Ad Hoc Networks - Routing and Security Issues

Mahalingam Ramkumar
Mississippi State University, MS

January 25, 2005
1 Outline

2 Introduction
   - Some Basic Terms
Basic Terms

- Ad Hoc vs Infrastructured
- AHN
- MANET (Mobile Ad hoc NETwork)
- Wireless Networks
- Peer-to-peer networks
- Multi-hop
- Sensor Networks
- Resource Constraints
- Autonomous
- Self Organizing
- Ubiquitous computing
- Pervasive networks
- Trusted Devices
Pre-Requisites

- CS 4153/6153
Routing protocols
WLAN / Bluetooth
Mobile IP
Cryptography and network security (basics)
Various routing protocols
Review of routing protocols for fixed networks
Security Issues

- Security under resource constraints
An intro to cryptography
Routing - read Chapter 5, Section 2 in Ref 2 (Tanenbaum) - Routing Algorithms
Optimality principle
Shortest path routing
Flooding
Distance vector routing
Link state
Hierarchical routing
Broadcast routing
Multicast routing
Routing for mobile hosts
Routing in AHN
Node look-up in P2P networks
POA - continued

- Ref 8 (tutorial paper on ad hoc routing protocols)
- DSDV
- AODV
- DSR
Ground Rules

- Random selection
- Grading
  - participation
  - attendance might have an indirect effect!
  - individual assignments
  - term-paper / project
Blom’s Key Predistribution

- \( F(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{n} a_{ij} x^i y^j \mod p, a_{ij} \in \mathbb{Z}_p \)
- \( \Phi = \{a_{ij}\} - \text{TA's secrets} \)
- Symmetric polynomial \( a_{ij} = a_{ji} \). Has \( \binom{n+2}{2} \) unique coefficients
- Consider \( n = 1 \), \( F(x, y) = a + b(x + y) + cxy. \)
- \( \Phi_A = F(x, A) = G_A(x) = (a + bA) + x(b + cA) = s_{A_1} + xs_{A_2}. \)
- \( \Phi_B = F(x, B) = G_B(x) = (a + bB) + x(b + cB) = s_{B_1} + xs_{B_2}. \)
- More concrete example, \( F(x, y) = 8 + 7(x + y) + 2xy \mod 17. \)
- \( A = 12, B = 7, C = 1. \)
- \( G_A(x) = F(x, 12) = 7 + 14x. \) Or \( \Phi_A = \{7, 14\}. \)
- \( G_B(x) = F(x, 7) = 6 + 4x. \) Or \( \Phi_B = \{6, 4\}. \)
- \( G_C(x) = F(x, 1) = 15 + 9x. \) Or \( \Phi_C = \{15, 9\}. \)
- \( K_{AB} = G_A(B) = G_B(A) = F(A, B) = F(B, A) = 3. \)
- \( K_{AC} = 4, K_{BC} = 10. \)
If you know the secrets in $C (15,9)$ you still would not be able to determine $K_{AB}$.

The example above is 1-secure

In general, for polynomials of degree $n$ the system is $n$-secure

You have to know secrets in $n + 1$ nodes to beat the system - or discover system secrets $a_{ij} \forall i, j$

Each node needs only $n + 1$ secrets! (Not a function of the total network size!)

What is the maximum network size?

Clue - every node needs a unique ID.

Why prime modulus?
C needs $K_{AB}$

$$K_{AB} = a + b(A + B) + c(AB) \mod p$$

Conspires with $D$

$C$ knows $s_{C_1} = a + bC$ and $s_{C_2} = b + cC$

$D$ knows $s_{D_1} = a + bD$ and $s_{D_2} = b + cD$

Together they know $s_{C_1}, s_{C_2}, s_{D_1}, s_{D_2}$

$$\begin{pmatrix}
1 & C & 0 \\
0 & 1 & C \\
1 & D & 0
\end{pmatrix} \begin{pmatrix}
a \\
b \\
c
\end{pmatrix} = \begin{pmatrix}
s_{C_1} \\
s_{C_2} \\
s_{D_1}
\end{pmatrix}$$

(1)

Just need to solve for $a, b, c$

Could have also used $s_{D_2}$ instead of $s_{D_1}$
But Without $D$...

$$K_{AB} = a + b(A + B) + c(AB) \mod p$$

$C$ knows $s_{C_1} = a + bC$ and $s_{C_2} = b + cC$

\[
\begin{pmatrix}
1 & C & 0 \\
0 & 1 & C \\
1 & A + B & AB
\end{pmatrix}
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
= 
\begin{pmatrix}
s_{C_1} \\
s_{C_2} \\
K_{AB}
\end{pmatrix}
\]

(2)

Any value of $K_{AB}$ would satisfy the set of equations!

Or all $K_{AB}$s are equally likely

$C$ does not get any information about $K_{AB}$

1-secure!
Generalized KPD

- $F(x, y)$
- $G_A(x) = F(x, A)$
- $G_B(x) = F(x, B)$
- $G_B(A) = G_A(B) = F(A, B) = F(B, A) = K_{AB}$.
- $n$-secure KPD
- Number of TA’s secrets $P$
- Number of secrets for each node $k$
- For efficient KPD schemes $k \propto n$
Random KPD (Random Preloaded Subsets)

- Each node assigned \( k \) keys randomly from a set of \( P \) keys
- \( K_{AB} \) is based on the secrets common to nodes \( A \) and \( B \)
- Remember birthday paradox?
- Even for small \( \frac{k}{P} \) chances of finding an intersection is surprisingly high!
- If \( k \approx \sqrt{P} \) probability of an intersection is about 0.5
- However, probability that a specific key can be found is lower by a factor \( \frac{1}{\sqrt{P}} \)
- So any two nodes can find a secret with high probability. The chance that the key can also be found by some other node is very small.
- Can be made arbitrarily secure by making \( P \) and \( k \) high enough.
- The ratio of \( P \) and \( k \) is very crucial!
So we have a TA with $P$ secrets $K_1 \cdots K_P$.

We have two nodes $A$ and $B$ that need to discover $K_{AB}$.

We have $n$ colluders $O_1 \cdots O_n$.

All of them have $k$ secrets ($A$ has secrets $\Phi_A$)

Let us assume $A$ and $B$ share $m$ secrets $K_{s_1} \cdots K_{s_m}$

$K_{AB} = h(K_{s_1} || K_{s_1} || \cdots || K_{s_m})$

How “secure” is $K_{AB}$?

When is $K_{AB}$ not secure?

If $\{\Phi_A \cap \Phi_B\} \subset \{\Phi_{O_1} \cup \Phi_{O_2} \cdots \cup \Phi_{O_n}\}$

Attackers have to know every shared secret to be successful.
Analysis

- Let $\frac{k}{P} = \xi$
- $\xi$ is the probability that a key is assigned to a node (any key to any node).
- Take a particular key (say key $i$)
- What is the probability that the $i$th key is secure?
- $A$ and $B$ should have the key
- $O_1 \cdots O_n$ should not have the key.
- $\epsilon = \xi^2(1 - \xi)^n$
- A game - with $P$ rounds
- Every round may yield a secret that is secure - but with a very very low probability!
- Attacker wins a round $i$ if key $i$ is not safe
- However - attacker has to win every round!
- $p_e = (1 - \epsilon)^P$
- $0.95^{1000} = 5.3 \times 10^{-23}$, $0.999^{50000} = 1.9 \times 10^{-22}$
Broadcast Authentication

- Message $M$
- Source has $k$ keys $K_1 \cdots K_k$
- $MAC_i = HMAC(M, K_i)$
- Broadcast $M \mathbin{\|} MAC_1 \mathbin{\|} MAC_2 \cdots \mathbin{\|} MAC_k$
- How can attackers fool some verifier into accepting a fake message as authentic?
- How can the attacker fool all verifiers that fake message is authentic?
Broadcast Encryption

- We have a universe of $N$ nodes
- Need to distribute a secret to $g$ nodes
- $r$ nodes excluded; $r + g = N$
- How?
  - If $g << N$
  - If $r << N$
- Node revocation with broadcast encryption
Multicast Security

- Broadcast authentication
- Broadcast Encryption
- Instantaneous conference communications (group secrets)
Topics

- Routing / forwarding, session routing
- Desirable properties
  - Correctness
  - Simplicity
  - Robustness
  - Stability
  - Fairness
  - Optimality
Topics

- Adaptive and nonadaptive algorithms
- Optimality Principle
- Shortest path routing
- Flooding
Assignment 1

Implement Dijkstra’s algorithm for 20 nodes
Suggested interface
dalg numnodes source ¡distances.dat¿
Should print out shortest path to all nodes from source
If no input file - generate distances randomly
(Generate random x,y coordinates
If distance between two nodes is more
than a threshold, set distance to \( \infty \))