Security for Distributed and Co-operative Computing Networks

Mahalingam Ramkumar
Assistant Professor
Department of Computer Science and Engineering
Mississippi State University
Overview

- Motivation
- Broad research goals
- Ongoing research in ad hoc network security
Co-operative Synergy

Many applications rely on the synergy of co-operation
- to perform tasks simply not possible otherwise
  - multi-hop mobile ad hoc networks
  - autonomic computing
- Require strategies to ensure that participants will co-operate in a fair manner
  - or adhere to rules
Byzantine Generals

The basic framework for co-operative and distributed computing applications

How do we identify “moles”

How can the application be resilient to non-cooperating nodes?
  - Can the application still meet the desired goal even in the presence of bad participants?

What are the set of rules that each node should adhere to?
The Two-Agent Approach

- A network of devices
  - Some associated with an end-user
- For many applications the device has a single purpose:
  - Serving the end user
- Devices taking part in distributed applications have dual purposes
  - Serving the end user
  - Serving the network
User and Network Agents

- Easier to think of each device as two entities:
  - User agent & Network agent
  - With often conflicting goals
  - End user has motivation to “preserve the health” of the user agent
  - No motivation to protect the network agent
    - Can even take gain personally by circumventing the network agent
User and Network Agents

- Network agent operates in a hostile environment
  - Primary goal: reduce the “degrees of freedom” of the user agent (to violate rules)
- The user agent could be a PDA
- The network agent could be
  - A secure co-processor
  - A trusted computing base
  - Obfuscated software running on the PDA
Constraints on Agents

- Need for tamper-proof boundary places significant constraints on the network agent
- Keeping the cost of network agent low
  - inexpensive strategies for tamper-proofing
  - very low complexity inside the trust boundary
    - eliminate proactive approaches for heat dissipation
    - unconstrained & low-cost shielding
- Very few constraints on user agent
Broad Requirements

- Securing co-operative applications
  - Very low complexity network agent
    - Low complexity secure co-processor
      - comparable in complexity to RFID chips
  - Utilizing the low complexity network agent to enforce adherence to rules
    - How do we reduce the “degrees of freedom” of the user agent to violate rules?
    - How can the resource constrained network agent *safely* utilize the resources of the user agent?
Trustworthiness

- Trustworthy computing module provides two assurances
  - The software run by the module cannot be modified
  - No one can impersonate such a module
  - Translates to
    - read-proofing
    - write-proofing
Read-proofing and Write-proofing

Read-proofing

- Secrets assigned to the module cannot be exposed
  - Secrets are used to establish cryptographic security associations (for authentication)
  - Also a stepping stone for write-proofing
    - Software can be authenticated using the secrets protected (to prevent modification)
    - But what about the software that verifies the integrity of the software (using secrets)?
Write-proofing

- Providing an assurance that the code executed by the module cannot be modified
- Well known that this can be achieved with two basic assurances
  - Read-proofing
  - Write-protected counter
Security Associations

- An SA between A and B provides both A and B assurances of
  - each other's identities
  - the ability to exchange messages protected from all other entities
  - for example, a shared secret
- SAs can be
  - One-to-one
  - One-to-many
  - Many-to-many
Network Scale

- The number of nodes that may be taking active part in a particular network
- Number of nodes that are assigned secrets
- Maximum number of nodes that could be assigned secrets
  - size of ID-space
- Solutions for small networks are not very difficult
Solutions for Large Networks

- Two predominant approaches
- Public key schemes in conjunction with a public key infrastructure
- ID based schemes with one or more escrows
Certificates Based Schemes

- Private keys chosen by nodes
  - thereafter corresponding public keys are calculated
  - public keys are seemingly random
    - provide no information about the entity
    - have to be certified by a certificate authority

PKI for very large scale networks
- Hierarchical certificate authorities
- Need to exchange chains of certificates
ID based Schemes

- Meaningful public values are assigned to nodes
- Node IDs could be secure one-way functions of meaningful public values
- Corresponding to the public value (or node ID) secret(s) are provided to the node by one or more escrows
- No need for certificates
ID-based Schemes

- Symmetric key schemes are by default ID based
  - But require storage of the order of network size
  - Does not cater for one-to-many security associations
- ID based public key schemes also exist
  - Each node needs to store only one private key
  - Computationally expensive
Key Pre-distribution

- Unlimited network size
- But security limited by the number of colluding nodes
- Storage requirement
  - proportional to the number of colluding nodes that can be tolerated
- We require unrestricted network size
  - Is it reasonable to expect that not more than a finite number of nodes can pool together secrets?
Key Pre-distribution

- We require unrestricted network size
  - Is it reasonable to expect that not more than a finite number of nodes can pool together secrets?
- Resistance to collusions of one million nodes?
  - Attacker has to expose all secrets from one million nodes to make any kind of dent in security
- Storage proportional to one million keys
  - in practice a few tens of millions – a few hundred MBs
  - How about computational complexity?
Random Key Predistribution

- Arbitrarily low computational complexity
- Security limited only by storage
- Lower computational complexity
  - Increased ability to protect secrets
    - Makes things more impractical for an attacker!
- A rich variety of security associations
Ongoing Research

Different pieces of the puzzle
- Inexpensive strategies for trustworthy computing
  - DOWN – Decrypt only when necessary policy
- Very low complexity key distribution schemes suitable for low complexity ScPs
  - KPI – Key pre-distribution infrastructure as an alternative to PKI
- Securing ad hoc routing protocols
  - With the two-agent approach
Decrypt only when necessary
- Relies on the ability to operate with *fractional* parts of secrets
- Applicable to all conventional public key schemes
  - eliminating the need for multi-step countermeasures for safe zeroisation
    - reduces complexity of tamper-proofing mechanisms
KPI

- Key Predistribution Infrastructure
- A set of low complexity key distribution schemes
  - Very low complexity inside the trust boundary
    - few symmetric cipher operations
  - Rich set of security associations
    - one-to-one, one-to-many and many-to-many
- Renewable
- Hierarchical extensions
- High scalability, ID-based
KPI

- Modeled in a manner similar to PKI
  - Or the Internet
- Can consist of many independent deployments (subnets)
  - Deployments can be merged together
  - To form the Internet
  - And seamlessly unmerged under large scale security breaches
- A “KPI entity” is a very low complexity ScP
  - Analogous to “end entities” in PKI
KPI

- Utilizes a set of key pre-distribution schemes (KPS)
  - KPSs can take advantage of
    - external untrusted storage resources
    - external computational resources
    - the DOWN policy
  - to realize substantial improvements to collision resistance
- Also supports mutual authentication of “external entities” (who do not need to employ an ScP) with core KPI entities
Ad Hoc Networks

- Task sharing between ScP and mobile node
  - shared execution of routing protocol
    - Network agent limited to symmetric cryptography
    - User agent (PDA) can perform asymmetric cryptography
- Apalla – An extension of Ariadne (for securing DSR)
- A new design constraint for secure routing protocols