



# SECURE ACTIVE NETWORK ENVIRONMENT (SANE)

*"TRUST, BUT VERIFY"*

*OLD RUSSIAN SAYING*

SCOTT ALEXANDER

BILL ARBAUGH

ANGELOS KEROMYTIS

JONATHAN SMITH

UNIVERSITY OF PENNSYLVANIA

# 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

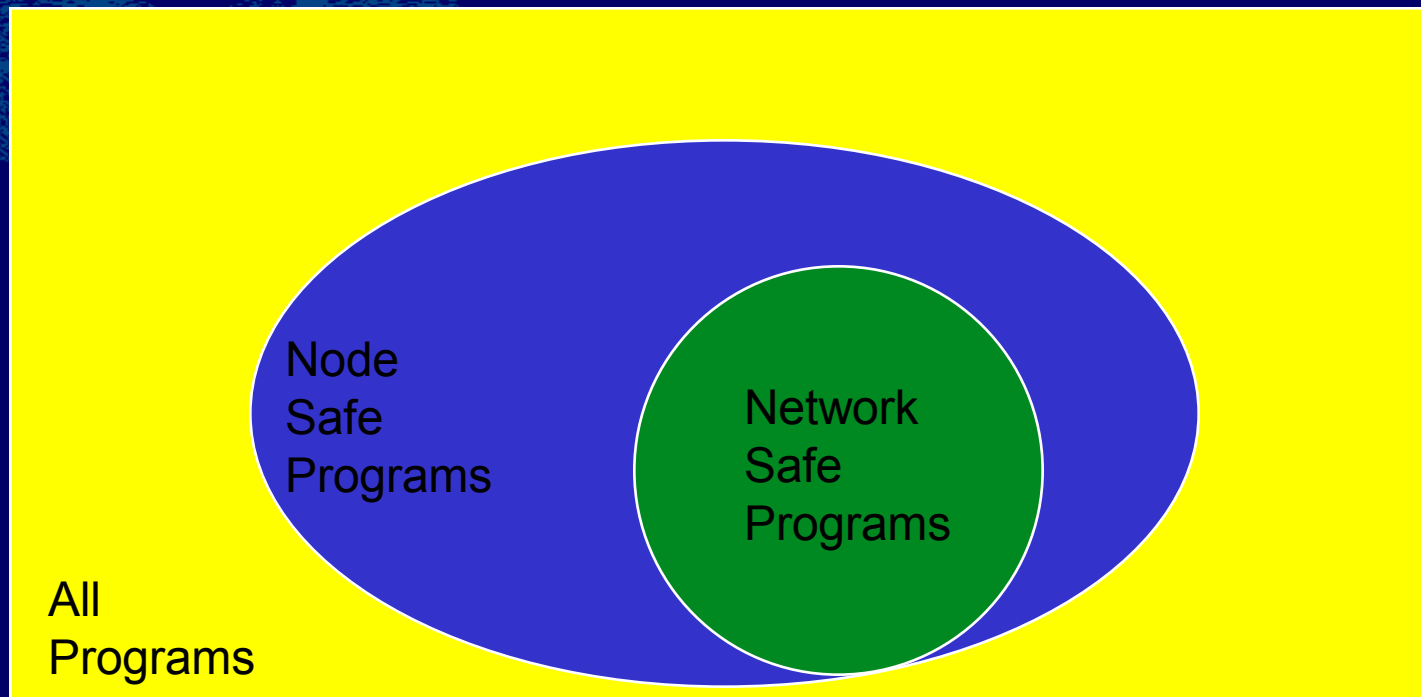
# Security is not Cryptography!



- ❑ 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 Differ on Selection...

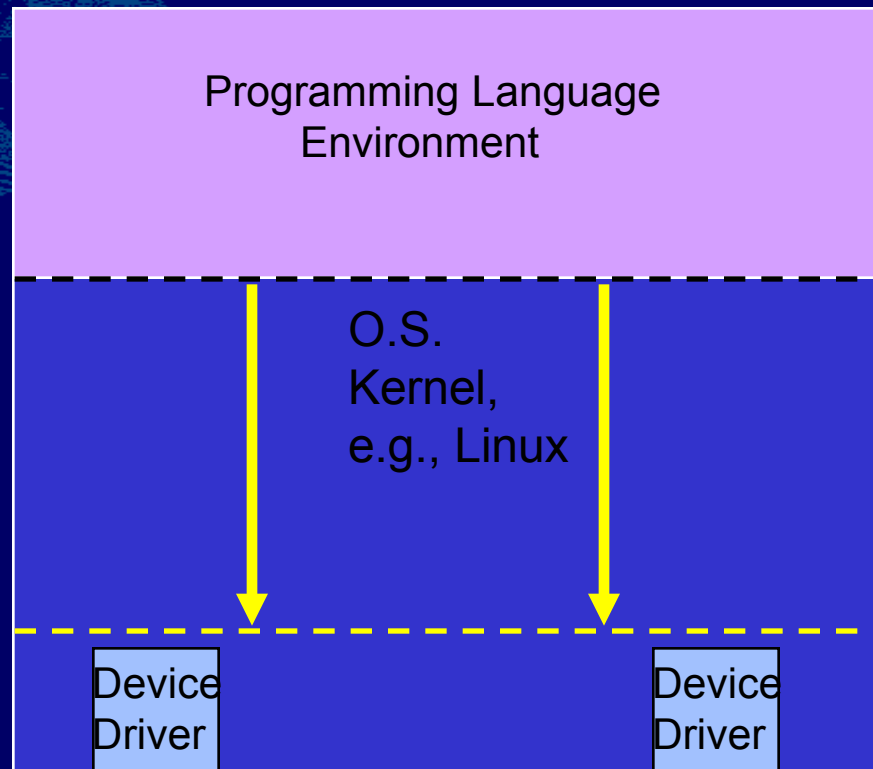
# Restricting Programs

## □ Node Safe Versus Network Safe



# How Do We Control Programs?

□ Safety & Security: P.L., O.S. or Hybrid?



# A Language-Oriented Model

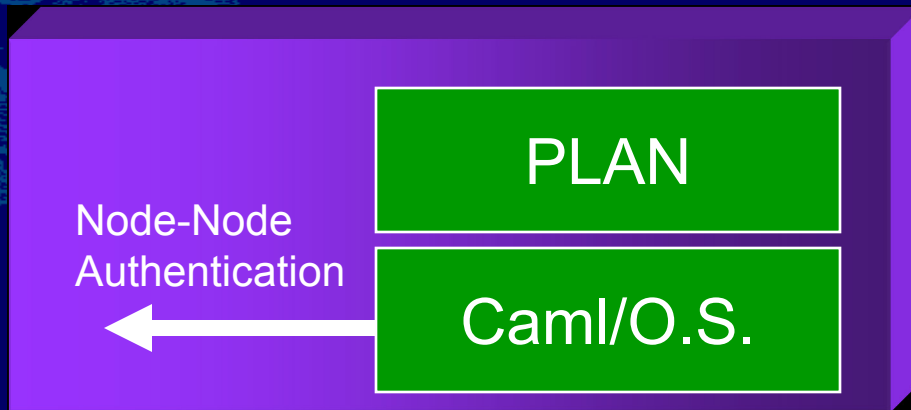


- Switchlet Language for Users (SL)
  - Formal Semantics Restrict Programs (e.g., Packet Filters use regexps)
- Wire Language for Communicating (WL)
  - Formal Semantics Across Boundaries
- Infrastructure Language for Virtual Machine (IL)
  - Formal Semantics Supported on Metal: Run-time

# Secure Active Network Environment (SANE)

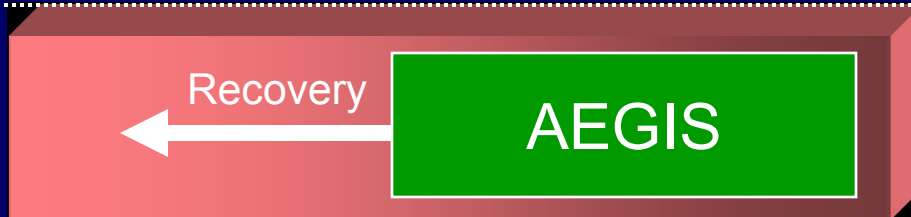
□ Again, “Trust, but Verify”!

NETWORK  
LEVEL



Dynamic Integrity  
Checks (Maybe per-  
packet/SwitchLet?)

NODE  
LEVEL



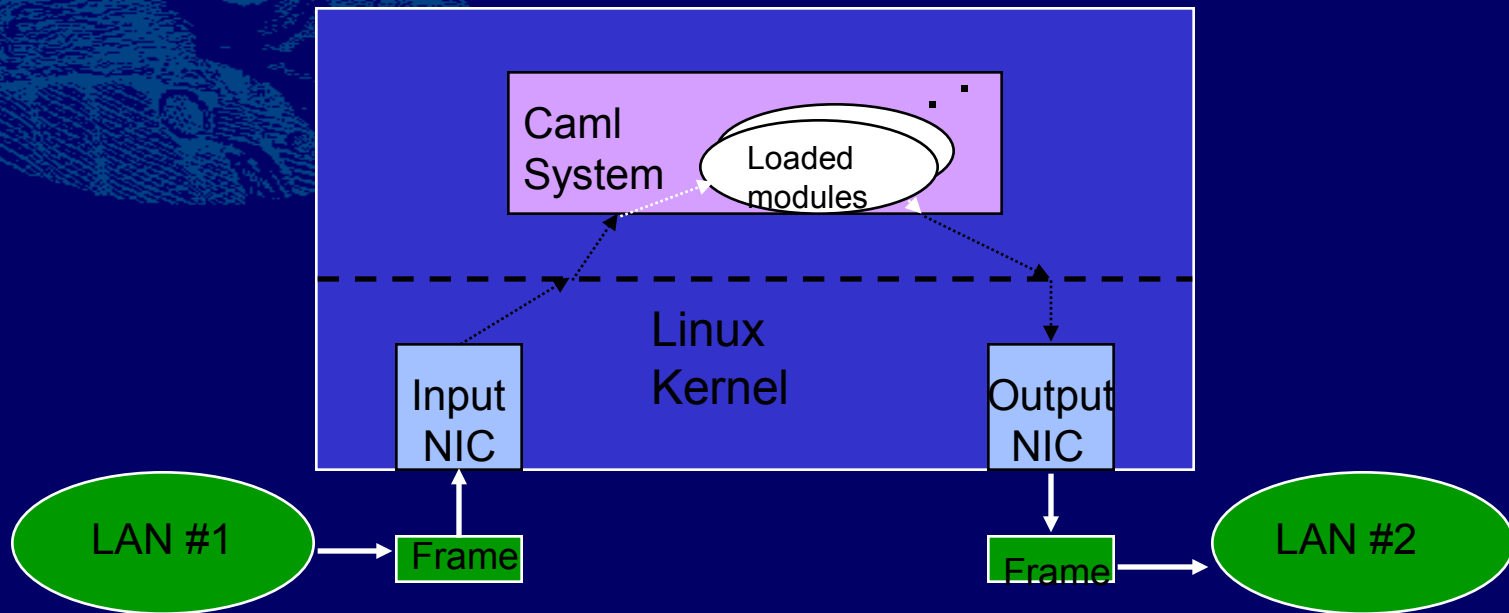
Static Integrity  
Checks (Done  
Once)

<http://www.cis.upenn.edu/~waa>

<http://www.cis.upenn.edu/~angelos>

# Per-module/Per-packet Integrity Checking

## Active Bridging (Scott Alexander)

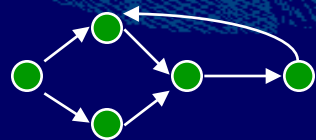


<http://oilhead.cis.upenn.edu/~salex>

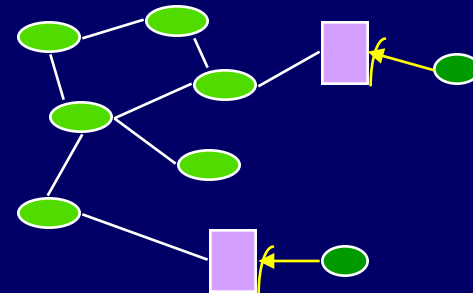
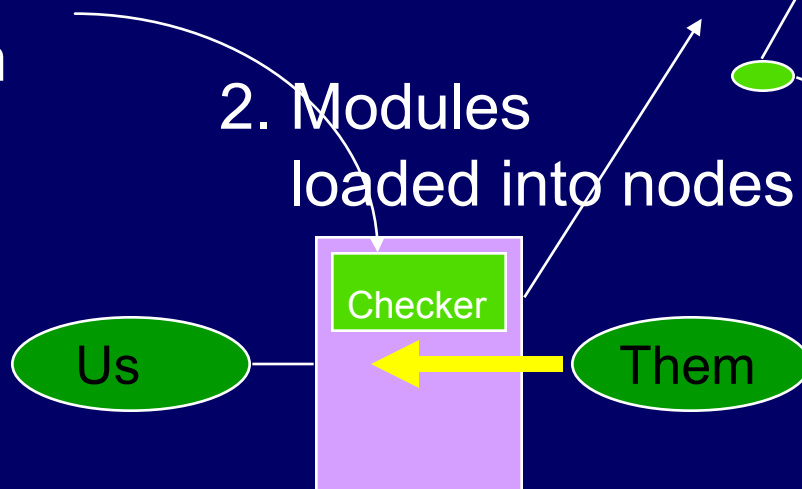


# REAL Security: Model to Actions and **NOTHING ELSE!**

- Syntax, Semantics, Node vs. Network
- Example: Securing a Network



1. System Model



3. Resulting in a robust Network

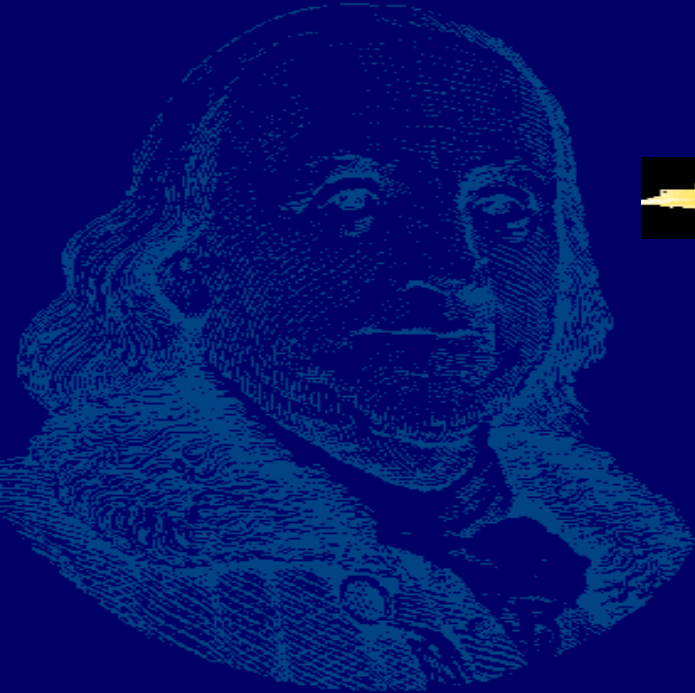
# The Node Problem

---



- Every Computer System is Currently Invoked by an Untrusted Process- Even “Secure Systems” .
- This Leads to a False Sense of Security for the Users of those Systems.

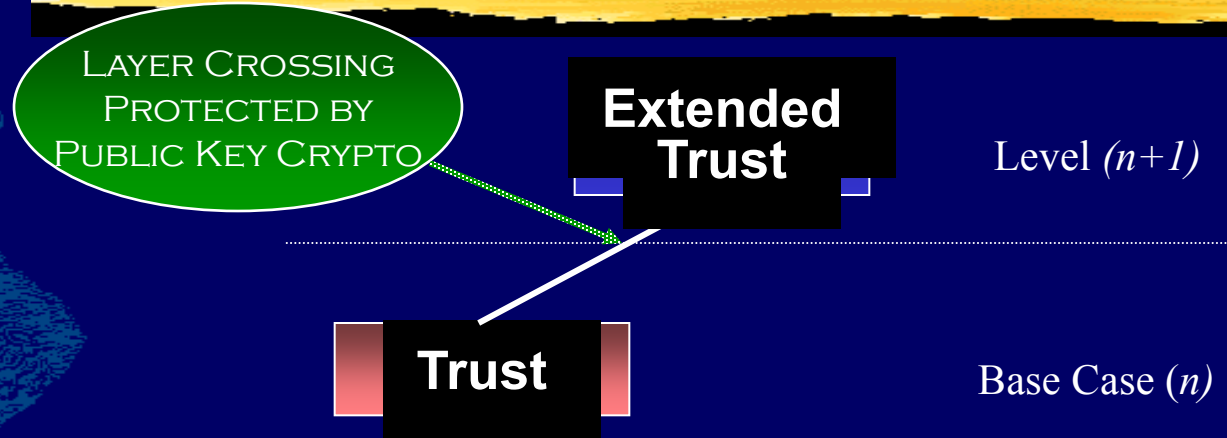
# Definition



□ We Define the Guaranteed Secure Bootstrap of an Active Network Node in Two Parts.

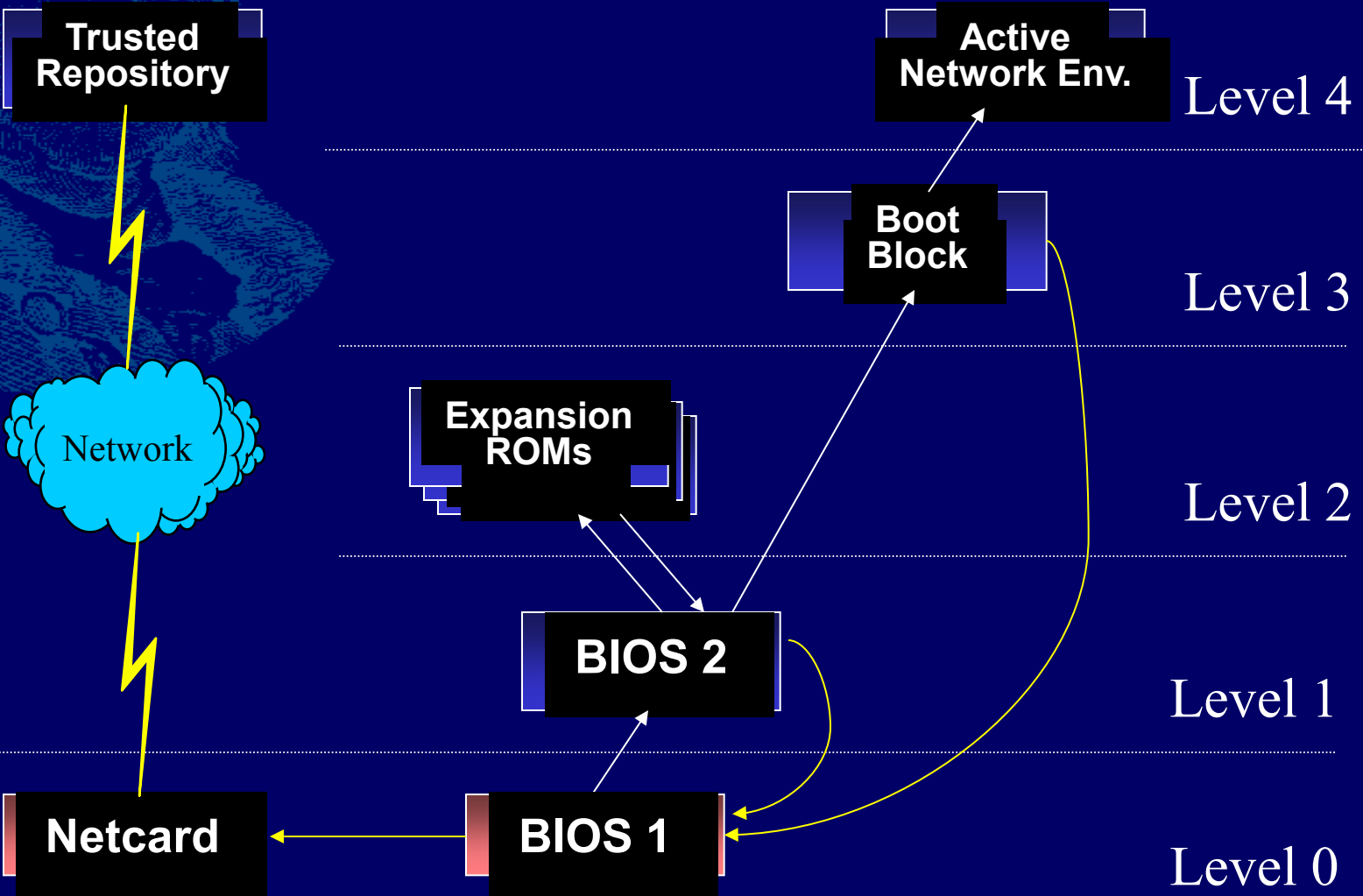
1. No Code is Executed Unless Explicitly Trusted or its Integrity is Verified Prior to Use.
2. When an Integrity Failure Occurs, There Exists a Method to Recover a Suitable Replacement.

# Approach

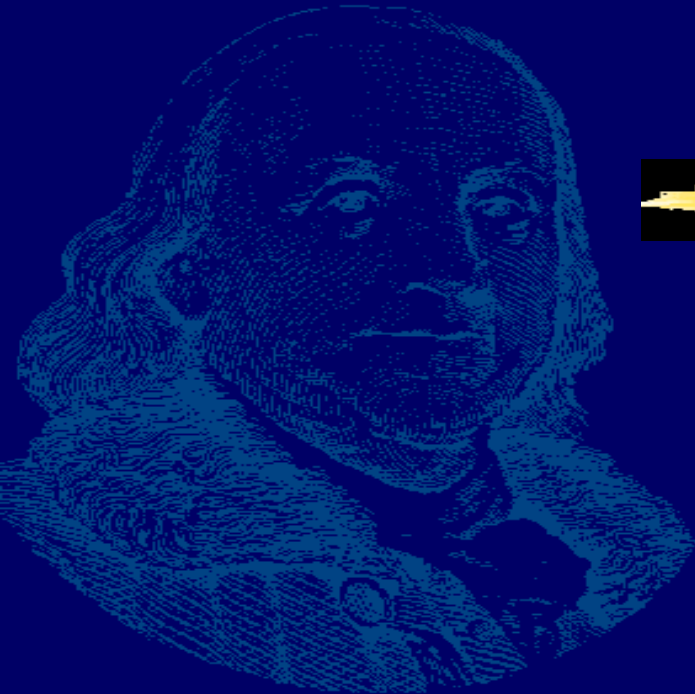


- Integrity and Trust Must be “*Grounded*” at the Lowest Possible Point.
- Chaining Layered Integrity Checks (CLIC) Extends Trust Beyond the Base Case.

# AEGIS Architecture



# Previous Work



## Previous research on the Secure Bootstrap Problem

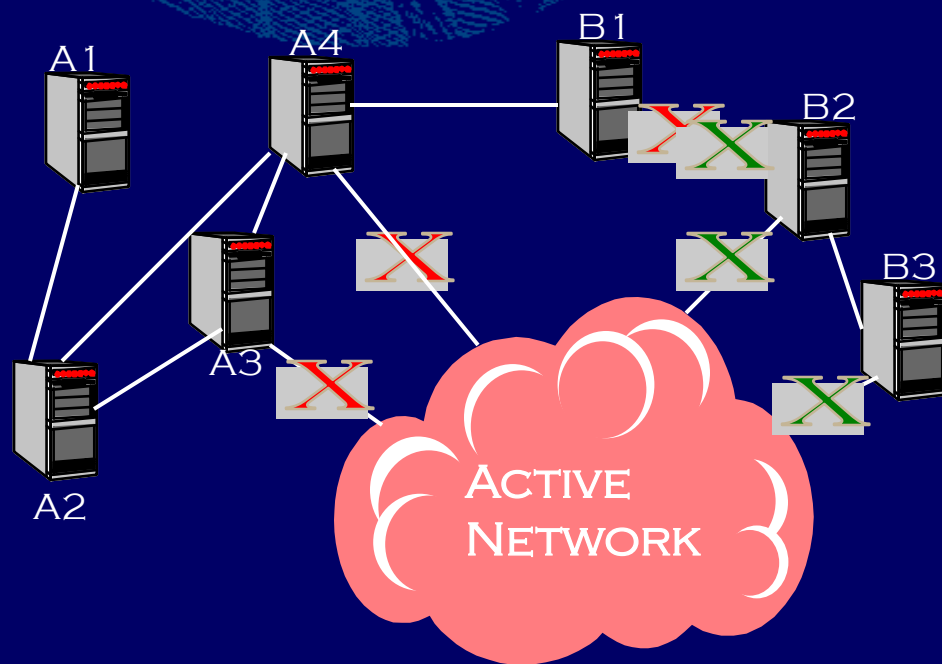
	SECURE?	PROTOTYPE?
YEE	YES / NO	NO / YES
RATBAG	NO	YES
LAMPSON / BIRLIX	NO / NO	NO / YES
ARNOLD / JABLON	NO / NO	?? / ??
SUN	PROBABLY	YES
BITS	NO	YES

# The Network Problem



- ❑ Network of Mutually Suspicious Active Nodes
- ❑ Nodes Need to Cooperate for the Network to Function
- ❑ Network Users Need to Interact with the NEs in a Controlled Manner
- ❑ Different from the Current Internet!

# Mutually Suspicious Nodes



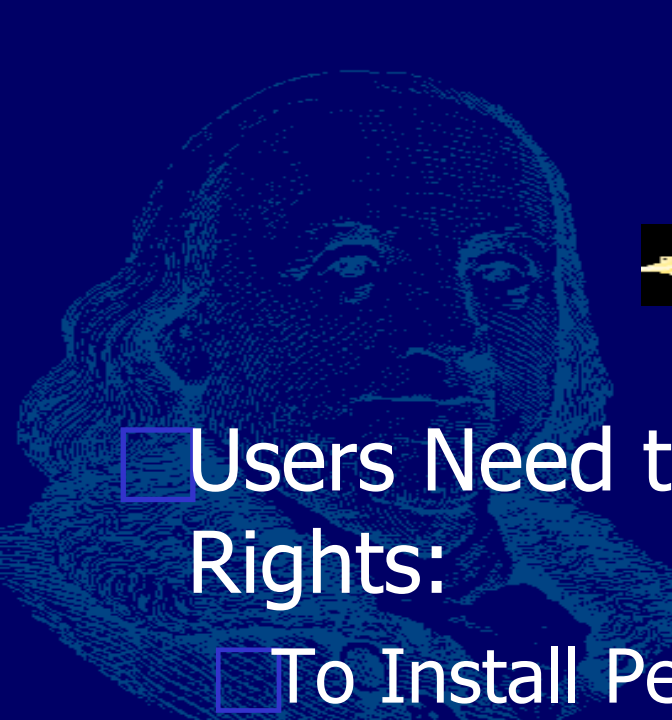
- Nodes Authenticate their Neighbors
- Establish Trust Relations with Peers (PolicyMaker?)
- Use Trust Relations to Solve Existing Problems (eg. Routing)
- Optimize Authentication



# Node to Node Authentication

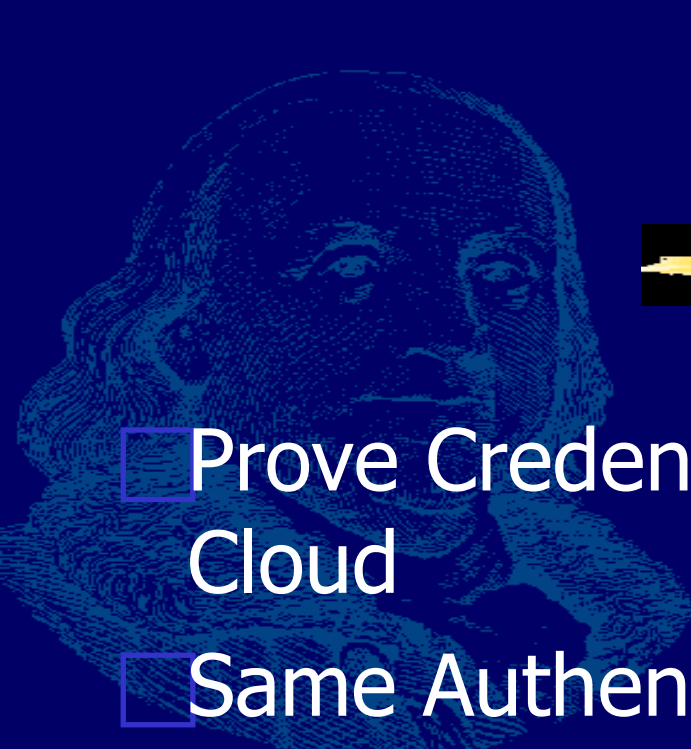
- ❑ Once at Boot Time, Periodically Thereafter (Crypto “heartbeat”)
- ❑ Modified STS Protocol (Well Known and Understood)
- ❑ Key Can be Used to Authenticate on a Hop-by-Hop Basis, Encrypt Sensitive Information
- ❑ Make Traffic Analysis Hard

# User to Node Authentication

- 
- Users Need to Prove Resource Usage Rights:
    - To Install Permanent Services
    - To have their Packets Identified for Further Processing
    - Perform other Privileged Operations
  - Authentication in a “Telescopic” Manner (“scout” packets)
  - Again, use of a Modified STS Protocol

# Make Use of Established Trust

---

- 
- Prove Credentials Once per Administrative Cloud
  - Same Authentication Inside that Cloud
  - Cross-Domain Authentication Acceptance Subject to Policy (Credential Forwarding, Session Key Sharing)
  - We Still Need Language Safety (Accidents Happen)

# Open Problems



- Public Key Infrastructure Needed
- Malicious Nodes and Byzantine Failures
- One Way Authentication
  - Negotiation too Costly in Some Cases (?)
  - Credential-Use Prediction ?
  - Protect Against Replay ?
  - Do We Need Synchronized Clocks ?

# SwitchWare: Accelerating SECURE Network Evolution!



- Active Nets: changing the “tempo” of network evolution from political to technological with programmable architecture*
- Secure Active Network Environment (SANE) Architecture: Moving from Secure NODES to Secure NETWORKS*
- Security by design, not afterthought!*

<http://www.cis.upenn.edu/~switchware>