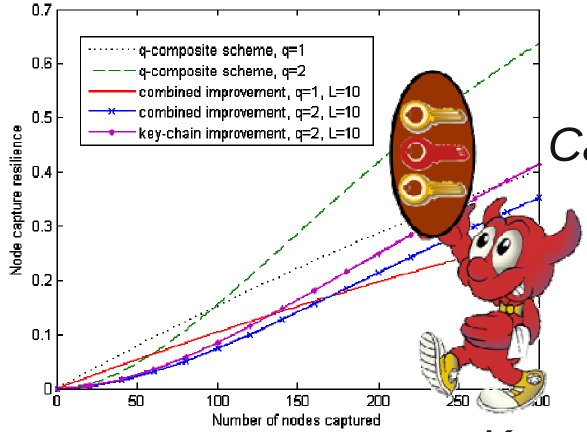
Brno University Security Laboratory

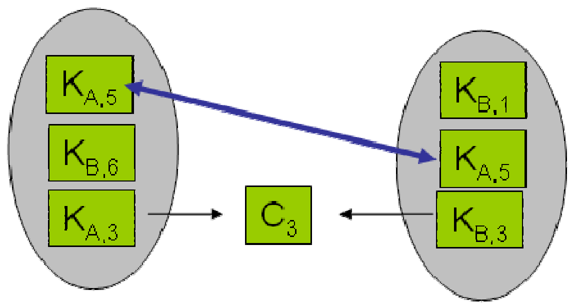
Two Improvements of Random Key Predistribution for Wireless Sensor Networks

Jiří Kůr, Vashek Matyáš, Petr Švenda
Faculty of Informatics
Masaryk University

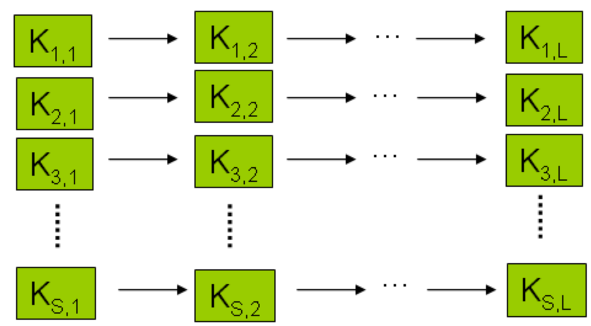


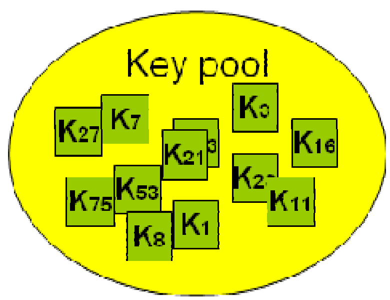
Capture resilience improvements

Collision key improvement

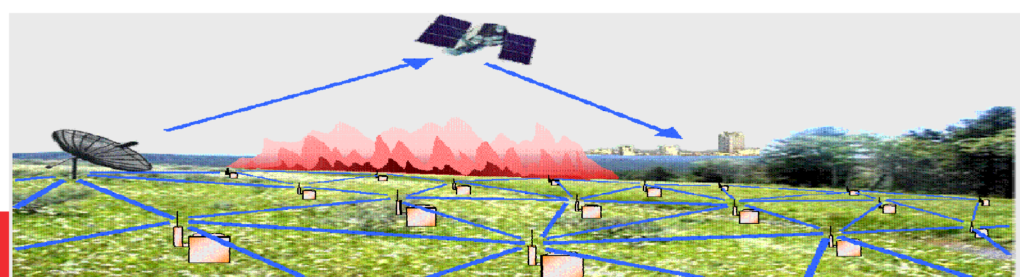


Key-chain improvement





Eschenauer & Gligor 2002
Chen et al. 2003

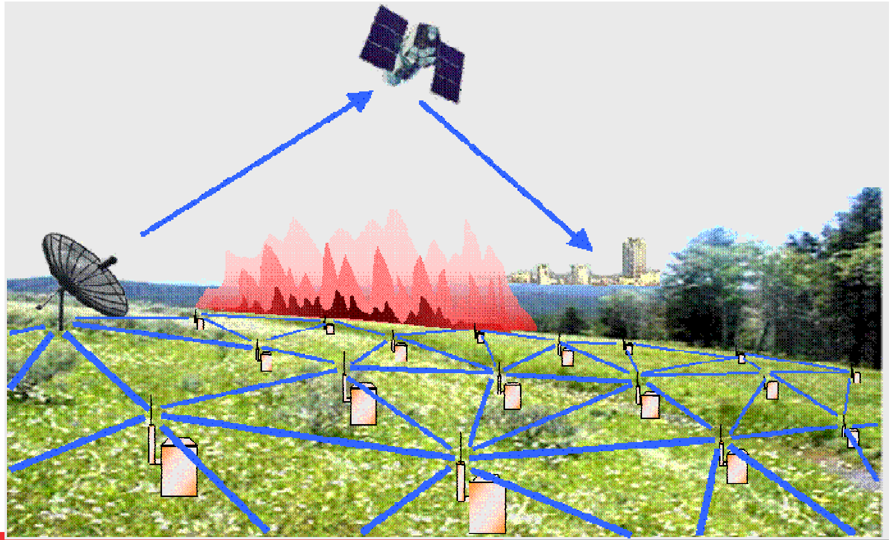


Wireless Sensor Network (WSN)

- Sensor nodes
  - environmental sensors
  - RF transceiver
  - battery powered
  - low computational and memory resources
  - 8-bit processor, 4KB RAM, < 128KB EEPROM
  - number of nodes: 10 – 10^5

- Topology
  - self-organized topology
  - ad-hoc position/neighbors – not known in advance
  - multi-hop communication

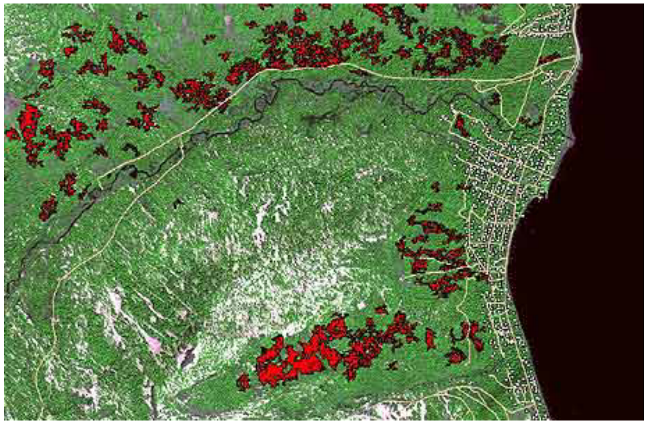
- Base station(s)
  - lap-top capabilities
  - almost unlimited energy resources



Applications of WSNs

Traffic control

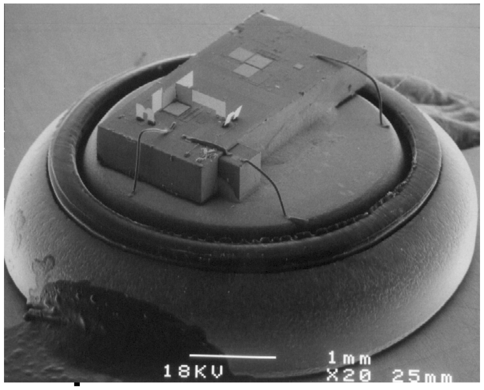
Medical information



Remote fire detection

Combat field control

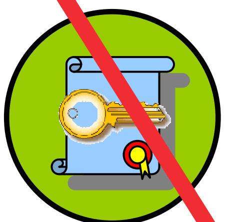
Some differences from standard networks



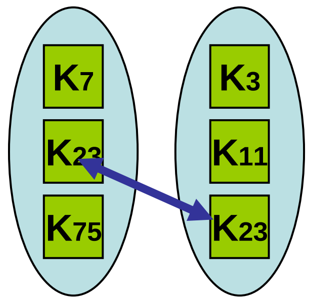
- Running on battery (limited resource)
  - days for personal network
  - years for large scale monitoring network
  - especially communication is energy-expensive
- Relatively limited computation power
  - powerful CPU possible, but energy demanding
- Nodes can be captured by an attacker
  - all secrets can be extracted from unprotected nodes
  - and returned back as malicious node

Many ways how to establish keys

~~Asymmetric cryptography~~

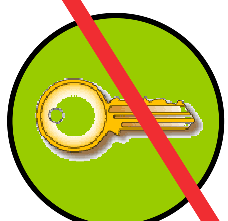


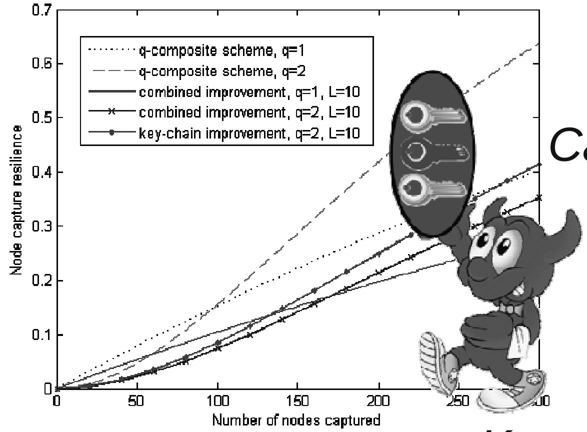
Random key pre-distribution



~~Trusted party~~

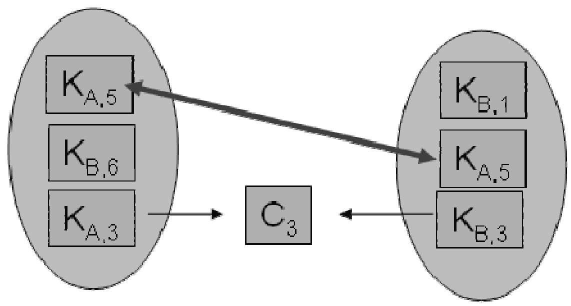
~~Master key, pairwise keys~~



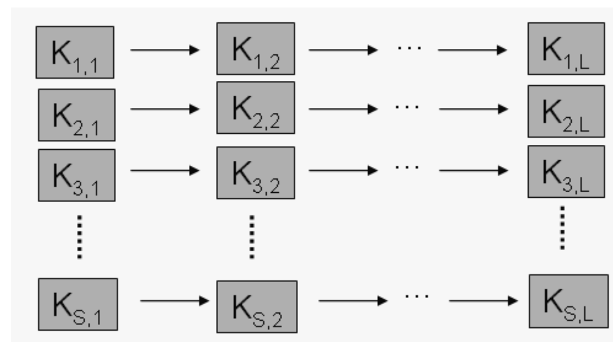


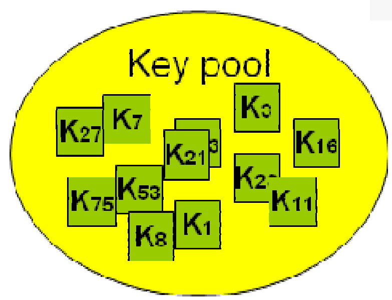
Capture resilience improvements

Collision key improvement

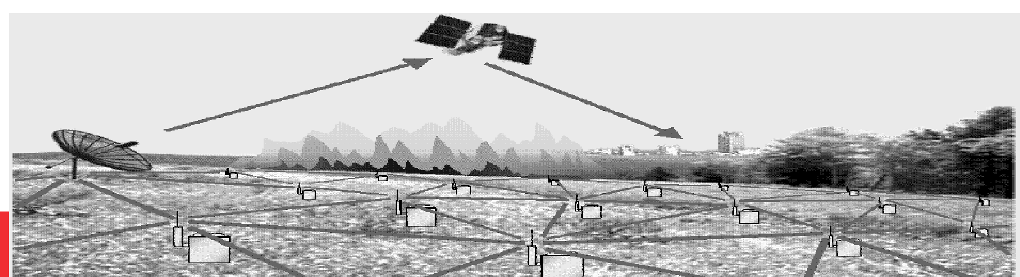


Key-chain improvement



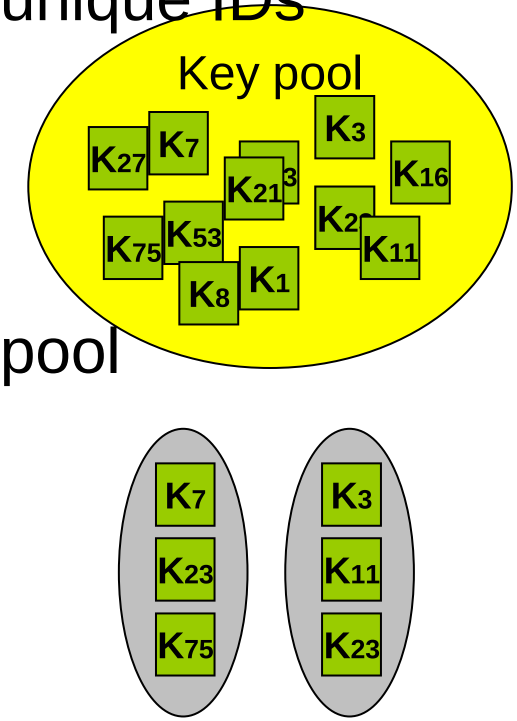


Eschenauer & Gligor 2002
Chen et al. 2003

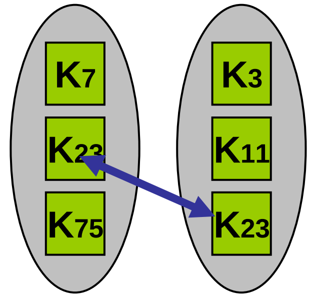


Random key pre-distribution

- *Eschenauer & Gligor 2002, Chen et al. 2003*
- Elegant idea with low memory requirements
  - based on birthday paradox
  - large pool of **S** cryptographic keys with unique IDs used
- For every node prior deployment:
  1. randomly select **m** keys from large key pool
  2. return selected keys back to pool
  3. proceed with next node



Random key pre-distribution (2)



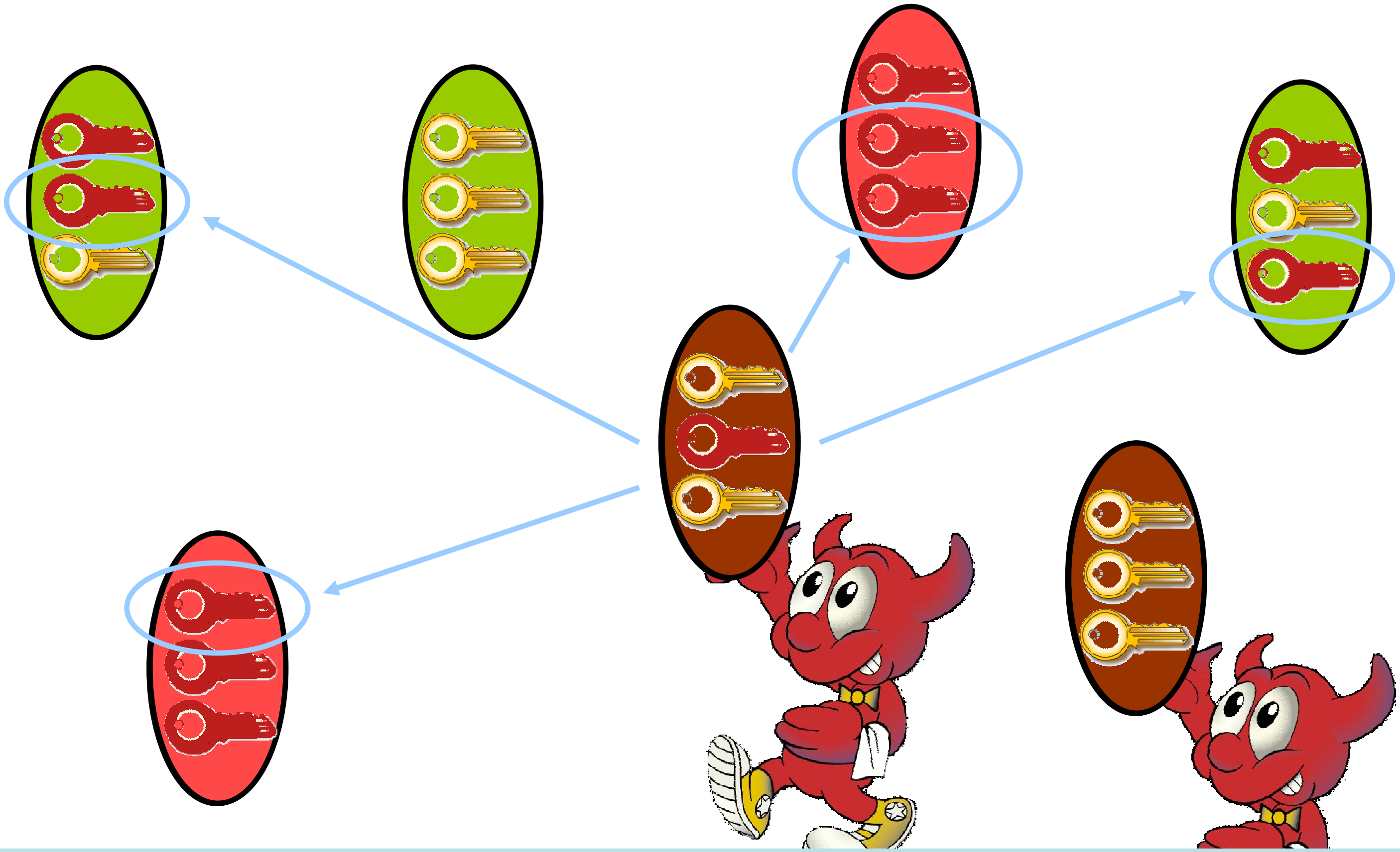
- During neighbour discovery:
  1. neighbours establish radio communication
  2. nodes iterate over their keyrings for shared key(s)
  3. if shared (by chance) key(s) are found, secure link is established

- What is key sharing probability?
  - e.g., 100 keys from 10000
  - 64% probability at least one key shared

- *q*-composite scheme – at least *q* keys shared

- Not all nodes can establish secure link
  - but sufficient connectivity probability can be set

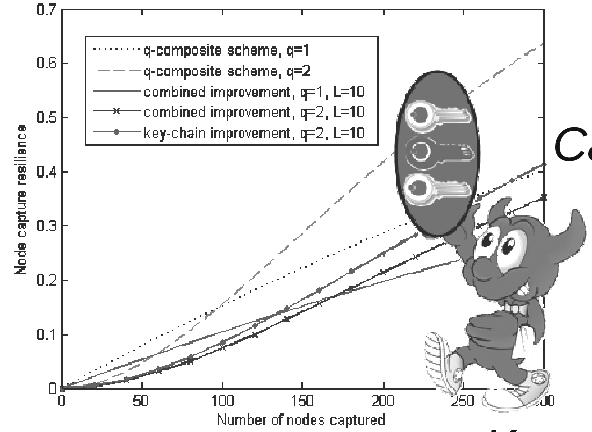
How random key pre-distribution fails



- Keys from uncaptured nodes compromised as well
- Good tradeoff between memory and security

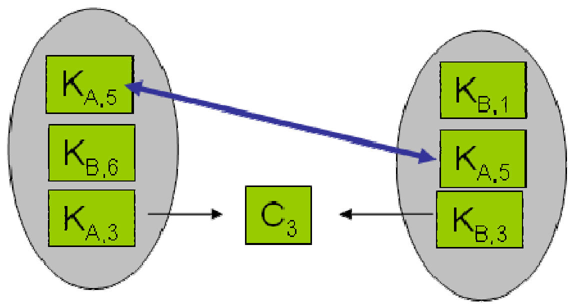
Random key pre-distribution - parameters

- S … key pool size
- m … key ring size
  - node memory limitation
- P … probability that two nodes share at least q keys
  - dependent on m , key pool size S and q
  - we can calculate minimal P required so the network graph remains connected
- *ncr* … node capture resilience
  - assume attacker randomly captured n nodes
  - fraction of secured links between uncaptured nodes that are compromised using keys from captured nodes

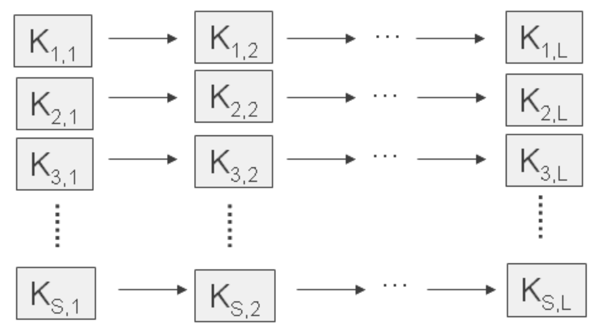


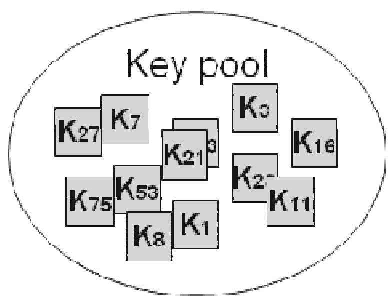
Capture resilience improvements

Collision key improvement

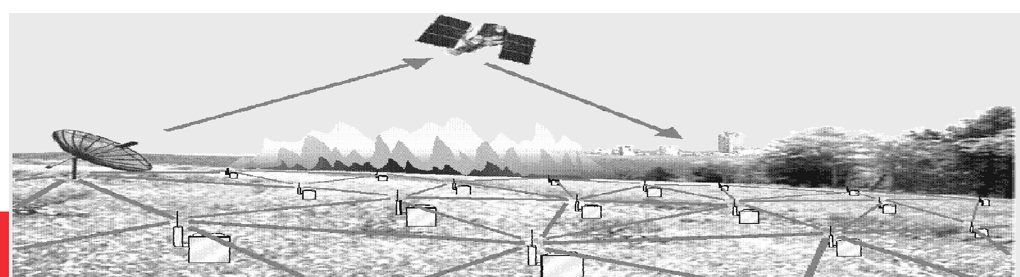


Key-chain improvement



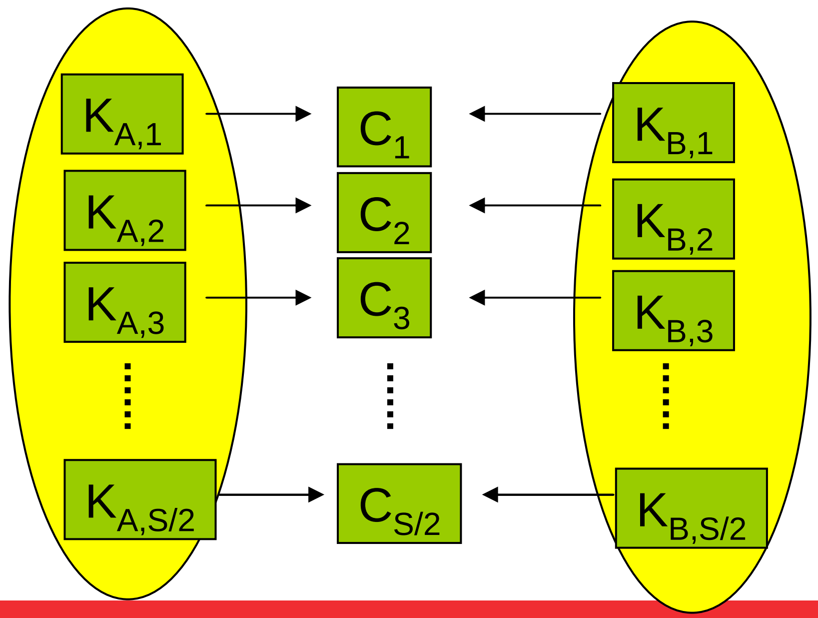


Eschenauer & Gligor 2002
Chen et al. 2003



Collision key improvement

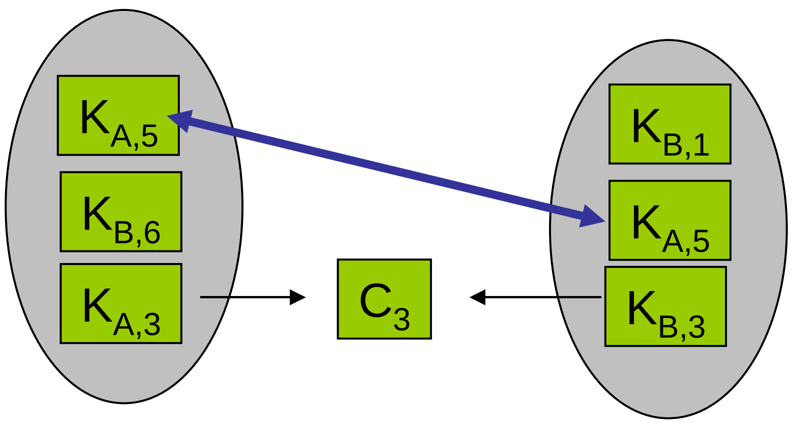
- Key pool created using ***S/2*** related key pairs K_A, K_B
- **$C = H(K_A) = H(K_B)$**
- H is cryptographically secure hash function with a limited input/output length, e.g. 80 bits
- Such collisions can be found with moderate computational power



Collision key improvement (2)

- For every node prior deployment:
  1. randomly select ***m*** keys (***no related key pair*** is allowed)
  2. return selected keys to a key pool
  3. proceed with next node
- During neighbor discovery:
  - beside normal keys, also collision keys can be shared
  - probability of link key establishment is higher

$C_3 = H(K_{A,3}) = H(K_{B,3})$

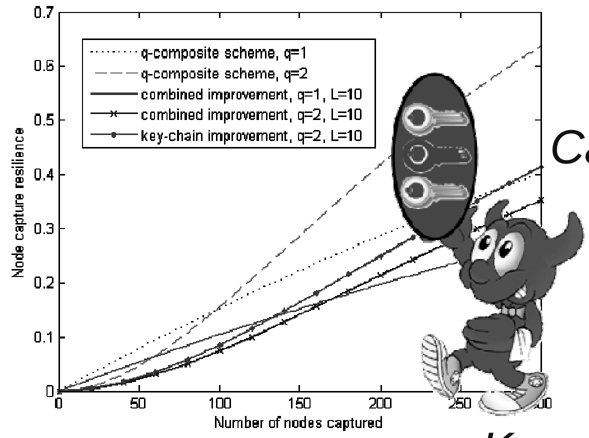


Key pool construction

- To find an n-bit collision approx. $2^{n/2}$ hash operations are needed
- To find c^2 collisions, approx. $c * 2^{n/2}$ hash operations are needed

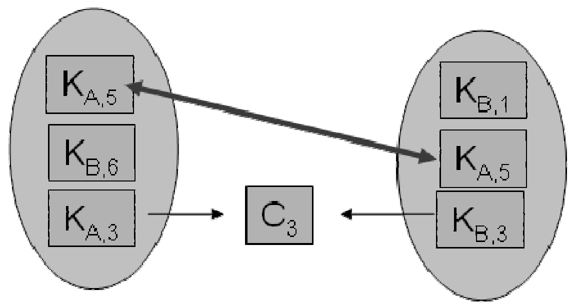
- Goal: to find 80-bit collisions in SHA-2
- Method: Van Oorschot and Wiener's parallel collision search
  - time-memory trade-off approach
- Hash operations computed: approx. 2^{47}
- Over 2^{12} collisions found - enough for key pool

- Aggregate time spent on single 3GHz core: 19 000 hours
- We have used BOINC framework and approx. 1000 cores
  - Final time: approx. 19 hours
  - GPUs could bring significant speed up

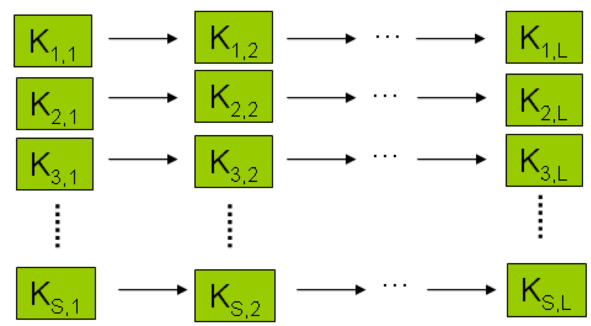


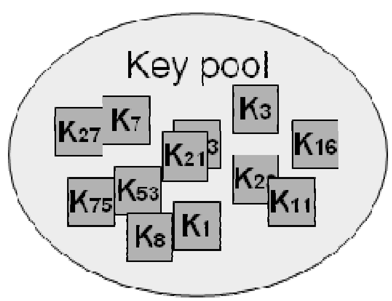
Capture resilience improvements

Collision key improvement

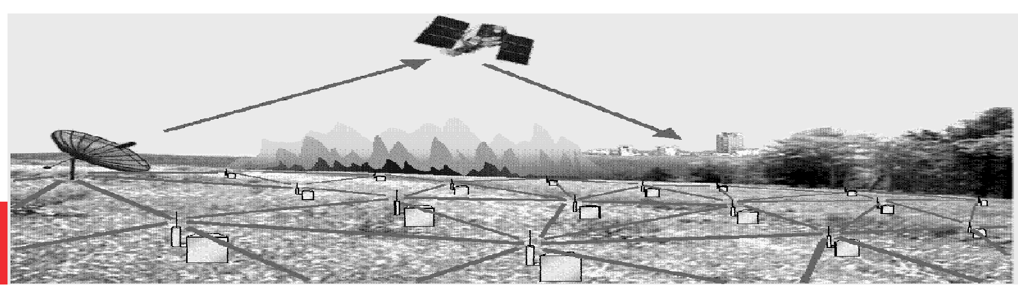


Key-chain improvement



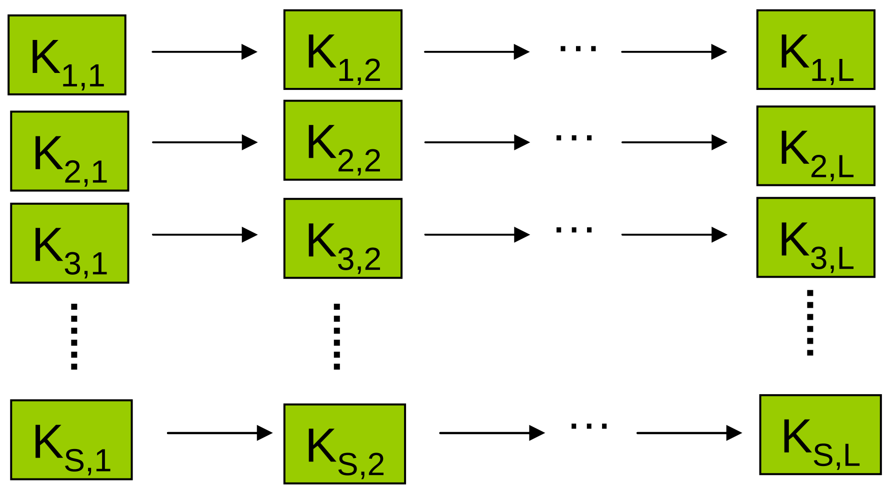


Eschenauer & Gligor 2002
Chen et al. 2003



Key-chain improvement

- Key pool created using ***S*** hash chains of a length ***L***



- For every node prior deployment:
  1. randomly select **m** hash chains
  2. randomly select single key from every selected chain
  3. return selected chains (keys) back to pool
  4. proceed with next node

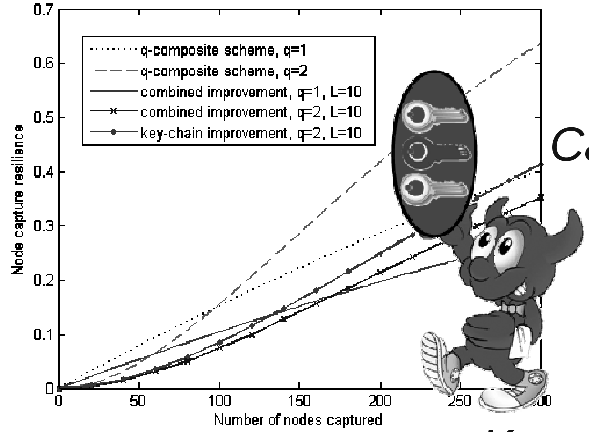
Key-chain improvement (2)

- During neighbor discovery:
  - two nodes can calculate shared key if they posses keys from the same hash chain (with index *i*)

$$K_{i,1} \rightarrow K_{i,2} \rightarrow K_{i,3} \rightarrow K_{i,4} \rightarrow K_{i,5} \rightarrow K_{i,6}$$

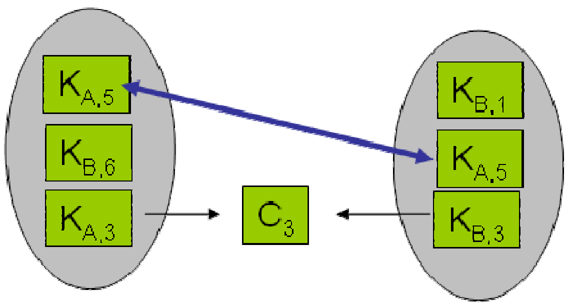
- Probability of key establishment remains as in original design
- Node capture resilience improves
  - attacker may capture keys that are further in the chain
  - slightly better than in collision key improvement

- Hash chains for key predistribution used also in *Ren et al. 2006*
  - different key ring construction, keyed hash function used

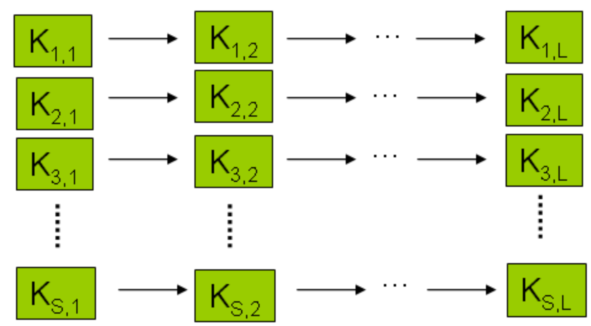


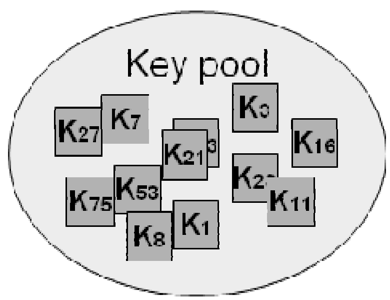
Capture resilience improvements

Collision key improvement

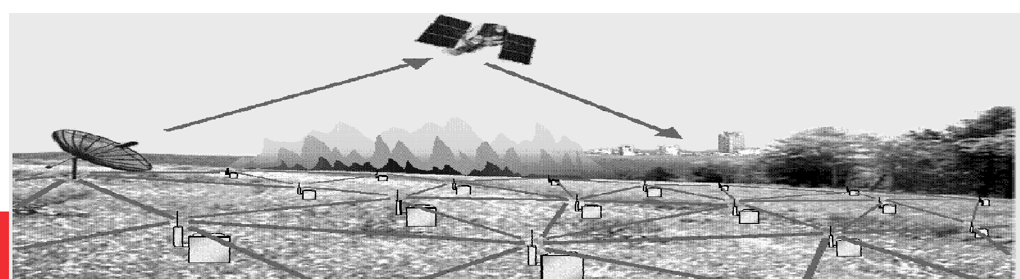


Key-chain improvement



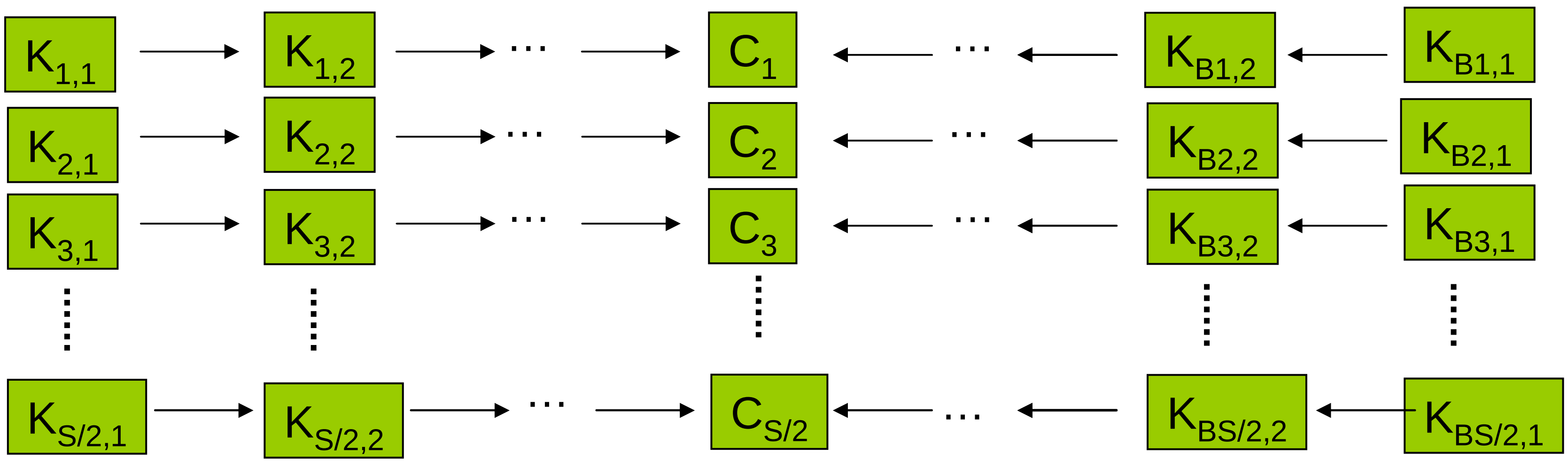


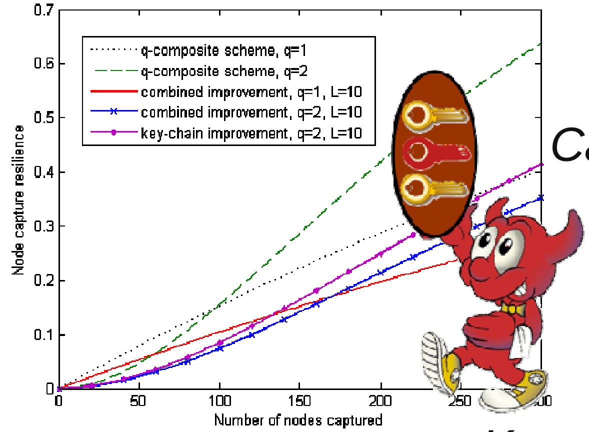
Eschenauer & Gligor 2002
Chen et al. 2003



Combination of improvements

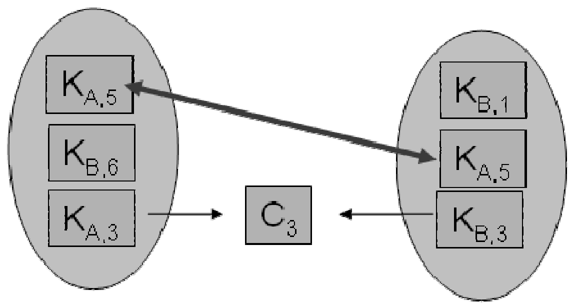
- Both improvements can be easily combined
- Collision search produces colliding hash chains



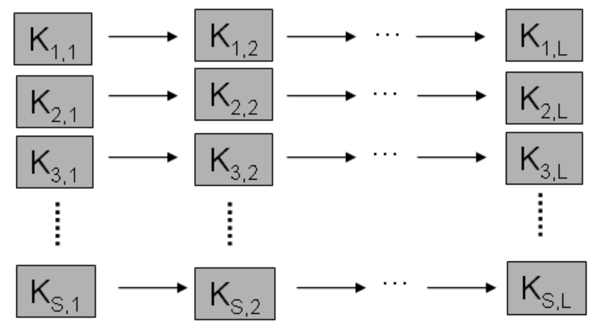


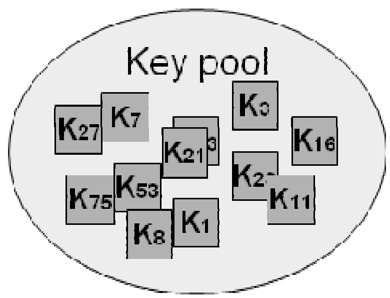
Capture resilience improvements

Collision key improvement

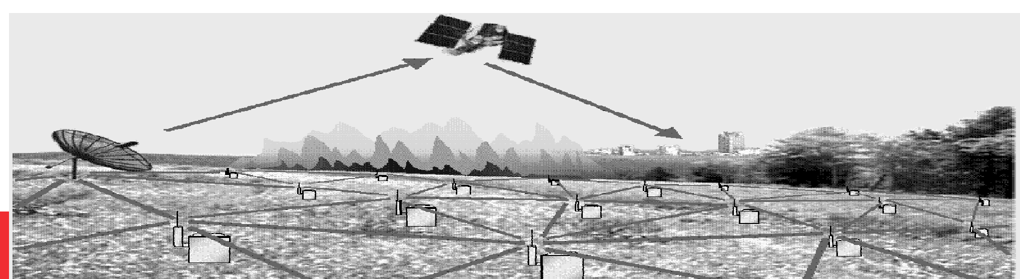


Key-chain improvement





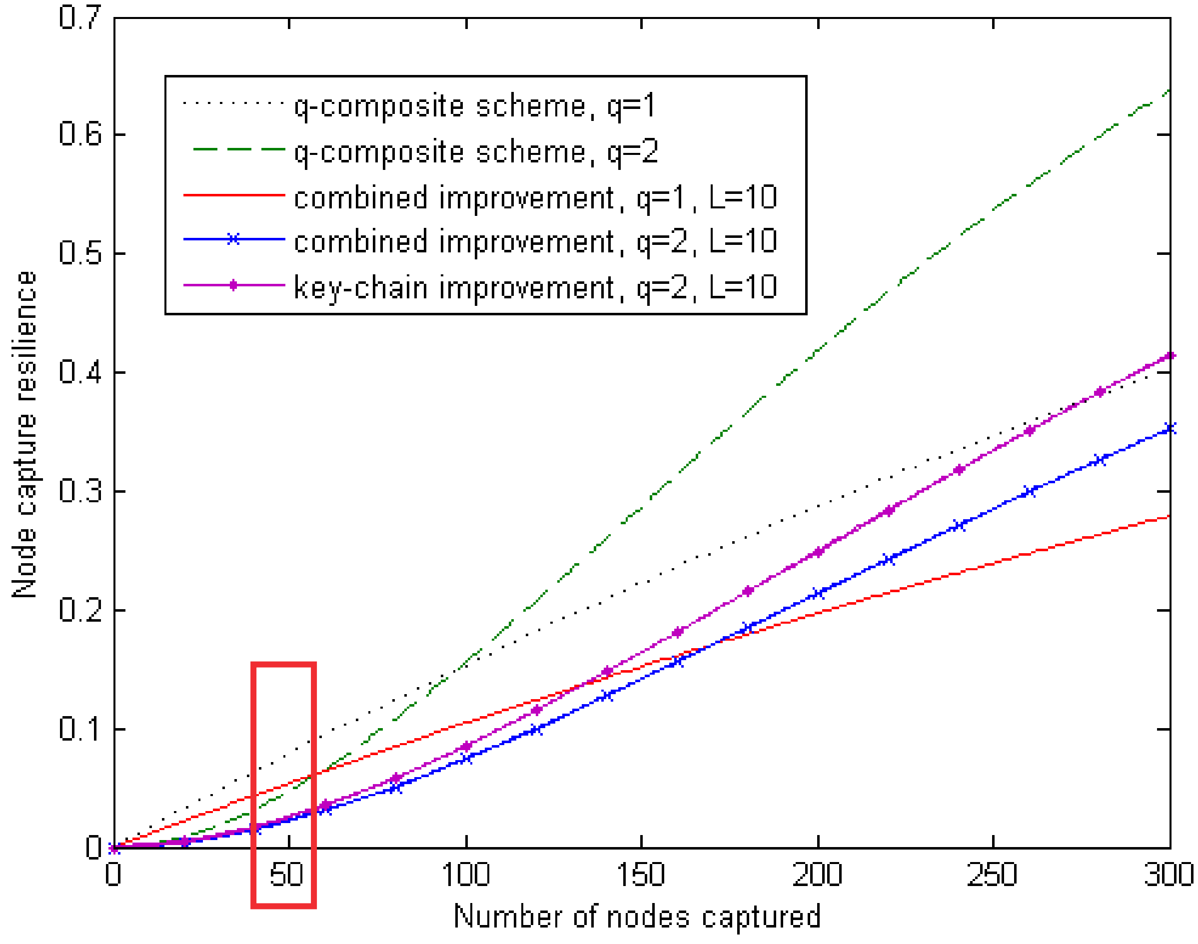
Eschenauer & Gligor 2002
Chen et al. 2003



Combination of improvements - evaluation

- *P = 0.33, m = 200*

- *For q = 2 and n = 50*
- *q-composite: ncr = 4.7%*
- *Collision key: ncr = 2.7%*
- *Key-chain: ncr = 2.5%*
- *Combined: ncr = 2.2%*

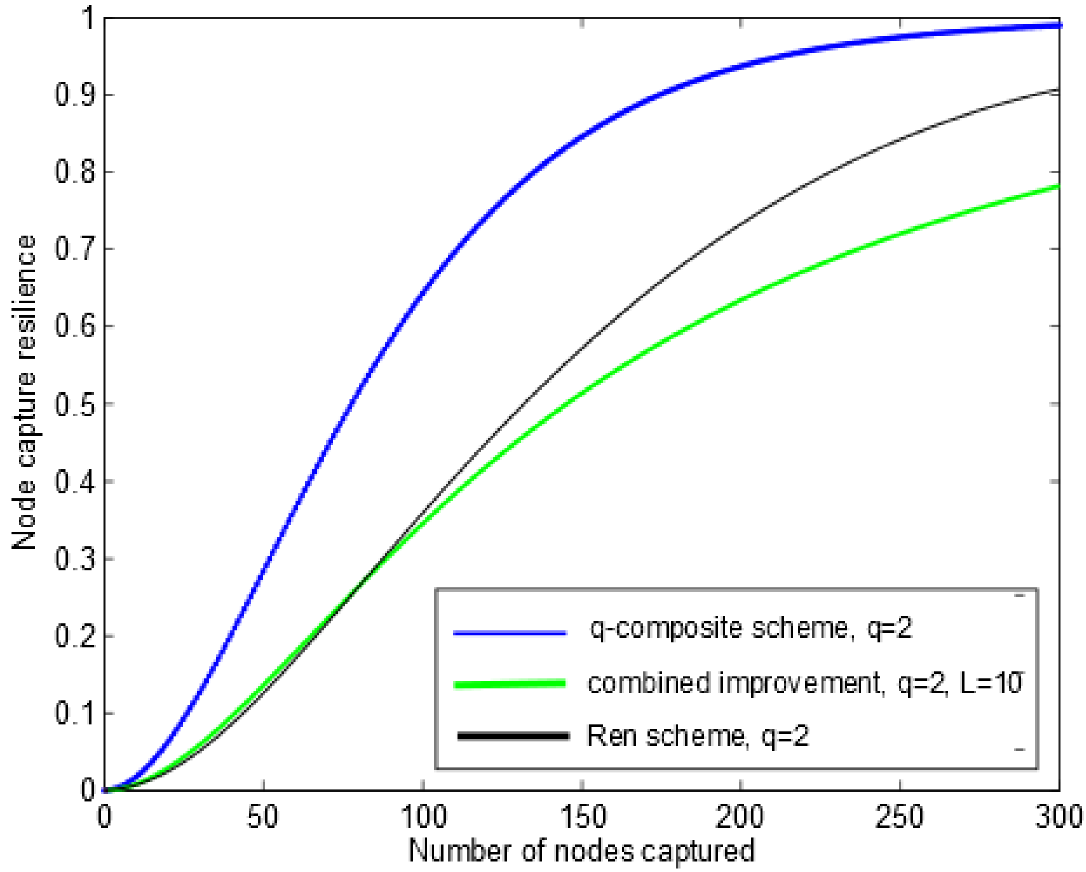


Comparison with Ren et al.

- *Ren et al. 2006*, random key predistribution based on keyed hash chains

- *P = 0.5, m = 90*
- Ren scheme setting
  - R_0 = 10, R_1=79,
  - L=1 000, K=100 000

- Combined improvement outperforms Ren scheme if number of nodes captured is high



Summary

- Eschenauer & Gligor 2002 is one of core schemes
  - many existing schemes extends or builds on it
- Two improvements of this core scheme proposed
  - security performance of extensions also influenced
- Hash collisions can be used in favor of security
  - limited length collisions with moderate CPU resources
- Unkeyed hash chain instead of single key used
- Both improvements combinable
- Results verified both analytically and with network simulator

Thank you for your attention.

Any questions?

References

[Eschenauer & Gligor 2002] Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: 9th ACM conference on Computer and Communications Security, CCS'02, pp. 41-47. ACM, New York (2002)

[Chan et al. 2003] Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: Symposium on Security and Privacy, 2003, pp. 197-213. IEEE, (2003)

[van Oorschot & Wiener 1999] van Oorschot, P.C., Wiener, M.J.: Parallel collision search with cryptanalytic applications. Journal of Cryptology, 12(1):1-28, (1999)

[Ren et al. 2006] Ren, K., Zeng, K., Lou, W.: A new approach for random key pre-distribution in large-scale wireless sensor networks. Wireless Communications and Mobile Computing, 6(3):307-318, (2006)

Key-chain length

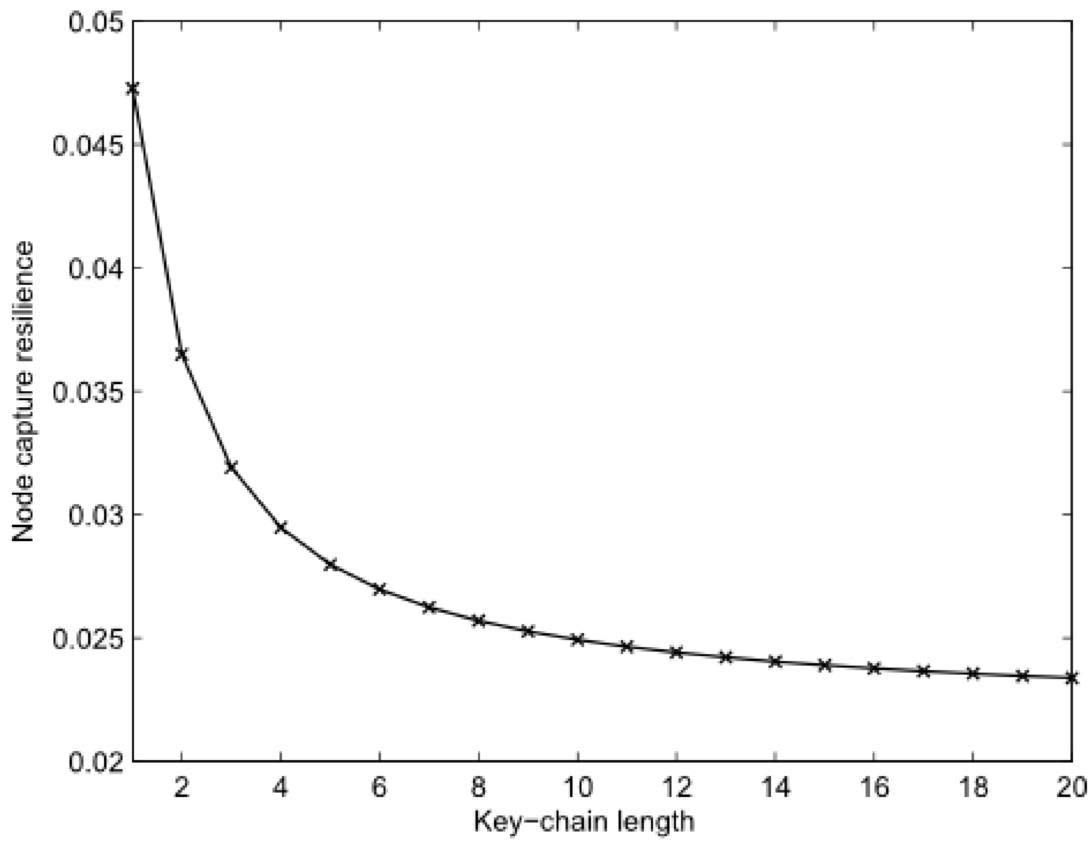
- The longer the chain the better the resilience, but …
- Effective chain le…
  - number of di…
    assigned to s…
  - dependent o…
- Practical value i…

P=0.33

m=200

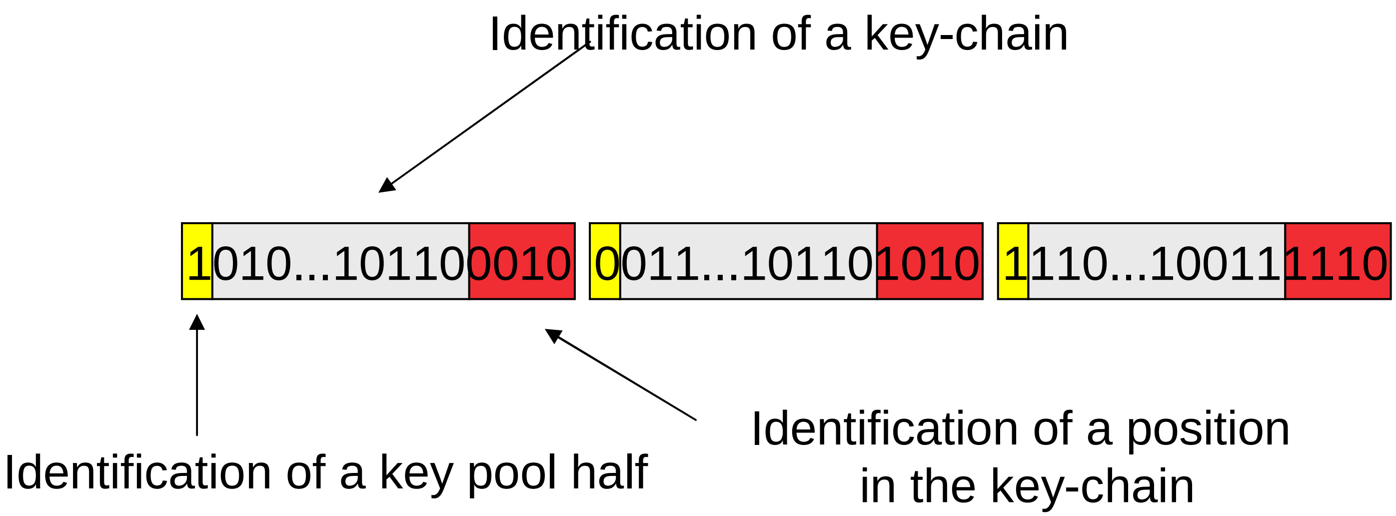
q=2

n=50



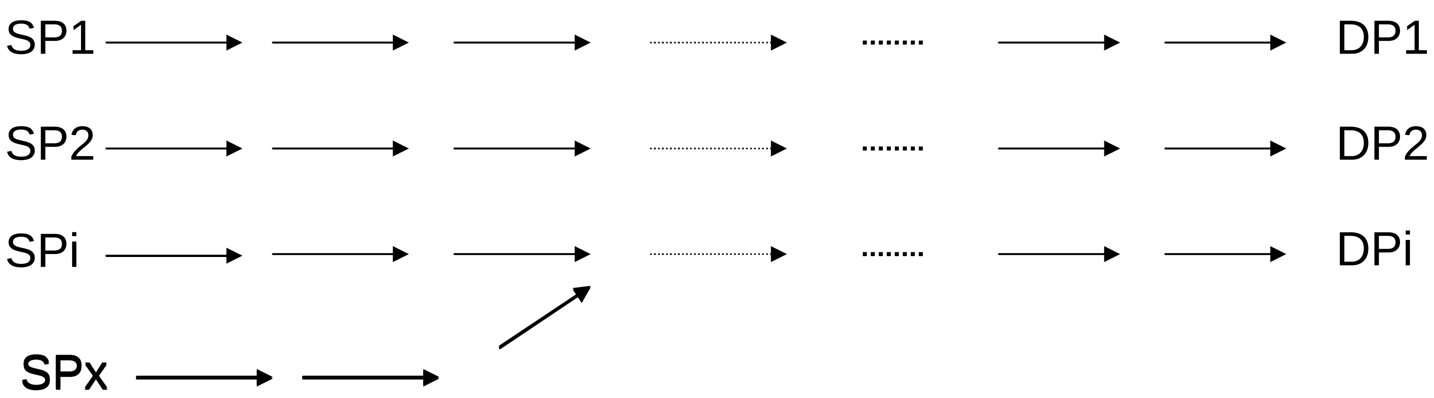
Seed based predistribution

- Generate pseudorandom stream using neighbor ID and pseudorandom number generator



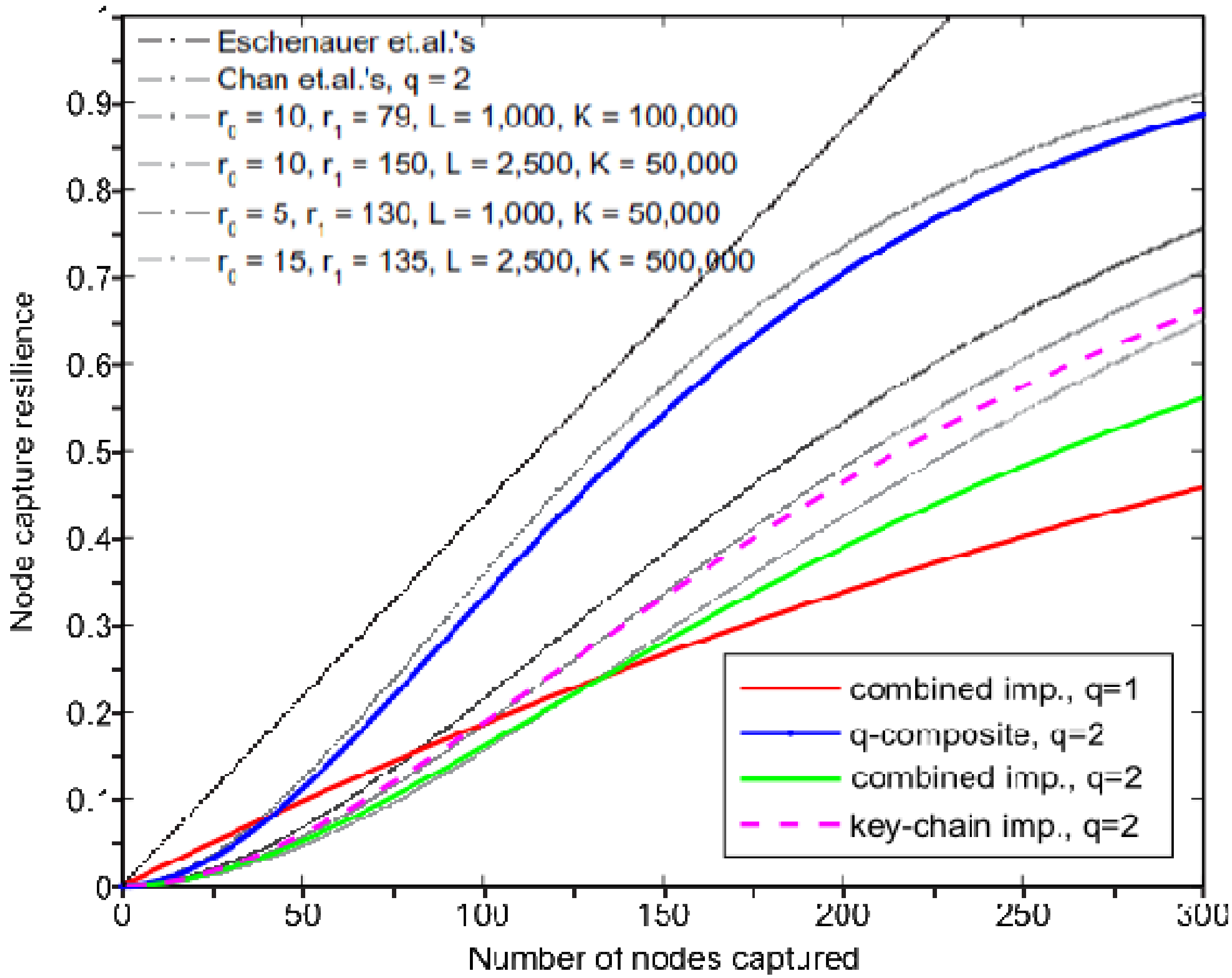
Key pool construction

- Van Oorschot and Wiener's parallel collision search
  - time-memory trade-off approach
- SP – random 80-bit starting point
- DP – 80-bit distinguished point, fixed number of leading zeros
- (SPi,DPi) pairs stored in memory



Comparison with Ren et al.

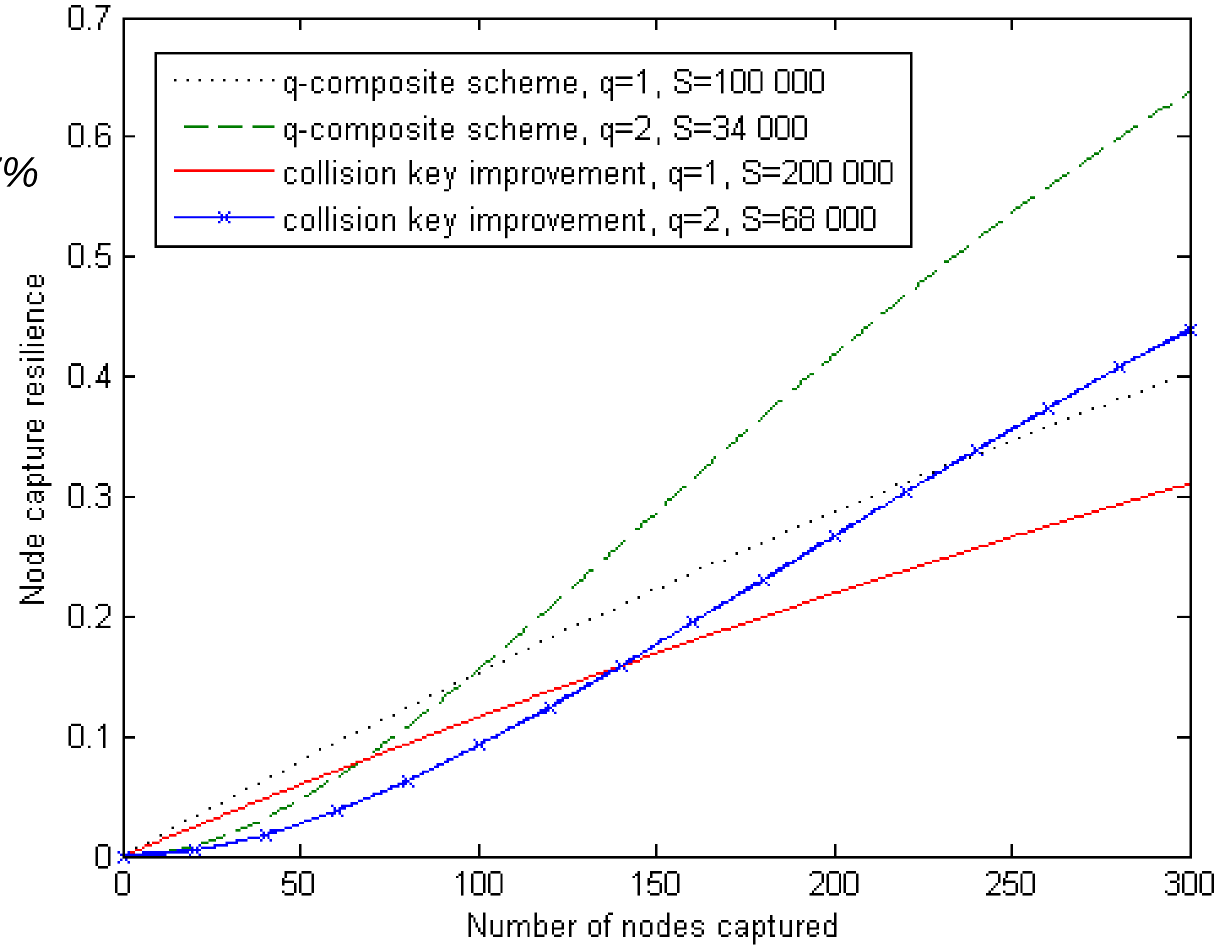
- *P = 0.5, m = 161*



Collision key improvement – evaluation

- *P = 0.33, m = 200*

- *For q = 2 and n = 50*
- *q-composite ncr = 4.7%*
- *Improved ncr = 2.7%*



Key chain improvement - evaluation

- *P = 0.33, m = 200*

- *For q = 2 and n = 50*
- *q-composite ncr = 4.7%*
- *Improved ncr = 2.5%*