributed Open Systems*, 1994.

[5] J. Larsen. Tackling Backhaul Costs, July 2006. http://www.wirelessweek.com/article.aspx?id=112174.

[6] W. Liang and W. Wang. A Lightweight Authentication Protocol with Local Security Association Control in Mobile Networks. In *Proc. of IEEE MILCOM*, 2004.

[7] W. Liang and W. Wang. A Quantitative Study of Authentication and QoS in Wireless IP Networks. In *Proc. of IEEE INFOCOM*, 2005.

[8] A. Mishra, M. Shin, N. L. Petroni Jr., T. C. Clancy, and W. Arbaugh. Pro-active Key Distribution using Neighbor Graphs. *Wireless Communications Magazine*, February 2004.

[9] H. Moustafa, G. Bourdon, and Y. Gourhant. Authentication, Authorization and Accounting (AAA) in Hybrid Ad-hoc Hotspot's Environments. In *Proc. of ACM Mobicom Workshop: WMASH*, 2006.

[10] W. Wang, W. Liang, and A. K. Agarwal. Integration of Authentication and Mobility Management in Third Generation and WLAN Data Networks. *Wireless Communications and Mobile Computing*, September 2005.