Review of "The Quest for Security in Mobile Ad Hoc Networks" by HuBaux, Buttyan and Capkun

Hua Li

04-09-2003

Outline

- Overview of Security Issues on Ad hoc networks
- Proposed mechanism: Self-Organized Key Establishment in Ad hoc networks

Attack Levels

- Attacks on Basic Mechanisms
  - Nodes captured and compromised
  - Eavesdropping
  - MAC cooperative
  - Routing ……
- Attacks on Security Mechanisms
  - On key management and keeping, such as public key maliciously replaced, keys compromised

Protection on Basic Mechanism

- Tamper Resistance
  - Hardware --- SIM card
- Securing Routing
  - Cooperative: Watchdog+Pathrater
  - Redundancy: diversity coding
- Service Enforcement
  - Availability: nuglet

Protection on Security Mechanism

- Key establishment
  - Key transport
    - Securely transfer
  - Key agreement
    - Shared key derived by two parties
- Focus of the Paper: Distributed Asymmetric key Establishment (eliminating centralized certification authority)

Three Families of Distributed Certification Authority

- Threshold Cryptography
  - Share refreshing
- Key Agreement
  - Physically present
- Self-Organized Public-key Scheme
  - Comes from PGP
  - But Distributed

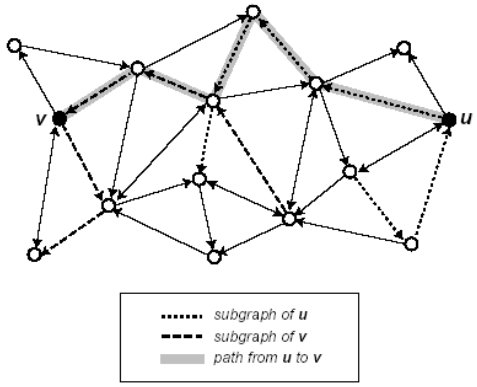
Basic Idea - I

- Public-key certificates issued by users
- Certificates stored and distributed by users
- Each user maintains a local certificates repository selected by author's algorithm

Basic Idea - II

- Each user stores the certificates it issued
- Each user stores a set of selected certificates issued by other users
- Merge the local certificate repositories of two nodes if one wants to obtain the public key of the other

Diagram



Model

- Transfers to Graphs Theory Problem
- Known: Trust (Certificates) Graph
- Purpose: Find the minimum number of edges (subgraph) stored in user u to v so that can find a certificate chain from u to v.
- Requirement: Performance, Scalability, Distribution, and Robustness

Algorithm-Shortcut Hunter

- Shortcut: A shortcut is defined as an edge, upon whose removal, the shortest undirected path between the nodes previously connected by that edge becomes strictly >=2
- Rule: Choose the vertex with the highest number of shortcuts (Please see appendix for details)
- Out-bound path and in-bound path

Performance Evaluation

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

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

Improvement

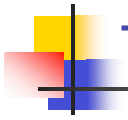
- Original algorithm not good for large size graph
- Propose: Star Shortcut Hunter Algorithm
- Main idea: Reduce Path length, multiple paths

Feathers

- Real Distributed Key Establishment
- Equivalent rule for each nodes
- However
  - How to get trust graph for each node before it can get its subgraph?
  - Partial Information recover total information is based on probability

Future Work

- Hierarchy or clustered based infrastructure might be better than purely distributed and centralized ones
- Key updating in each nodes has to be considered
- Storage skyrocketed--→ why not whole graph?



Thank You!

Appendix

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