G 364: Mobile and Wireless Networking

CLASS 10, Mon. Feb. 9 2004

Stefano Basagni

Spring 2004

M-W, 11:40am-1:20pm, 109 Rob

Clustering for Ad Hoc Networks

- Giving hierarchical structure to networks
- Decrease the amount of information at each node
- Enhances scalability
- Helps in “resource assignment”

Some Applications

- Stateless multipoint communication: Routing, multicast and broadcast over the backbone
- Resource/user discovery
- Implementation of security in large networks of sensors
- Network management

Pebblenets

Ad hoc nets with a twist:

- Very small communication devices with:
  - Embedded processing capabilities
  - Storage
  - Communication
- Large number
- Battery powered

Why Yet Another Name?

- Pebblenets can be deployed
  - For enabling peer-to-peer communication among mobile network users
  - As sensor networks, for sensing activities and gathering and transport of data
- Each device is like a “pebble”

Security in Pebblenets

Security requirements:

- Confidentiality: all communication are intelligible by authorized principals only
- Integrity: all communications are modifiable by authorized principals only
- Authenticity: all communication are generated by authorized principals only
- Availability: the service is available only to authorized principals when needed

Pebblenets: The System, 1

- Pebbles are born equal
- Pebbles are small things: symmetric key cryptography is the only feasible solution
- Each pebble is equipped with a secret *group identity key (K_{GI}),* and one-way funtion h
- Authentication is based upon group membership and not individual identities
- Data is protected with a global *Traffic Encryption Key (TEK)*

Pebblenets: The System, 2

- Each pebble has local unique identifier ID
- They can dynamically compute a *weight* that accounts for:
  - Pebble parameters (mobility, battery power, etc.)
  - Surrounding environment
  - Etc.
- Pebbles are tamper-resistant (capturing honest pebbles does not allow to insert malicious ones)

Secure Pebblenets: The Problem

- Secure pebblenets should guarantee data traffic confidentiality and authenticity
- Data traffic is encrypted by all pebbles by using the same *Traffic Encryption Key (TEK)*
- The TEK changes during network lifetime
- **Need for a key management protocol for securely updating and distributing the TEK to the pebbles**

The Re-keying Protocol

- Executed periodically at each pebble
- Selection of key managers (that produce the new TEK)
- Security is increased by selecting each time a different key manager
- The selection is based on the dynamically changing weight associated to each pebble (unpredictable selection of the fittest pebble for the role)

Re-keying Protocol: Two Phases

- I: Distributed and secure selection of a small fraction of the fittest pebbles and their organization into a "small backbone"
- II: Distributed and secure, weight-based selection of the pebble(s) that generate the new TEK

Re-keying Protocol: Phase I

- Secure neighbor discovery: p => N(p)

  $E(K_i,H)(w(p)|ID(p)|MAC(K_{GI}))$

- Secure "elector" selection: p => N(p)

  $E(K_i,H)(w(p)|ID(p)|Role|MAC(K_{GI}))$

- Electors covers some non-elector pebbles: e => N(e)

  $E(K_i,H)(ID(e)|K_e|MAC(K_{GI}))$

Re-keying Protocols: Phase I

- Secure backbone construction
  - The described electors selection protocols leads to a topology such that:
    - Pebbles are either electors or not (tertium non datur)
    - No two electors are neighbors
    - All non-electors are covered by one elector
  - A backbone of electors is obtained by "joining" electors that are at most three hops away
    - The backbone is connected iff the pebblenet is
    - There is a backbone key to secure this operation

Re-keying Protocol: Phase II

- The *Key Manager* should be the fittest elector, i.e., the elector with the biggest weight
- Ad hoc leader election is expensive (backbone wide)
- We define a unique, localized way for each pebble to decide whether it can be a *Potential Key Manager (PKM)*

Re-keying Protocol: Phase II

- A pebble is a PKM if it has the bigger weight among its neighboring electors
- A PKM generates and broadcasts over the backbone a new *TEK* (and a new backbone key) after an exponentially computed time
- *TEK*s are sent with ID of generating pebble
- Other backbone pebbles compare ID and make decision to keep or discard *TEK*

Re-keying Protocol: Phase II

- "Collisions" are handled by individual backbone pebbles
- Result is global selection of new *TEK* based on ID
- New *TEK* is securely distributed by backbone pebble to their covered pebbles
- All pebbles use the new *TEK* for data encryption

Secure Pebblenets, What's there and what's missing

- Key management protocol to secure communication in networks of small devices
- Fast and simple, adaptive to changing network condition
- Next: Extensive simulation results
- Next: The exponential thing vs. leader election?
- Next: Splits and joins and other amenities

Assignments

- Download the paper from webpage
- Updated information on the class web page:

  www.ece.neu.edu/courses/eceg364/2004sp