The Quest for Security in Mobile Ad Hoc Networks

Hubaux, Buttyan, Capkun
Swiss Federal Institute of Technology

Imran Shah

Overview

- Introduction
- Threats to Mobile Ad Hoc Networks
  -Basic Mechanisms
  -Security Mechanisms
- Protection of Basic Mechanisms
- Protection of Security Mechanisms
- Self Organized Public Key Infrastructure
- Conclusion

Introduction

- Focus on security mechanisms related to mobile ad hoc networks
- *Self-organization* ability of mobile ad hoc network to function without external management or configuration

Threats to Mobile Ad Hoc Networks

- Two levels of attack:

-Basic Mechanisms
Physical security, routing, security of medium

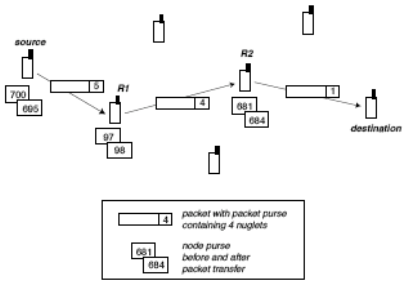
-Security Mechanisms
Attacks against cryptography used

Protection of Basic Mechanisms

- Physical Security - Tamper Resistance
- Master Slave Relationship - System Imprinting
- Routing Based Mechanisms – Solution of watchdog and pathrater

Protection of Basic Mechanisms

- Service availability – Seek cooperation and prevention of selfishness
- Virtual Currency as a solution: *Packet Purse Model* and *Packet Trade Model*



Protection of Security Mechanisms

- Focus on key establishment
- Assume no authority and no fixed server
- Desire to use asymmetric cryptography but there are problems:
  -Key revocation
  -How do you trust the public key of a node?

Eliminating Need For Central Authority

-Emulate certification authority – Public key of CA is known to every node and private key is divided amongst n nodes
Threshold cryptography (n, t+1) scheme where (n >= 3t+1)

-Key Agreement – Sharing of a prior context allows for derivation of strong shared key

-Self Organized Public Key Infrastructure

Self Organized Public Key Infrastructure

- Certificates stored and distributed by users
- Each user maintains local repository of public key certificates that consists of two parts
  -certificates issued
  -set of selected certificates issued by others in system

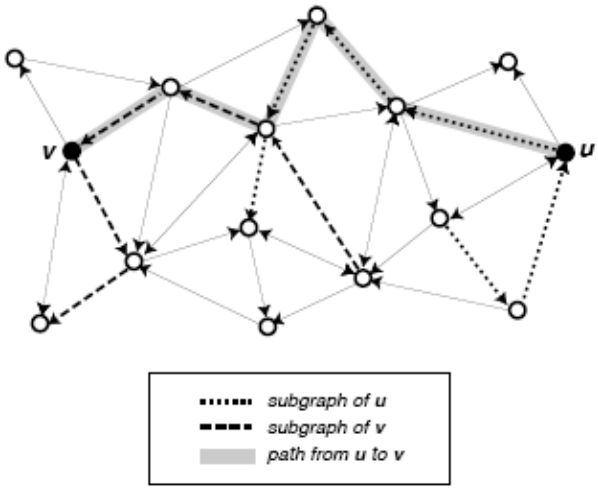
Model and Framework

- Relationships modeled as directed graphs *G (V,E)* called trust graph
- A directed edge from a vertex *u* to another *v* exists if *u* issued a public key certificate to *v*
- Certificate chain from *u* to *w* is a directed path from *u* to *w* in *G*
- Reachability, the existence of a directed path from *u* to *w* denotes the existence of a certificate chain from *u* to *w*

Model and Framework

- Subgraph selection algorithm *A* used to build subgraph
- Execution of *A* on *G* by *u* resultant subgraph is $S_A(G,u)$
- Union of $S_A(G,u)$ and $S_A(G,v)$ is $S_A(G,u,v)$ and $S_A(G,u,v) = S_A(G,v,u)$

Model and Framework



Performance

- Ratio of number of user pairs (u,v) where there is a directed path from u to v in the merged subgraph to the number of user pairs (u,v) where there is a directed path from u to v in the trust graph

$$p_{\mathcal{A}}(G) = \frac{\#\{(u,v) \in V \times V : u \rightsquigarrow_{S_{\mathcal{A}}(G,u,v)} v\}}{\#\{(u,v) \in V \times V : u \rightsquigarrow_{G} v\}}$$

Design Objectives of Subgraph Selection Algorithm

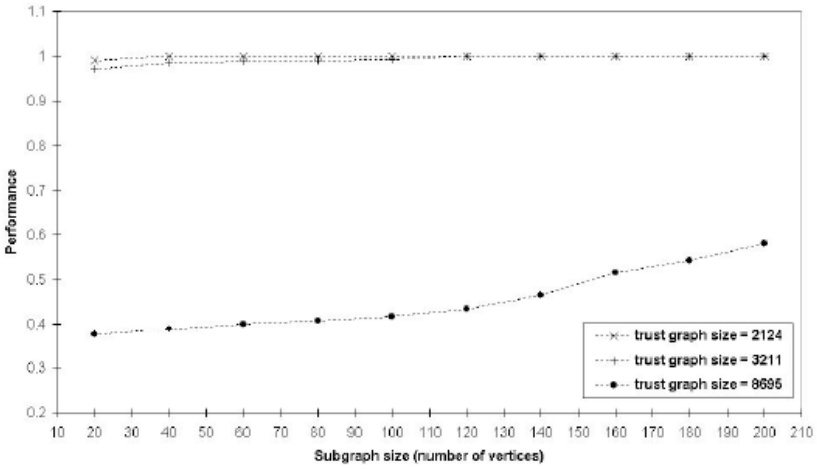
- Performance
- Scalability
- Distribution
- Robustness

Shortcut Hunter Algorithm

- Shortcut – an edge, upon whose removal, the shortest directed path between the two nodes previously connected by the edge becomes strictly larger than two.
- Algorithm selects a subgraph with one out-bound and one in-bound path

1. Initialization: $V(S) := \{u\}$, $E(S) := \emptyset$, $N := \emptyset$, $w := u$, $i := 0$
2. $T := \{(w, z) \in E(G) : z \notin V(S) \text{ and } z \notin N\}$
3. If $T = \emptyset$, then *backtracking*:
   (a) If $w = u$, then go to step 9
   (b) Add w to N
   (c) Take the edge $(v, w) \in E(S)$
   (d) Remove (v, w) from $E(S)$, and remove w from $V(S)$
   (e) $w := v$, $i := i - 1$
   (f) Go to step 2
4. Choose the edge $(w, z) \in T$ the terminating vertex z of which has the highest number c of shortcuts (if there are several such edges, then choose one randomly)
5. If $c = 0$, then choose the edge $(w, z) \in T$ the terminating vertex z of which has the highest number of outgoing edges (if there are several such edges, then choose one randomly)
6. Add (w, z) to $E(S)$, and add z to $V(S)$
7. $w := z$, $i := i + 1$
8. If $i < s$, then go to step 2
9. Output the path $(V(S), E(S))$ and stop

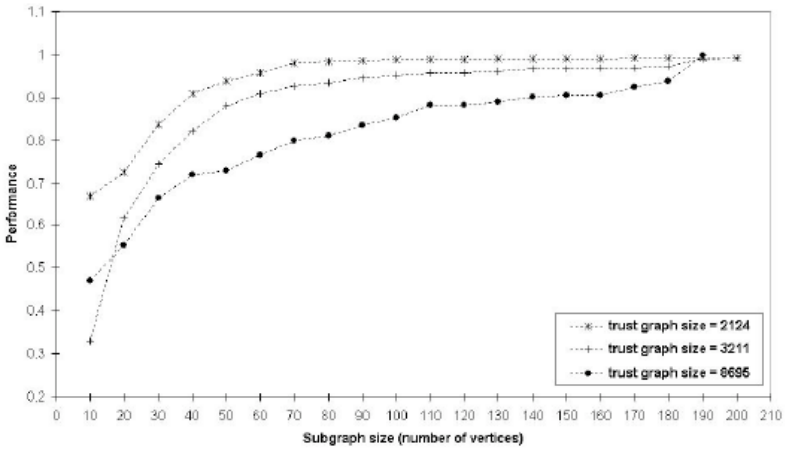
Evaluation of Shortcut Hunter



Star Shortcut Hunter Algorithm

- Modification of Shortcut Hunter algorithm to allow for multiple disjoint out-bound and in-bound paths
- Builds $p_{out} = \min(n_{out}, c)$ out-bound and $p_{in} = \min(n_{in}, c)$ in-bound paths
- n_{out} *and* n_{in} are the number of u's outgoing and incoming edges, respectively
- Length of each path is ceiling of $s/(p_{out} + p_{in})$

Evaluation of Star Shortcut Hunter



What about Dishonest Users?

- Introduction of an authentication metric

$$\mu(u, v, G)$$

$$p_{\mathcal{A},\mu}(G) = \frac{1}{\#W} \sum_{(u,v)\in W} \frac{\mu(u, v, S_{\mathcal{A}}(G, u, v))}{\mu(u, v, G)}$$

$$W = \{(u, v) \in V \times V : \mu(u, v, G) \neq 0\}$$

Conclusion

- Proposed architecture obviates need for a certificate directory
- Innovative method for self organization

Discussion

- As nodes are mobile how often is the path information updated and how much resources will need to be consumed for continuous updates when devices change the neighbors in radio range?
- Performance was evaluated on the largest strongly connected components from PGP databases.
- What controls could be places to prevent a user from lying about number of connections?