Security is not Cryptography!

- Is your message “secure” if it doesn’t get there? (e.g., denial of service)
- Security is adherence to a security policy
- Unfortunately, in many systems policy is informal, defined in *ad hoc* manner, and focused only on selected attacks
- NB: Attacker may not agree on selection
Network Infrastructures

- Shared, so virtualization matters
- Need timing, privacy and authentication
- Focus must be on protection of the network elements (what will be programmed), in spite of improved flexibility
- Node security, then network security
Penn/Bellcore SwitchWare Project: A Language-Oriented Model

- Switchlet Language for users (SL)
  - formal semantics restrict programs
  - e.g., Prog. Language for Active Nets (PLAN)

- Wire Language for communicating (WL)
  - formal semantics across boundaries
  - Java or Caml bytecodes

- Infrastructure Language for Virtual Machine (IL)
  - formal semantics supported on metal: run-time

See http://www.cis.upenn.edu/~switchware
Current Software

Active Bridging

See http://oilhead.cis.upenn.edu/~salex
AEGIS Secure Bootstrap

- Integrity Guarantees for Dynamic Integrity Checking (http://www.cis.upenn.edu/~waa)

Diagram:
- Trusted Repository
- AEGIS ROM
- POST
- O.S. (BSD)
- SwitchLets
- ROMs, Boot Block, ...
- BIOS Sec. 1
Secure Active Network Element (SANE)

- "Trust, but Verify" (U.S. Nuclear Policy..)
- PLAN
- Caml/O.S.
- AEGIS
- Node-Node Authentication
- Recovery
- Dynamic Integrity Checks (Maybe per-packet/SwitchLet?)
- Static Integrity Checks (Done Once)

http://www.cis.upenn.edu/~waa
http://www.cis.upenn.edu/~angelos
Restricting Programs

Node safe versus network safe

- Node Safe Programs
- Network Safe Programs