

# Evolutionary design of attack strategies

Jiří Kůr, Vashek Matyáš, Petr Švenda

Faculty of Informatics, MU Brno

*Presented at the 17<sup>th</sup> International Workshop on Security Protocols,  
Cambridge, 2009*

# Why design an attack strategy?

- To secure systems
- Smart people learn from their mistakes, while wise people learn from the mistakes of others
- No mistakes allowed while designing secure system
- Defender needs to “think” as an attacker
- Defender who knows nothing about possible attacks almost always fails

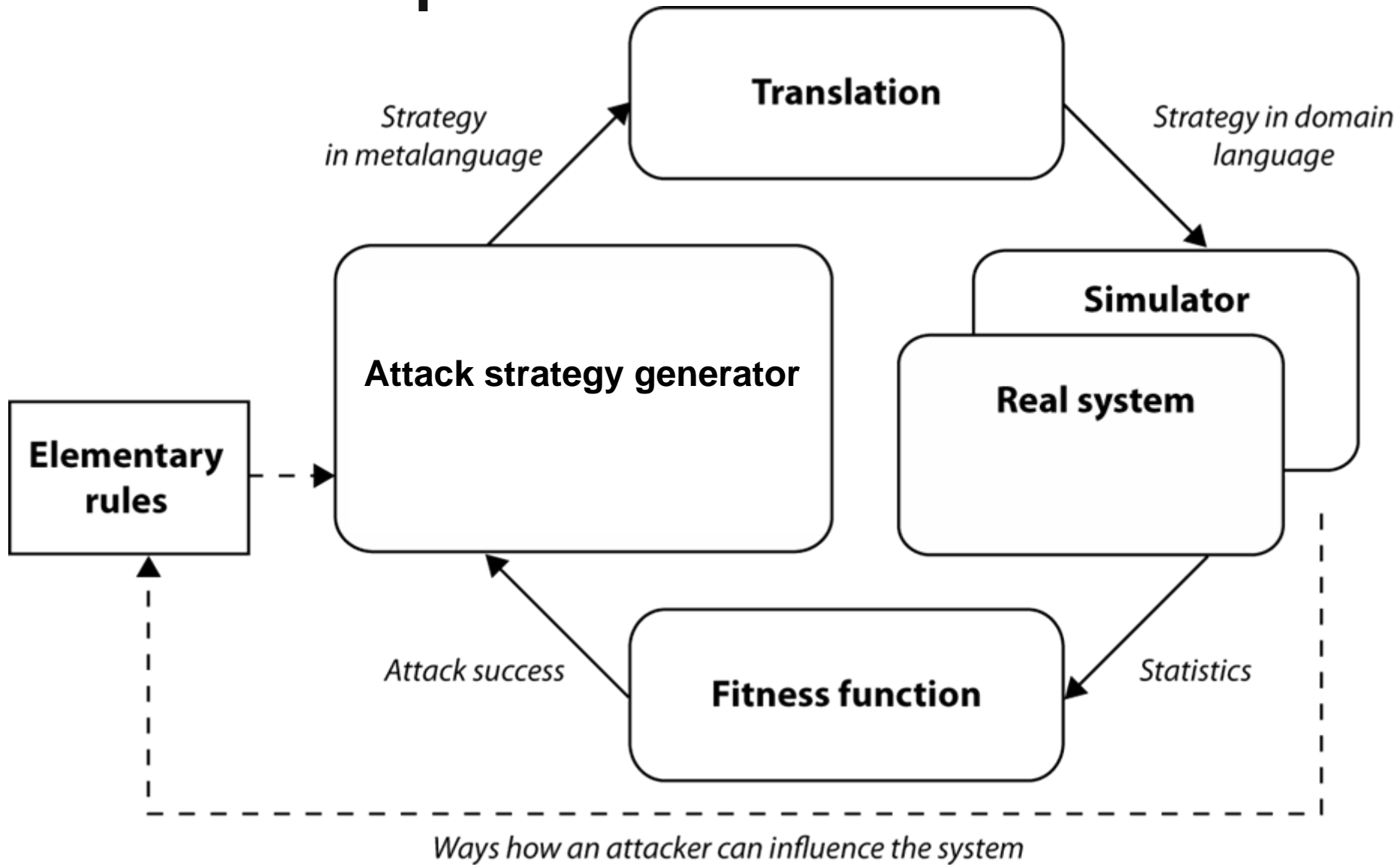
# Attacker vs. defender

- Fundamental asymmetry between the attacker and the defender
  - attacker needs to find only one attack path
  - defender should secure all of them
- Exhaustive search over the space of possible attack paths
  - suitable approach for the defender
- Informed search for possible attacks without inspecting all possibilities
  - suitable for an attacker

# Automatic attack generation

- Idea: use a guided search to automatically generate attack strategies
- Basic approaches
  - Optimization of known attack strategy
    - search for optimal parameters
  - Novel attacks from elementary rules

# Basic concept

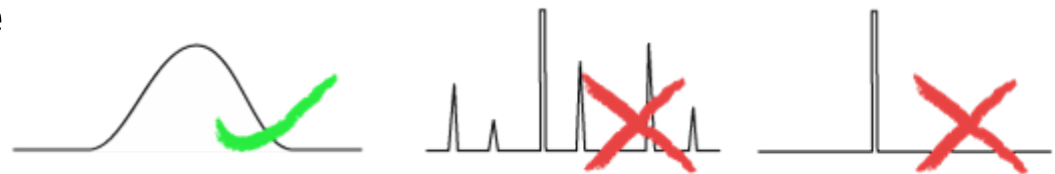


# Attack strategy generator

- Educated guess
  - sometimes too complicated for humans
  - cannot be automated
- Exhaustive search
  - sometimes too large search space
- Random search
  - low probability of success in case of too large search space
- Guided search
  - evolutionary algorithms, hill climbing

# Evolutionary algorithms

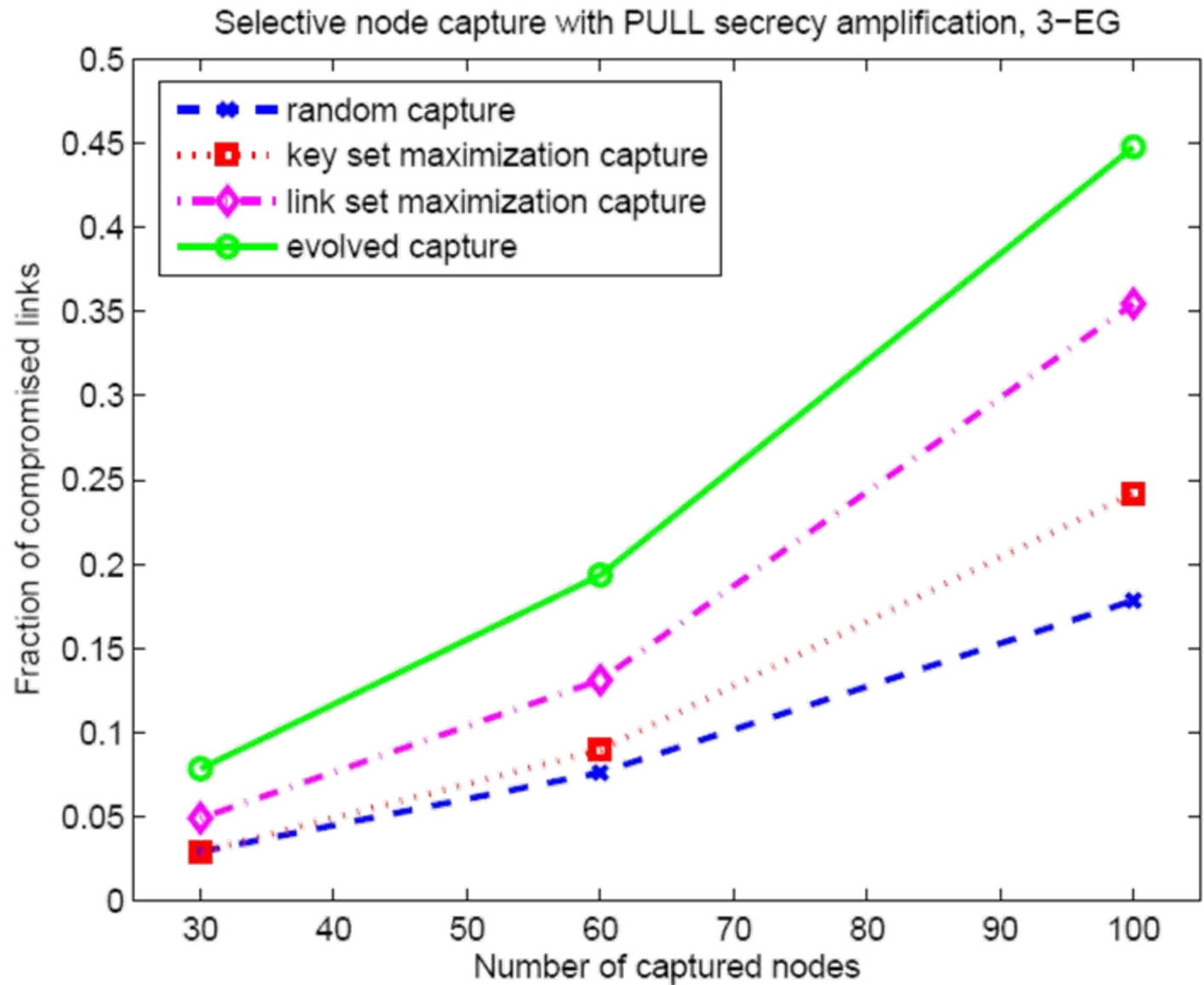
- Inspired by Darwin's evolution theory
- Multiple attack strategies generated in each round
- New attack strategies generated from the most successful attack strategies of a previous round
- Algorithmic function to evaluate attacker's success – fitness function
  - capture progress towards optimum
  - sufficient granularity
  - fast to compute



# Attack 1: Selective node capture

- Attacks in the area of wireless sensor networks
- Probabilistic pre-distribution with overlapping key sets performed in the network
- Attacker goes for maximum advantage with fixed number of captured nodes
  - maximum number of compromised links
  - with information about actual topology and key distribution
- Example attack settings:
  - probabilistic pre-distribution (3 keys at minimum)
  - secrecy amplification protocol run atop
- Compared for several deterministic algorithms

# Selective node capture - results



# Attack 2: Malicious routing

- Attacker controls several misbehaving nodes
  - search for attacks against insecure routing
  - fitness functions: non-delivered messages, message hops, messages routed over malicious node, ...
  - elementary rules: store/load value, send message, time counters
  - triggers of response code on specific action
- Usually hard to analyze the results
  - complex behavior and interleaving of elementary actions

# Malicious routing – results

- Implicit geographic forwarding (IGF)
  - next hop selected during ORTS/CTS handshake
  - selection based on geographic positions of the nodes and base station, remaining energy and random element
- Found attacks for IGF
  - rushing attack – ignoring waiting time period
    - malicious node always selected as next hop
  - selective MAC layer collisions
    - to maximize number of hops / undelivered messages
  - overloading of neighbours' message buffers
    - resulting in message drop
- Shift to optimization
  - selected routing algorithms are insecure, several simple attacks exist
  - shift to the optimization problem when these simple attack found
  - possibility for over learning on specific topology

# Conclusions

- It is beneficial to search for novel attack strategies
- Automated approaches are welcome due to diversity of usage scenarios
  - can be tuned to specific environment, fast to adopt
  - more complex relations in solved problem, better the results w.r.t. algorithmic solution
- Approach works well for optimization of known attacks
- Potential to search for new attacks
  - progress must be controlled (powerful but simple to detect attacks)
  - it is hard to provide a exhaustive list of (implicit) assumptions
- We were not able to find appropriate fitness function for all attacks

**Thank you for your attention.**

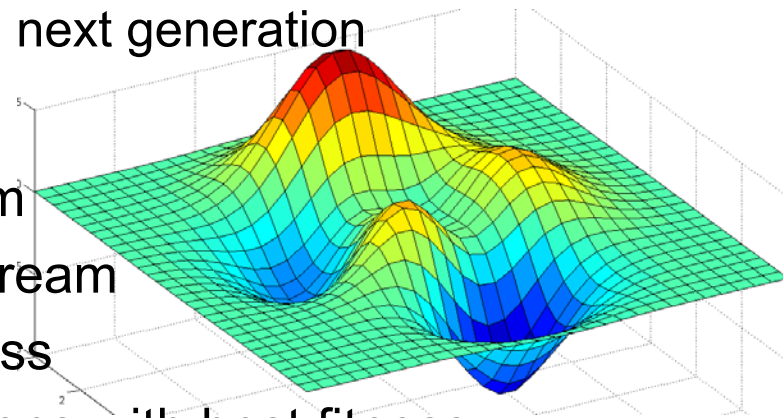


# Elementary rules

- Each node contain memory slots for temporal storage of messages, node identities
- Instructions and triggers
- INS\_SEND\_M p1 – send message from memory slot p1
- INS\_DROP\_M p1 – drop message from memory slot p1
- INS\_FAKE\_N p1 p2 p3 – forge item of type p2 in message from memory slot p1, new value is stored in memory slot p3
- TRIG\_DATA – data packet was overheard
- TRIG\_RNG p1 – start action randomly with probability p1

# Method solving inspired by evolution

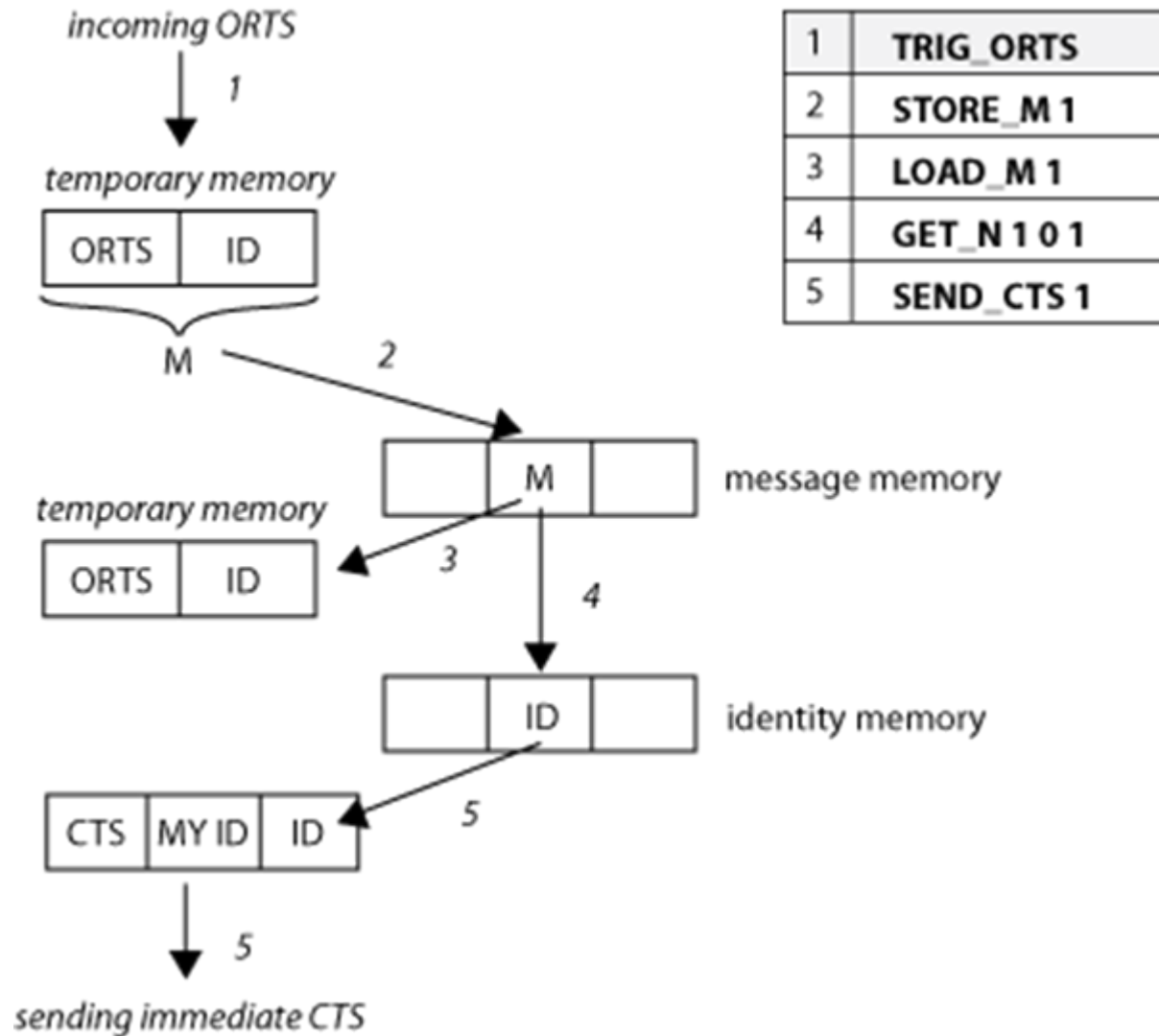
- Charles Darwin - *On the Origin of Species* (1859)
- Necessary prerequisite for evolution to work:
  - elementary units – genes
  - possibility to reproduction - copy itself with reasonable quality
  - possibility to mutation - new information can be introduced
  - natural selection – “better” specimen has more offspring
    - and its genes will be more often in next generation
- Evolutionary algorithms
  - “clever” search for function maximum
  - candidate solution encoded as bit stream
  - algorithmic function to evaluate fitness
  - next generation selected from solutions with best fitness



# Malicious routing – results

- Minimum cost forwarding (MCF)
  - minimal spanning tree build by beacons from base station
- Initial attacks found
  - fitness: route most messages to attacker
  - forging beacons, impersonation of base station
  - broadcast as low cost field as possible (or replay received beacon)
  - extremely simple yet powerful, no proceed towards other attacks
    - reached a local maximum of fitness landscape
- Some constructions had to be banned
  - to allow search for more complex attacks
  - but same goal achieved by different techniques
    - drop message – instruction for drop, storing and overwriting, ...
- Hard to really ban the simple attacks
- Might fool the signature-based intrusion detection

# Rushing attack



# Fitness function

- Key to successful usage of evolution
- Represents an attackers goal
- Properties:
  - capture the progress towards the optimum
  - sufficient granularity
    - potential for evolution to gradually increase the quality of the solution
  - fast to compute
    - evaluation of  $10^2$  to  $10^6$  or more candidates in reasonable time

# Categories of generated attacks

- Re-combination of the existing attacks
  - put existing attacks together in meaningful order
  - e.g., capture packet, forge IP, replay packet
- Improvement (optimization) of known attack strategy
  - principle is known, “tuning” of parameters
  - e.g., which subset of nodes should be captured
- Finding novel attack strategies
  - attacks composed from very simple actions
  - e.g., set/store byte X of message, transmit Y millise., ...

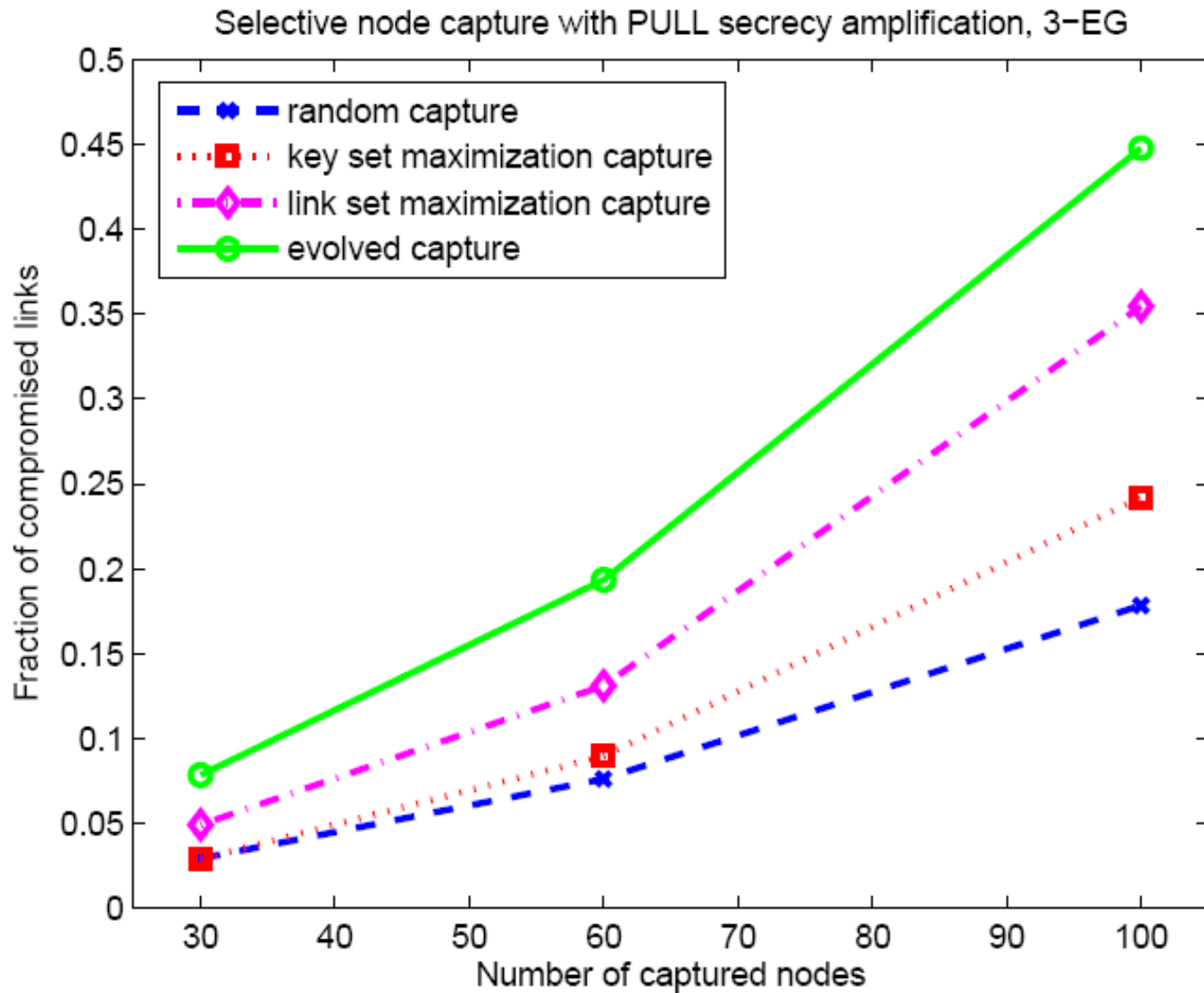
# Applications

- In the domain of Wireless sensor networks
- Improvement (optimization) of known attack strategy
  - optimal eavesdropping pattern
  - selective node capture
- Finding novel attack strategies
  - attacks against routing protocols

# Attack 1: Selective node capture

- Attacker likes to obtain maximum advantage with fixed number of captured nodes
  - compromised links, carried keys, impact on data aggregation, ...
  - has information about actual deployment
- Genom encodes ID of nodes to capture
- Attacker success is evaluated by network simulator
- Example attack settings:
  - probabilistic pre-distribution
  - PULL secrecy amplification
  - 3 keys at minimum

# Selective node capture - results



# Attack 2: Malicious routing

- Misbehaving attacker nodes
  - search for attacks against standard routing
  - we focused on networks with relatively fixed communication pattern
  - elementary actions store/load value, send message, time counters
  - triggers binded on specific action (type of message in air)
  - fitness as fraction of non-delivered messages, message hops, messages routed over malicious node
- Minimum cost forwarding (MCF)
  - minimum spanning tree based with base station as a root,
  - periodic broadcast of beacons, BS has cost 0
  - cost based on distance and remaining energy of node
- Implicit geographic forwarding (IGF)
  - next hop selected during ORTS/CTS handshake
  - selection based on geographic positions of the nodes and base station, remaining energy and random element

# Malicious routing - results

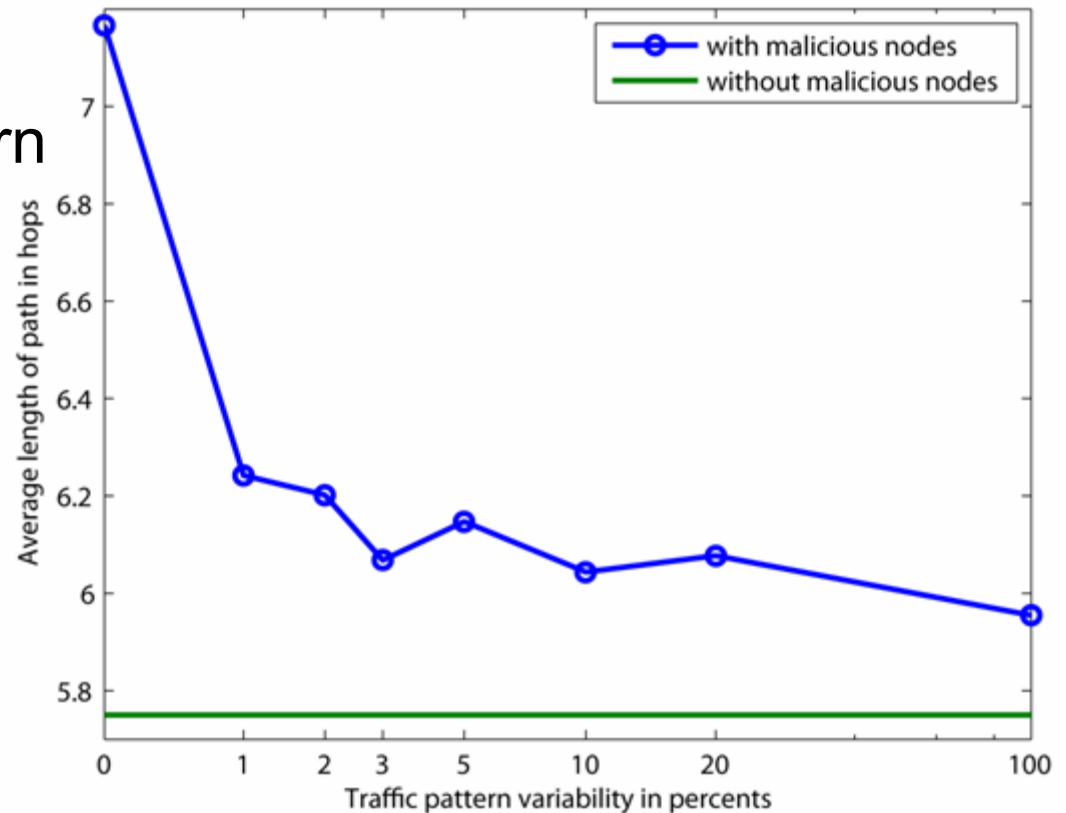
- Usually hard to analyze
  - complex behavior and interleaving of elementary actions
  - pruning - actions without impact on fitness are discarded
  - still, we were unable to fully interpret all details
- MCF
  - impersonation of BS, forging beacons
    - simple yet powerful attack, banned for following search
  - selective message forwarding/dropping
- IGF
  - rushing attack - immediate answer to Open Request To Send
    - malicious node is always selected as a next hop
  - selective MAC layer collisions
    - to maximize number of hops / undelivered messages
  - overloading of neighbours message buffers – message drop

# Malicious routing - example

- Goal: extend the average length of the routing path
- Attacker dropped packets travelling long distances 😊
- Static traffic pattern

vs.

Variable traffic pattern



# Conclusions

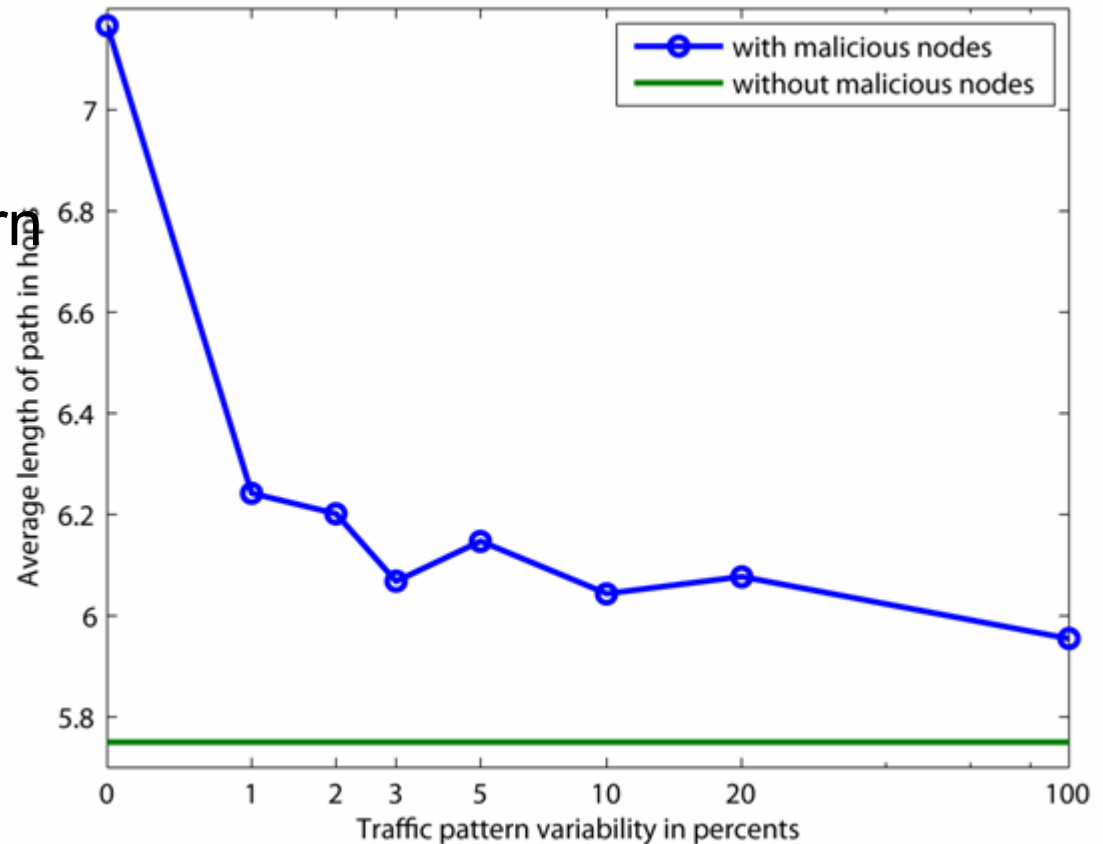
- Potential also for finding recombinations of attacks or novel attacks
  - only limited application area
  - extremely hard to analyze the results

**Thank you for your attention.**

- Attack strategies used abnormal amount of communication
  - battery exhaustion, malicious nodes easy to detect
  - we introduced credits to limit radio activity
  - strategies with more credit had contra intuitively worse results
    - larger search space

# Malicious routing - example

- Goal: extend the average length of the routing path
- Attacker dropped packets travelling long distances
- Problem with
- Static traffic pattern  
vs.  
Variable traffic pattern



# Malicious routing – experience

- Genome structure
  - subgenoms
  - trigger + sequence of instructions
  - specific message type overheard, time trigger, random trigger, ...
- Conditional counters – extreme complexity of generated strategies, no significant improvement
- Memory slots – high number of memory slots slows down the evolution
- 500 nodes, 2 base stations
  - limited size of network and number of send messages
  - fast evaluation of a single attack strategy
- 500 000 – 1 000 000 individuals generated and tested during evolution