“Privacy and Authentication for Wireless Local Area Networks”
Ashar Aziz and Whitfield Diffie
Summarized by Mark Stemm

stemm@cs.berkeley.edu
Overview

- Make the wireless link no worse than the wired link by adding authentication and privacy of the wireless link.
- Implemented as a new level in the protocol stack at the link level.
Design Goals

Many existing applications assume some degree of security in the wired network which isn’t in wireless network.

Soln: *Make the wireless no worse than the wired.*

Can’t do it at the application layer or at the transport layer, because you would have to change too many hosts --> do at link layer.
Design Overview

- Public key system (including Certification Authorities (CAs)) for authentication, shared-key system for privacy.

- Message #1: Mobile --> Base: {Cert_Mobile, CH1, List of SKCS}
  - Cert_Mobile: Here’s my public key (and a way to prove it is valid).
  - CH1: challenge value.
  - List: So different shared-key algorithms can be supported.
Design Overview (contd.)

- Message #2. Base-->Mobile
  - \{Cert\_Base, E(Pub\_Mobile, RN1), Chosen SKCS, Sig(Priv\_Base, \{E(Pub\_Mobile, RN1), Chosen SKCS, CH1, List of SKCS \})\}
    - Cert\_Base: Here’s my public key (and a way to prove it is valid).
    - E(...): Base’s half of shared-key.
    - Sig(..): Proof that the base received the message and responded to the challenge.
Design Overview (contd.)

● Message #3: Mobile--->Base
  – \{E(\text{Pub\_Base}, \text{RN2}), \text{Sig}\{\text{Priv\_Mobile},
  \{E(\text{Pub\_Base}, \text{RN2}), E(\text{Pub\_Mobile}, \text{RN1})\}\}\}\}
  • E(\ldots): Mobile’s half of the shared-key (the final key used will be \(\text{RN1 XOR RN2}\)).
  • \text{Sig}\{\ldots\}: Proof that the mobile actually received message #2 from the base.
Data Packet Issues

- Need to know which packets were lost (i.e. by using sequence numbers) so one can use the correct part of the shared key.
- All packets also have integrity--a checksum inside of the encrypted part.
- Playback attacks are possible (but only for datagram protocols, where multiples are possible anyway).
Key Change Protocol

● If Base initiates it:
  – 1. Base-->Mobile: Signed(Priv_Base,
      {E(Pub_Mobile, New_RN1), E(Pub_Mobile,
      RN1)})
  – 2. Mobile-->Base: Signed(Priv_Mobile,
      E(Pub_Base, New_RN2), E(Pub_base, RN2)})

● By including old keys, allows protection against replay attacks w/o sequence #s.
Operation w/ Multiple CAs

- A single CA is not scalable...
- So use a hierarchy of CAs...
- Which has been created by other people.
Criticisms

● Shouldn’t we really do it at the application layer anyway?
  – “No worse than the wired network” is proving to be not good enough.

● Setting up all of these CAs might be just as much work as changing the applications.

● ??