MobiCom Poster: Public-Key-based Secure Internet Access

Daniel B. Faria
dbfaria@cs.stanford.edu

David R. Cheriton
cheriton@cs.stanford.edu

Computer Science Department, Stanford University, Stanford, CA, USA

After the vulnerabilities found in the WEP protocol, providing secure wireless network access has been shown to be a complicated task. This paper describes the specification of a two-protocol architecture that provides secure, flexible, and convenient Internet access. Mutual authentication between mobile clients and access points is performed using public keys tied to domain names while confidentiality, sender authentication and replay detection are provided on a per-frame basis, using per-client dynamic session keys. Designed for the mobile environment, the architecture also provides users with mobility between networks with different address prefixes.

I. Introduction

Internet access is an important and expected amenity in many settings. For example, a typical professional employee has a laptop with an WiFi LAN card, allowing this laptop to access the Internet at work, at home and in other locations such as airports, cafes and companies the employee may visit. However, the traffic to and from his or her laptop should be secure from others even if the user is just accessing an airline web site to reschedule a next flight.

There has been some effort to provide secure wireless access, as WiFi installations are being rapidly deployed and WEP has been shown to be of little use [1, 3, 4]. However, much of the work has been placed around the IEEE 802.1X specification, which apparently encourages a diversity of solutions at the higher-level, essentially creating incompatibility between different networks.

This work describes a protocol architecture providing secure Internet access that solves the security vulnerabilities while providing ease of use, transparency, flexibility, interoperability, and mobility capabilities between networks with different address prefixes. A distinctive aspect to our approach is basing the request for access on names and public-key-level identification of the requesting host. Authentication is performed by the SIAP protocol, while the SLAP protocol provides confidentiality, integrity, and sender authentication over link-layer frames. A prototype implementation and measurements thereof indicate it is feasible to implement SIAP and SLAP with acceptable performance even without hardware support.

Compared to solutions based on the IEEE 802.1X standard, our approach extends the services provided in many significant ways. First, the specification of the SIAP protocol permits interoperability between domains, mutual authentication between the mobile client and the access points in the network, and user-transparent mobility between networks with different IP prefixes. Second, by coalescing authentication and IP address assignment, SIAP enables the implementation of different network views based on the IP address given to the client and also avoids the DoS attacks that can be performed against DHCP. As the client's IP address becomes tied to its session key, the APs can identify and block both MAC and IP address spoofing. Third, SIAP defines a client-driven state propagation mechanism that eliminates the need for an inter-AP protocol and prevents the propagation of state to access points not reachable to the client. Finally, SLAP services extend WEP services by making its services link-layer independent and providing a replay detection mechanism. The architecture briefly described here has been shown to avoid the DoS attacks reported so far [2].

II. Public-Key-based Secure Internet Access

SLAP, the *Secure Link Access Protocol*, is a protocol located just above the link layer, intercepting and processing all incoming and outgoing frames and providing a secure tunnel between the wireless host and the access point (AP). Given a per-client state consisting of MAC address, IP address, and session keys, SLAP performs its services over all outgoing frames and reconstruct frames sent by its peer entity. SLAP services include encryption, per-packet authentication, and replay detection. In order to set up this per-client state in

both the client's laptop and in the neighboring access points, an application-layer authentication protocol is used, called *Secure Internet Access Protocol* (SIAP).

SIAP is responsible for providing the authentication service using RSA public keys. The SIAP client present in a laptop performs a three-message handshake with the SIAP server in the access point. This handshake provides mutual authentication and provides the client with the IP address to be used and the session keys associated with it. From this point on, the client's MAC and IP addresses and the session keys are tied together, and its correct use is enforced by the SLAP entity in the access point.

The SLAP module waits for the SIAP entity to perform the authentication process and inform it about the security state to use. After that, all the frames sent between client and AP receive the SLAP services with the session keys just established. By placing SLAP over the link layer, we make it technology independent, being suitable for IEEE 802.11 as well as Ethernet networks. SLAP uses AES in CTR mode and HMAC-MD5 to provide the confidentiality and message authentication services, respectively. A replay detection mechanism is also implemented, using an authenticated counter present in the SLAP header.

The mutual authentication provided by SIAP depends on the ability of clients and servers to verify signatures over public keys. The ideal solution, necessary to achieve complete interoperability between domains, is to have a deployed public-key infrastructure (PKI). However, with the lack of such mechanism, our architecture can be implemented locally on a network provided with a single-domain certification authority (SDCA). In this case, all the mobile computers have their names tied to public keys signed by the local SDCA and know its public key, needed to authenticate the local access points. Using SDCAs, multi-homed users need a signed public key for each network they are wiling to use.

By the end of the SIAP handshake, the client receives a *ticket*, a piece of data signed by the AP that enables the client to prove to other access points that it has already been successfully authenticated. As the client moves, it propagates its security state by sending the ticket to other APs in the network. Using a second, shared key pair, the APs test the validity of the ticket and configure the client state. The advantages of this client-driven state propagation are twofold. First, as the client is responsible for propagating its security state, there is no need for an inter-AP protocol. Second, this mechanism avoids the propagation of state to access points that are never used by the client.

III. Experiments

Both SLAP and SIAP have been implemented in Linux 2.4. Our testbed is composed by a laptop that works as the client and a desktop computer that plays the role of the access point. The laptop is a 333-MHz Pentium II computer, with 64 MB of main memory, and a FastEthernet 100Mbps card. The desktop computer runs with a 900-MHz Duron processor, 256 MB of memory, and contains two FastEthernet network interfaces. The FastEthernet cards were used to connect the client to the access point and emulate a wireless link with higher bandwidth.

The measurements show that the operations involving an RSA private key are very demanding, incurring overheads in the order of tens of milliseconds. The first consequence of these high costs is that the authentication handshake takes in the order of 400-600ms to finish. This delay may affect the throughput on the SIAP server, but may have no impact on how smoothly the SIAP client switches from one network to another, as it can authenticate with the second network and get a second IP address before it performs the handoff and possibly deletes its state in the previously used network.

The overhead incurred by SLAP in each direction (adding the processing time at both the client and the AP) varies between 50μs and 460μs in our current test bed. This means that the round-trip time (RTT) between the laptop and a server in the Internet can be increased by up to almost 1 millisecond for large frames. As small packets are predominant in local wireless networks [5] and measurements performed in Internet backbones show that 175- and 400-byte average packets are common [6], we expect this 1-millisecond increase to be rare.

To quantify the impact of this RTT increase over real applications, we performed several long (50 MB) file transfers using FTP. When using a server with a RTT of 1 ms from the wireless host, the total download time was increased by 17%. This increase drops to 7% when using a 40ms-away server, which we believe to be a more representative scenario. We expect SLAP services to incur an even smaller overhead as code optimizations are performed and no noticeable overhead as hardware implementations are used.

IV. Conclusion

Secure Internet access is an important facility to provide as the use of mobile Internet devices increases. SIAP provides a simple protocol solution that is efficient, secure, flexible and convenient for the end user.

It avoids the denial-of-service and security openings that are problematic with DHCP. SIAP and SLAP allow relatively simple layer 2 devices while ensuring security of access. The name basis for identification allows a site to assign IP addresses to newly arrived hosts to classify them as visitor or employee and then tunnel packets accordingly.

SIAP and SLAP provide an attractive alternative to the approaches to secure access than have been attempted with 802.11b, including WEP and 802.1X. They are either insecure or inflexible and both seem to require a comparable amount of mechanism in the access points to the architecture described here. Moreover, SIAP/SLAP use AES-based encryption, proven PKE technology and higher-level protocol design, avoiding the security weaknesses that have plagued link-level efforts. The performance results presented show that a software implementation is viable to be used as a temporary solution, obtaining performance suitable for current 11Mbps wireless networks.

References

[1] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: The insecurity of 802.11. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking - Mobicom'01*, pages 180-189, July 2001.

[2] D. B. Faria and D. R. Cheriton. DoS and Authentication in Wireless Public Access Networks. In *Proceedings of the First ACM Workshop on Wireless Security (WiSe'02)*, pages 47-56, Sept. 2002.

[3] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In *Eighth Annual Workshop on Selected Areas in Cryptography*, August 2001.

[4] A. Stubblefield, J. Ioannidis, and A. D. Rubin. Using the Fluhrer, Mantin, and Shamir attack to break WEP. Technical Report TD-4ZCPZZ, AT&T Labs Research, Aug. 2001.

[5] D. Tang and M. Baker. Analysis of a local-area wireless network. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking - Mobicom'00*, pages 1-10, Boston, MA, USA, Aug. 2000.

[6] K. Thompson, G. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network*, 11(6):10-23, Nov. 1997.