Part I: A Clean Slate Proposal for a Secure Wireless Network

Part II: Vignettes of the Algorithms and Protocols

Yih-Chun Hu and P.R. Kumar

Dept. of Electrical and Computer Engineering, and Coordinated Science Lab
University of Illinois, Urbana-Champaign
Goals

- Clean slate design to secure wireless networking
- Provably secure design
- Min-Max Optimal
- Complete suite of algorithms/protocols
- Run applications based on temporal coordination

- Principled approach to security
  - Holistic security to security
  - Security is not an afterthought
- Security first, performance second
  - Performance subsequently optimized while preserving security
- Reverse of the usual approach
A lot of explanation is clearly needed …
My talk will explain
- Motivation
- Intuition
- Approach
- Results
- Architecture
- Outline of protocols

Yih-Chun Hu’s talk will provide vignettes of the protocol suite

Work in progress – This is the first airing of these results

So your comments and suggestions are highly appreciated
Basic objective

- A complete suite of algorithms/protocols that takes you from startup:
  - With just a set of nodes
  - Some good
  - Some bad
  - Good nodes don’t know who the bad nodes are

- To an optimized functional network carrying data reliably
What can go wrong with a network formed in presence of bad nodes?

- Some nodes are bad. What can go wrong?
- Lots of things. A bad node could
  - Refrain from relaying a packet
  - Advertise a wrong hop count
  - Advertise a wrong logical topology
  - Jam
  - Cause packet collisions
  - Behave uncooperatively vis-à-vis medium access
  - Disrupt attempts at cooperative scheduling
  - Drop an “ACK”
  - Refuse to acknowledge a neighbor’s handshake
  - Behave inconsistently

Byzantine behavior
One approach

- Identify “ATTACKS”
- Provide “DEFENSES”
- Result is
  - A sequence of patches
  - Arms race

- Issue
  - What other attacks are possible?

- Can we come up with clean slate provably secure architecture?
- Principled design: Holistic approach to Security, not afterthought
- Complete suite of protocols from start-up to reliable operation?
Main results

- Scheme that leads from start-up to functional policed system
- Resulting network is Min-Max optimal with respect to utility

\[ \text{Min} \quad \text{Max} \quad U(x) \]
\[
\text{All behaviors of bad nodes} \quad \text{Protocols}
\]

- In fact we will show

\[ \text{Min} \quad \text{Max} \quad U(x) \]
\[
\text{Bad nodes can choose to either Jam or Cooperate} \quad \text{Protocols}
\]

- Hence, bad nodes are restricted to following actions
  - Either Jam or Cooperate in a consistent way

- Also, timings applications can be consistently run on network
Implications of results

- Bad nodes can either choose to Jam or Cooperate
  - In a way that is consistent for each concurrent transmission set

- Other more “Byzantine” behaviors are ruled out
  - Not relaying a packet
  - Dropping an ACK
  - Presenting a wrong logical topological view
  - Disrupt medium access cooperation
  - Disrupt timing applications by inconsistent behavior
  - Not cooperating, disrupting, lying, spreading rumors, etc

- Bad nodes can only do two things – Jam or Cooperate

- Moreover, nobody can prevent Jamming or Cooperating when it is done in a consistent way
Why would a bad node ever cooperate?

- $U(x) = \text{Min}(x_i)$

- If C jams, it can reduce $x_{AB}$

  $$\lim_{|BC| \to \infty} x_{AB} = x_{AB}^{\text{Max}}$$

- If C pretends to be good, it can insist on equal share

  $$\lim_{|BC| \to \infty} x_{AB} = 0$$
Limitations and extensions under study

- Approach is not information-theory based
- It is packet based
- In particular, probabilistic unreliable channel is abstracted as a reliable channel of lesser rate
  - “Rate Adaptation”
- Issues of attacking this very abstraction are not addressed today
- Extensions under study
Fundamental ingredients of our approach

- Standard cryptographic primitives are assumed
  - All packets are encrypted
  - Bad nodes cannot create fake packets, cannot alter good packets without getting caught, etc

- And, importantly,

- Clocks and synchronization
Why clocks and synchronization?

- Without a notion of time, we cannot even talk of throughput
  - Without throughput we cannot talk of network Utility
- So time is an essential ingredient

- Without a notion of common time, nodes cannot cooperate temporally
  - They cannot share resources in a time-based way
  - Cooperative scheduling, etc., will be impossible
- So synchronization will be a fundamental ingredient
  - Facilitates temporal cooperation
Additional capabilities enabled by clocks and synchronization

- Good nodes can catch bad nodes that misbehave temporally
  - For example, by delaying some packets for a longer time
  - Thus timing-based applications can be consistently run

- Good nodes can catch some Man-in-the-middle (MITM) attacks
  - Can catch all half-duplex MITMs

- Good nodes can make bad nodes that cooperate behave in a logically and temporally consistent fashion
  - So timing-based applications can be run over the network
Technical Assumptions (1)

- Bounded domain
- $n$ nodes, some bad
- Minimum distance between any pair of nodes
- Nodes are not mobile

- Max power constraint at each node
- Noise at each node

- Path loss is a function of distance
- SINR based rate
- Half-duplex nodes (can relax somewhat)
Technical Assumptions (2)

- Affine clock at each node
  - \( 0 < 1 - \varepsilon \leq \text{Skew} \leq 1 + \delta \) for all nodes

- Packets take a delay \( d_{ij} \) from node \( i \) to node \( j \)

- Orthogonal MAC code to avoid primary conflicts

- Each node has a private key

- Each node has a certificate which binds a public key to its identity and that is signed by a trusted authority
Technical Assumptions (3)

◆ Assumption on connectedness
  – Suppose all nodes transmit at Max power
  – Then suppose there is an edge between each pair of nodes \((i, j)\) an for which \(\text{SINR}_{ij}\) and \(\text{SINR}_{ji}\) both exceed \(\text{SINR}_{\text{threshold}}\)

– Assumptions
  » Resulting graph is connected
  » Subgraph of good nodes is also connected
Phases of protocol

- **Neighbor Discovery Protocol**
  - Use orthogonal codes
  - Within a bounded time all nodes discover their neighbors
  - Also pass on their public keys to their neighbors

- **Clock Synchronization Phase**
  - Pairs of neighboring nodes synchronize clocks
    - Skews can be determined
    - Offsets and one-way delays *cannot* be identified
    - Round trip delays are determined
    - They obtain capability to predict when packet reception times
    - They also identify and certify each end-point and certify state of link
Phases of protocol (2)

- Route discovery phase
  - Nodes flood the link-states throughput the network

- Consistency check phase
  - Nodes check that for all cycles
    \[ \prod_{(i,j) \in \text{Cycle}} Skew_{ij} = 1 \]

- At this point
  - All MITMs that are not conforming to consistent timing are caught
Phases of protocol (3)

- Attack on all half-duplex MITMs
  - Every pair of neighboring nodes sends a long packet that exceeds the round-trip delay

- Repeat as needed: Nodes have view of the network
  - Which sets of nodes can concurrently transmit
  - Link-state including clock-synchronization parameters

- Choice of operating point for Network Utility Maximization
  - Optimal network resource scheduling is chosen
  - And agreed to by all nodes
Phases of protocol (4)

- **Data transfer phase**
  - The nodes send their data
    » Over the agreed paths
    » According to the agreed schedule
    » Relaying taking place according to the schedule

- **Verification of operation**
  - Route prefix verification is done to ensure that nodes are conforming
  - Can identify concurrent transmission sets that are not reliable
  - Detected non-conforming concurrent transmission sets are eliminated and network view is established all over again
Experimental evaluation upto consistency phase

- Consistency checks

![Image of a grid with numbers 1 to 25 and a physical setup with objects arranged in a pattern]
Thank you