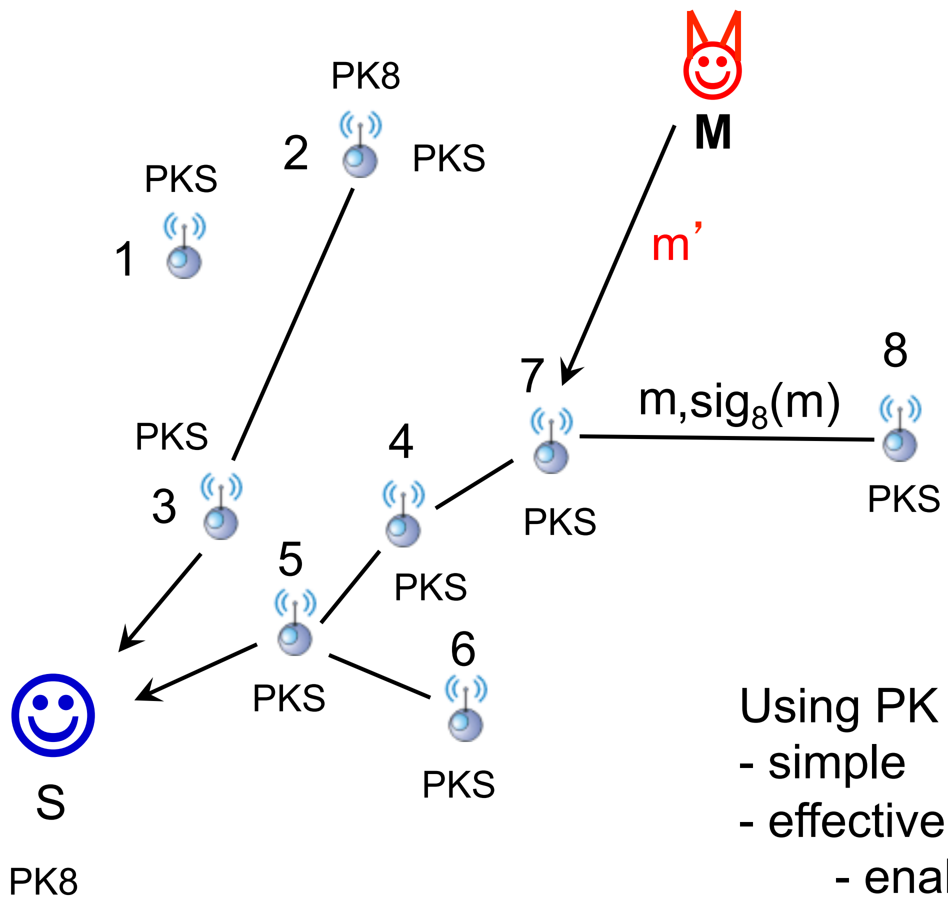
Key Distribution in Sensor Networks

Data integrity, authentication



Using PK crypto in distributed networks is:
- simple
- effective
    - enables broadcast authentication
    - distribution of new keys and insertion of new nodes is straightforward

- expensive

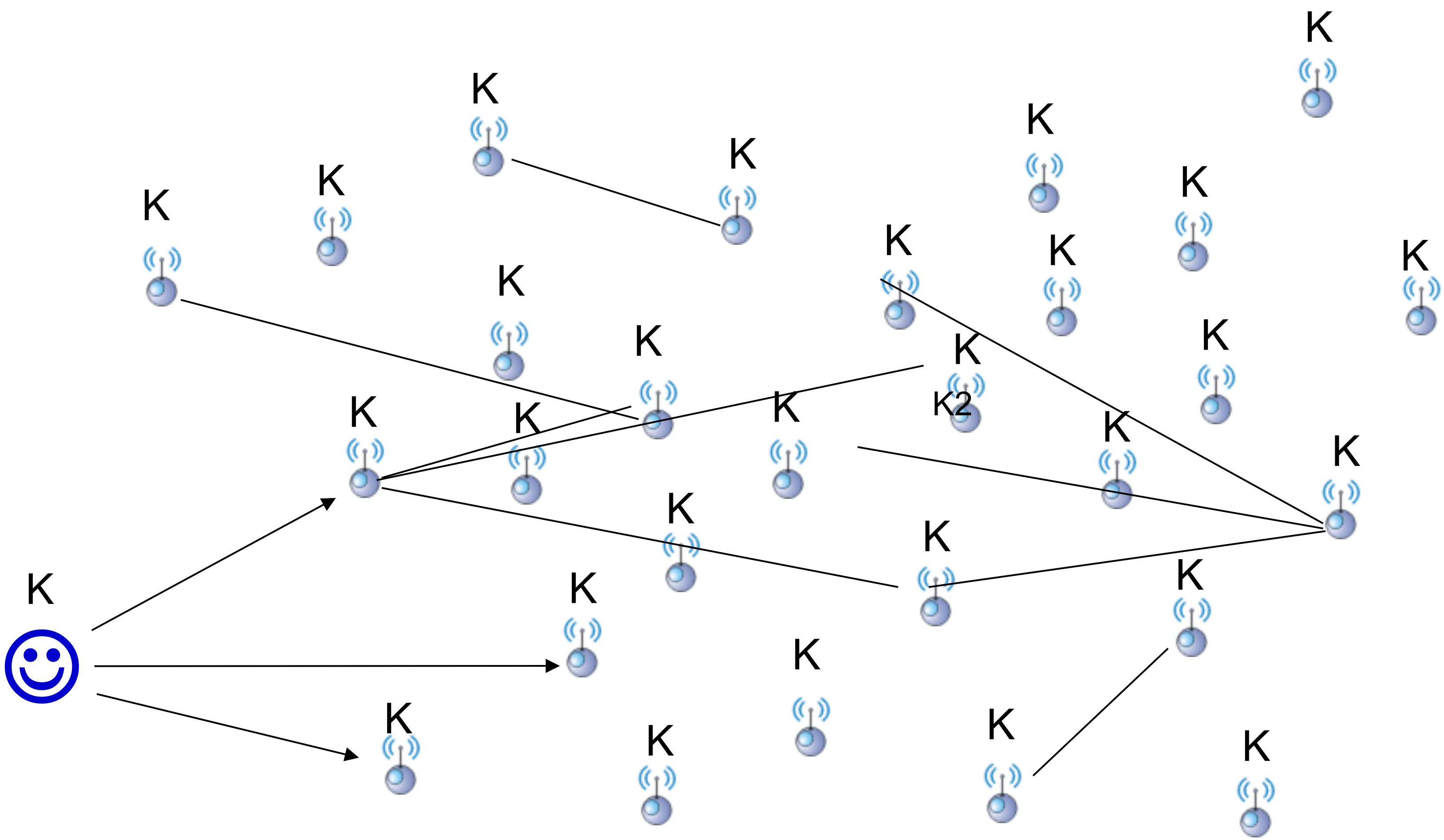
Symmetric-key and PK crypto in sensor nets

- Use PK for all operations
  - \+ simple key distribution
  - \+ simple broadcast authentication
  - – sensors need to be able to perform PK crypto

- PK for key establishment (DH) and SK for the rest
  - \+ simple key distribution
  - – no efficient broadcast authentication
  - – sensors need to be able to perform SK and PK crypto

- Use SK for all operations
  - – **key distribution becomes an issue**
  - – **no efficient broadcast authentication**
  - \+ sensors need to be able to perform only SK crypto

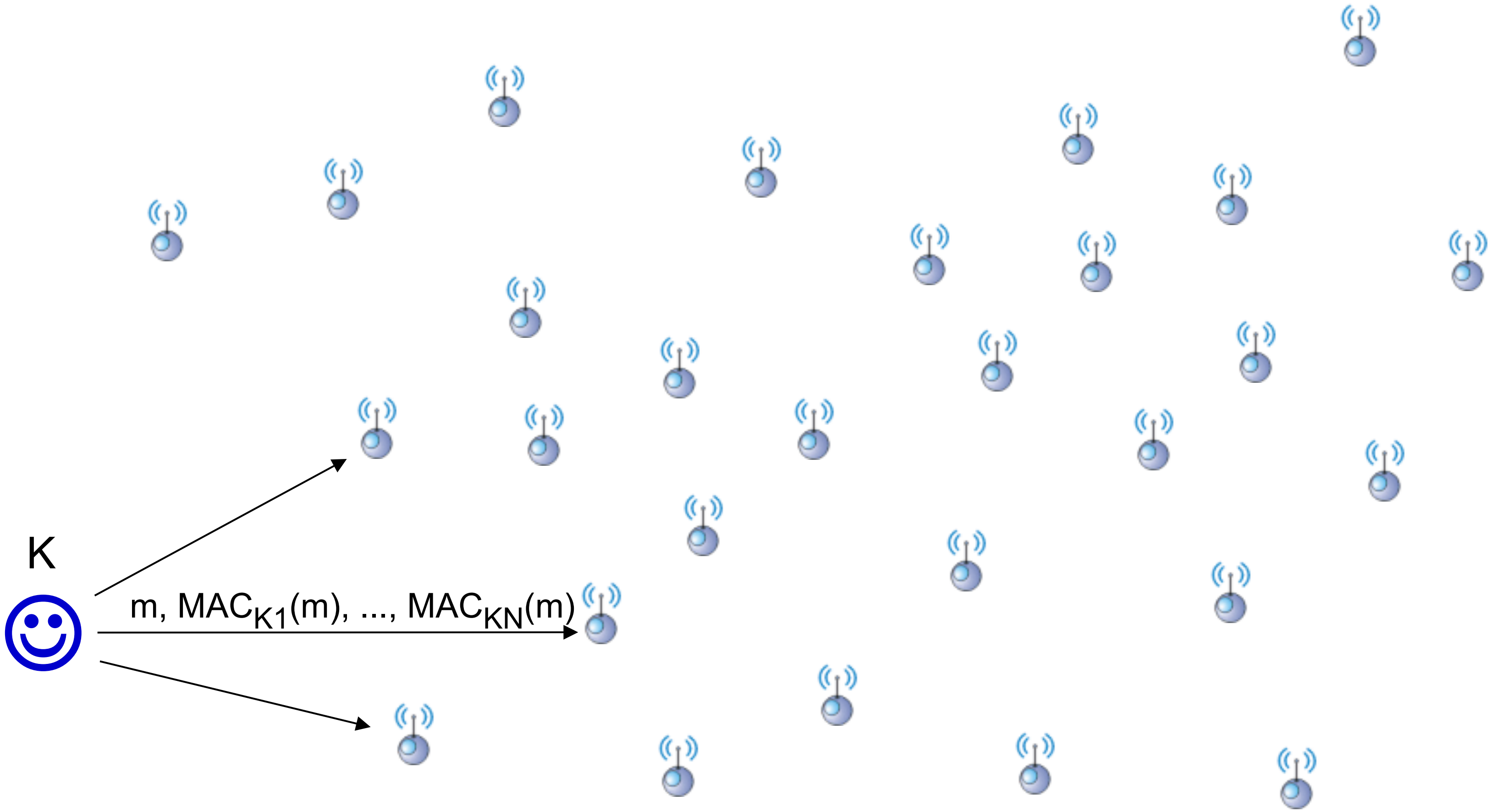
(S)Key distribution in sensor networks [Eschenauer, Gligor]



1 key for all network nodes

+ low storage (1key)
+ efficient broadcast authentication
- no resilience to compromise
- easy to add new nodes

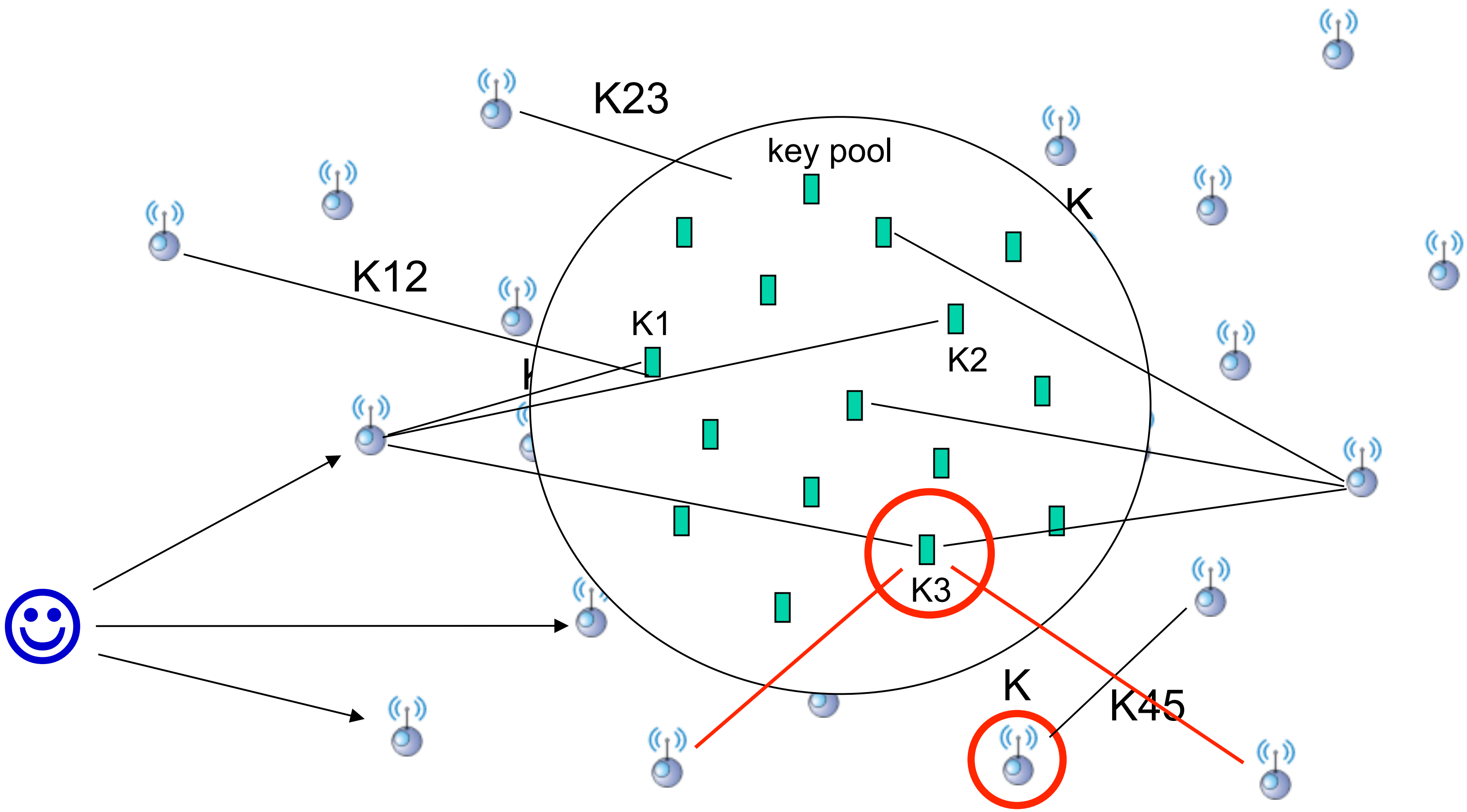
(S)Key distribution in sensor networks [Eschenauer, Gligor]



Each node pair has a different key

- high storage (n keys)
- inefficient broadcast authentication
+ resilience to node compromise
- expensive to add new nodes

(S)Key distribution in sensor networks [Eschenauer, Gligor]



Some node pairs end-up with the same keys
- lower storage (sqrt(n) keys)
- inefficient broadcast authentication
+ some resilience to node compromise
+ easy to add new nodes

(S)Key distribution in sensor networks

Main idea:

- instead of preloading *n* keys in each node, preload just a small subset of values (k<<n) that make sure that most nodes (probabilistic) or all nodes (deterministic) establish keys

Placed between two extremes:

- single master key *(1)*
- distinct pair-wise keys for all node pairs *(n^2)*

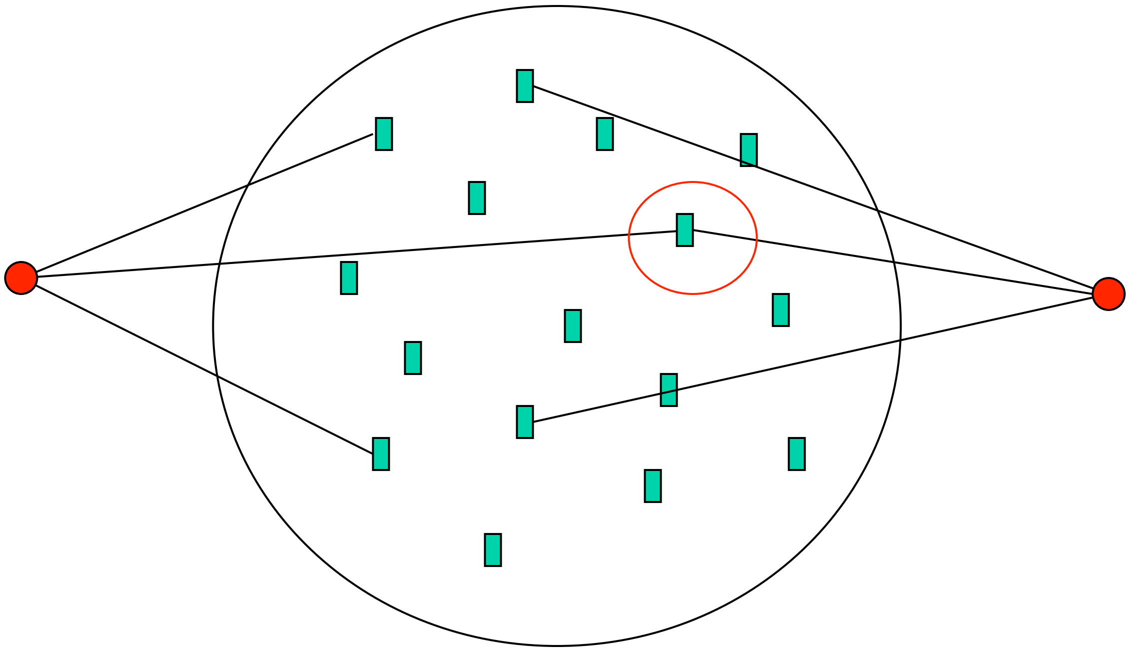
Main issues

- Computation (per key established)
- Communication (per key established)
- Memory (sensor storage)
- Key sharing graph connectivity
- Resiliency (how many sensors need to be compromised before the entire pool is disclosed)
- Scalability

[EG] Scheme

Basic probabilistic key pre-distribution

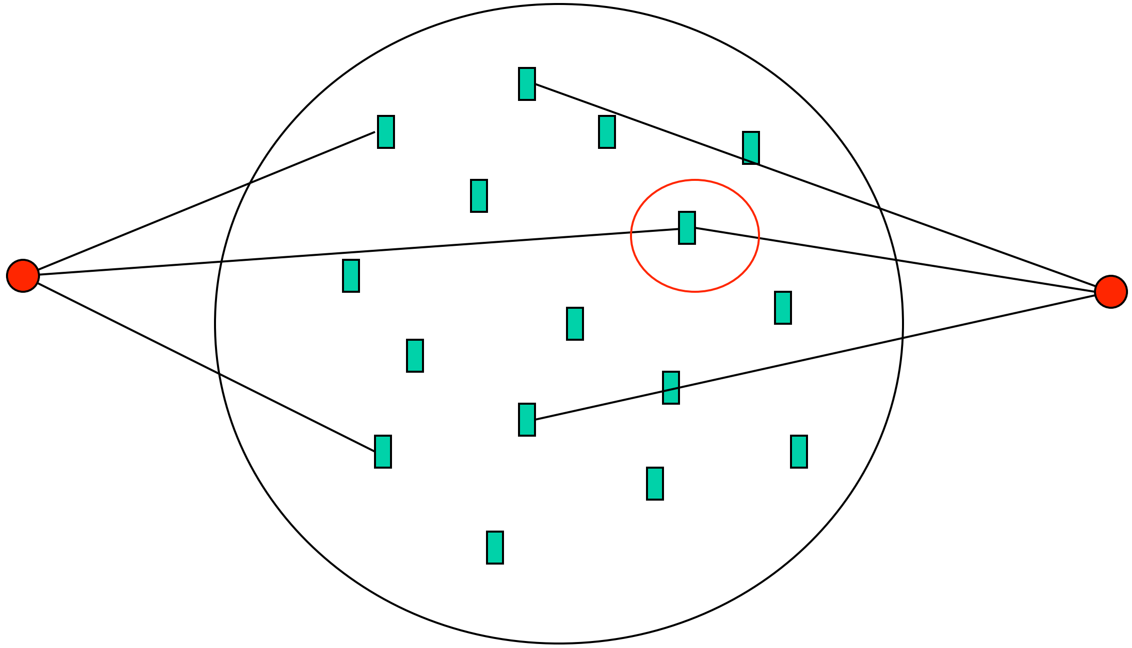
- Eschenauer and Gligor (EG), CCS 2002



k keys in the pool ;   sqrt(k) stored per node

[EG] Scheme

- **Key setup prior to deployment:**
  keys are generated and loaded into memory (the whole pool is known only to the authority)
- **Shared-key discovery after deployment:**
  each sensor node broadcasts a key identifier list to *one-hop neighborhood* (more than one pair may share the same key)
- **Path-key establishment:**
  if two sensor nodes still do not share a key



[EG] Probability of sharing a key

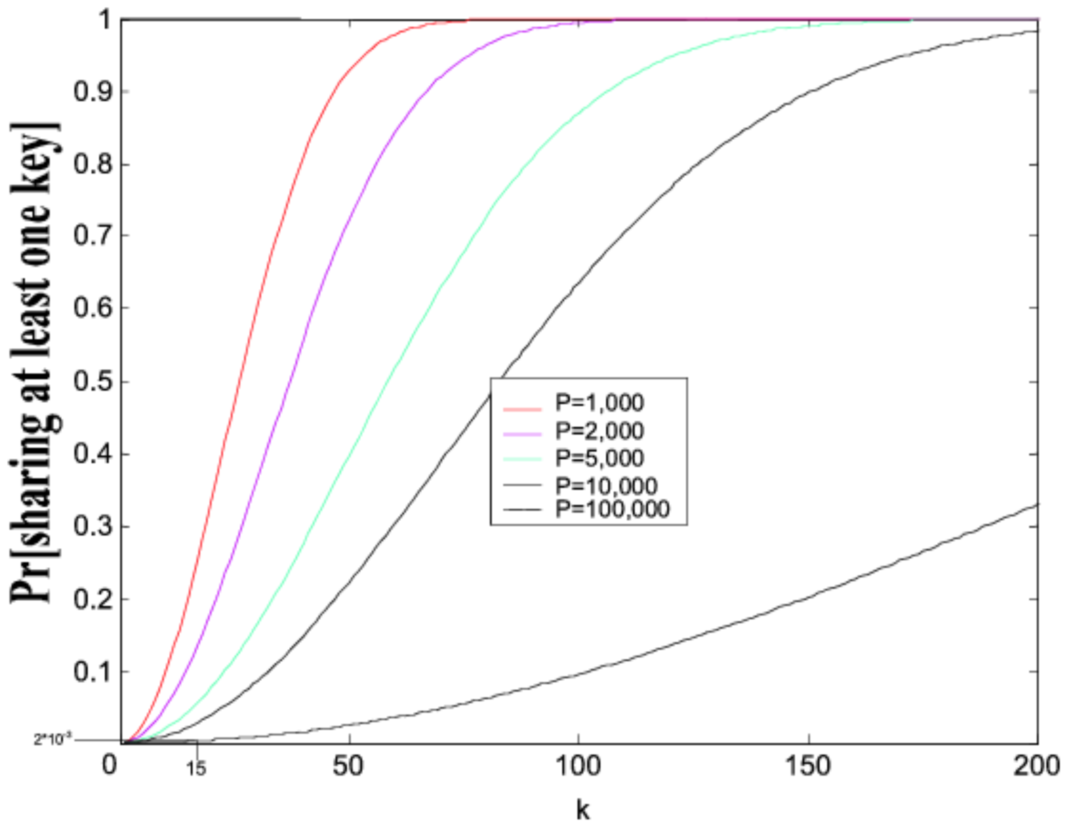
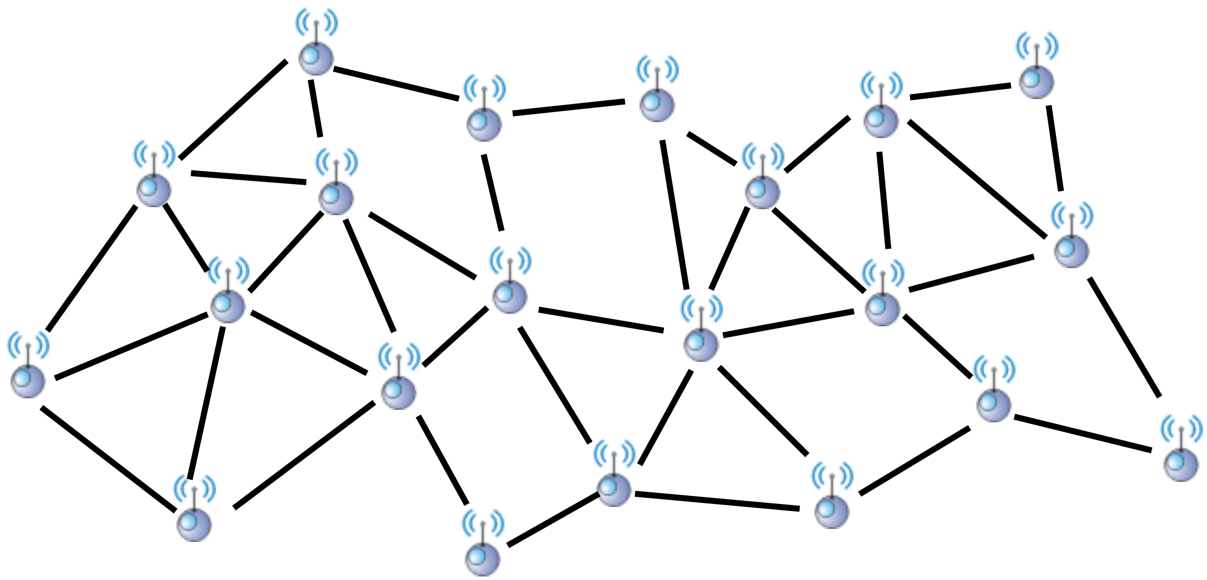


Figure 2: Probability of sharing at least one key when two nodes choose k keys from a pool of size P

[EG] Key Graph and Key Sharing Graph

- Key graph $G_k(V,E)$ is defined as follows:
  - V represents all the nodes in the sensor net
  - For any tow nodes i and j in V, there exists an edge between them if and only if :
    - 1) i and j share at least one common key
- Key sharing graph $G_{sk}(V,E')$
  - i and j have an edge if and only if
    - 1) And 2) They are within wireless transmission range



[EG] Key Graph and Key Sharing Graph

- Key graph $G_k(V,E)$ is defined as follows:
  - V represents all the nodes in the sensor net
  - For any tow nodes i and j in V, there exists an edge between them if and only if :
    - 1) i and j share at least one common key
- Key sharing graph $G_{sk}(V,E')$
  - i and j have an edge if and only if
    - 1) And 2) They are within wireless transmission range

Better connected Key sharing graph = increased communication ability/security

Better connected key graph = increased vulnerability to compromise **...**

[EG] Connectivity vs. Resiliency

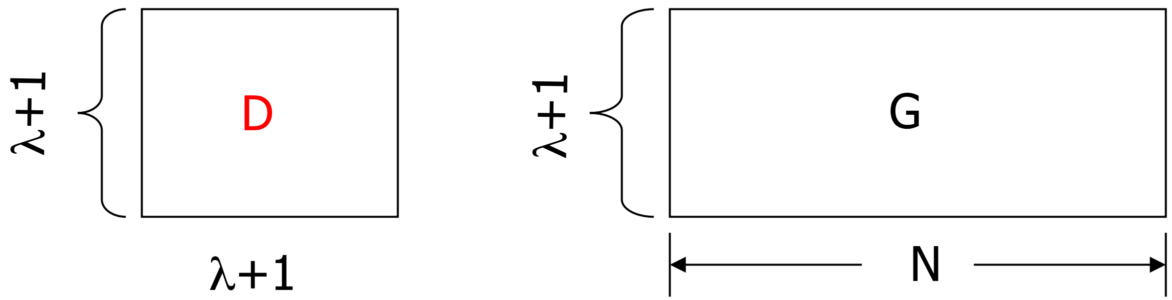
- The contradictory requirement on Key Pool size |P|
    - Larger key pool size – better resiliency
    - Smaller key pool size – better connectivity
- The key pool size is restricted by network size
    - **$|P| < k^2/\ln(1/(1-p))$**
      p is the probability that two nodes share a key (*k* – number of stored keys)
    - **$p > O(\ln N)/n$**
      N is the number of sensor nodes in the network and *n* is the average node degree.
    - As *N* increases, in order to maintain connectivity, *p* would increase, which leads to shrink in *|P|*
- Property of resiliency does not scale with network size
    - *p* should be non decreasing as network enlarges.
    - compromising *k* nodes compromises *kp* links

Deterministic Approaches

- Used to design the key pool and the key chains to provide better connectivity
  - Matrix Based Scheme [Blom 1985]
  - Polynomial Based Key Generation [Blundo et al. 1992]

Deterministic approaches: Blom's Scheme [B]

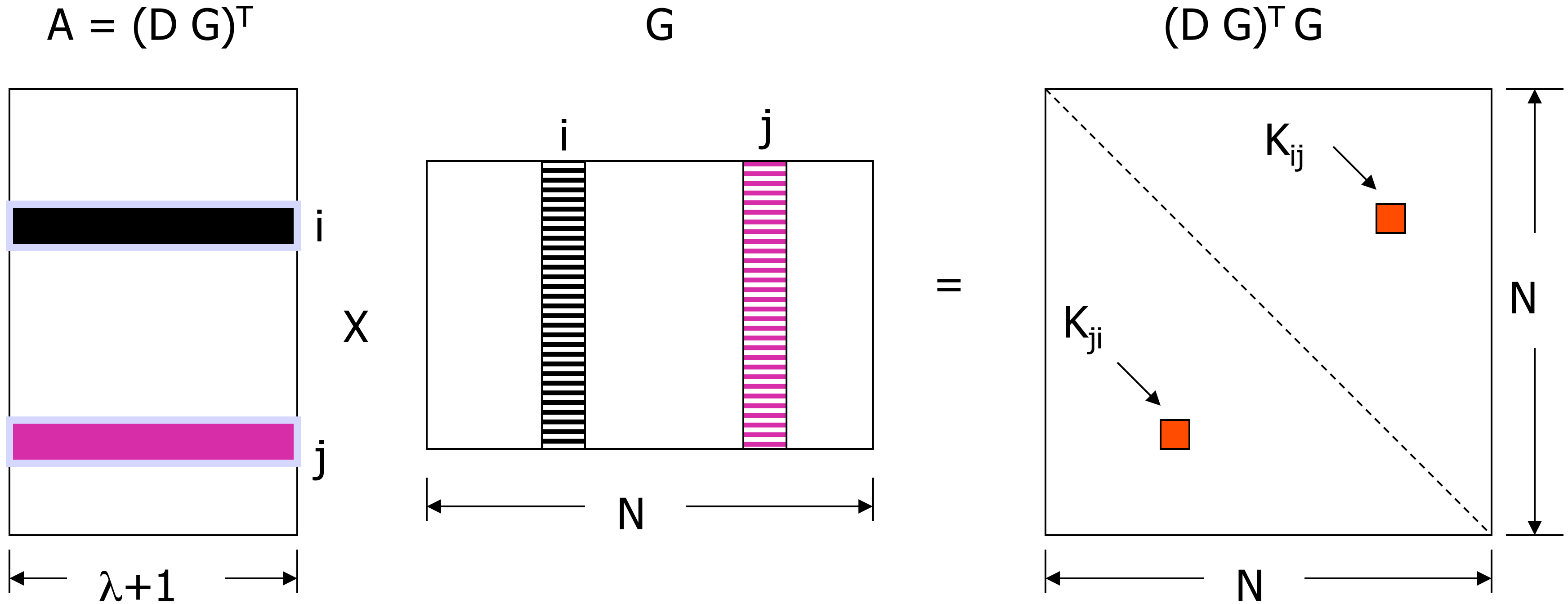
- Public matrix G
- Private matrix D (symmetric).



Let $A = (D\ G)^T$

$$A\ G = (D\ G)^T G = G^T D^T G = G^T D\ G = (A\ G)^T$$

[B] Scheme

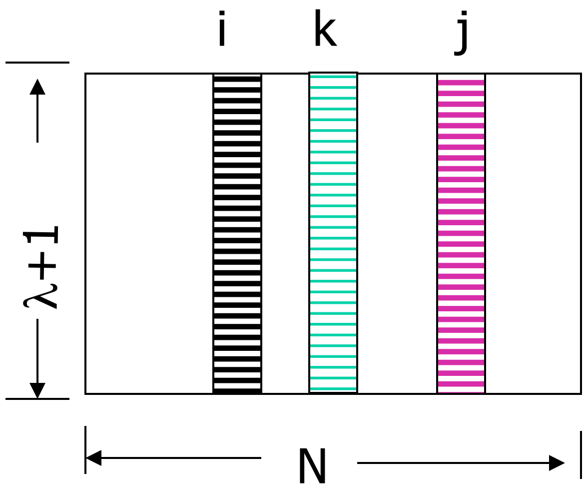


Node i carries:

Node j carries:

[B] λ-secure Property

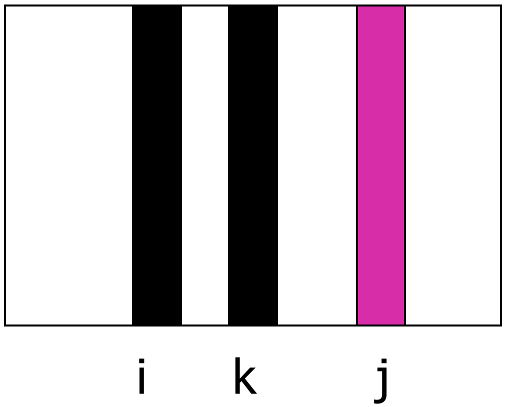
G



Undesirable Situation:
if
u*G(i) + v*G(j) = G(k)

then
u*A(i) + v*A(j) = A(k)

A^T=D G



this would allow
colluding nodes (i and j) to
impersonate other nodes (k)

[B] λ-secure Property

- **ALL** λ**+1** columns in G are linear independent.
  - Different from saying that G has rank λ**+1**
  - **Rank:** there are λ**+1** lineary independent columns
- Can tolerate compromise up to λ nodes.
  - Once λ**+1** nodes are compromised, the rest can be calculated if these λ **+1** columns are linear independent.
- How to find such a matrix G?

[B] Vandermonde Matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^N \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^N)^2 \\ & & \vdots & & \\ s^\lambda & (s^2)^\lambda & (s^3)^\lambda & \cdots & (s^N)^\lambda \end{pmatrix}$$

[B] Properties of Blom Scheme

- Blom's Scheme
  - Network size is N
  - Any pair of nodes can directly find a secret key
  - Tolerate compromise up to λ nodes
  - Need to store λ+2 keys

Key distribution schemes for sensor networks

http://www.cs.rpi.edu/research/pdf/05-07.pdf

Problem	Approach	Mechanism	Keying style	Papers
Pair-wise	Probabilistic	Pre-distribution	Random key-chain	C, E, F, J K, N, S
			Pair-wise key	E
	Deterministic	Pre-distribution	Pair-wise key	G, M
			Combinatorial	P, Q
		Dynamic Key Generation	Master key	D, L
			Key matrix	A
			Polynomial	B, G
	Hybrid	Pre-distribution	Combinatorial	P, Q
		Dynamic Key Generation	Key matrix	H, M, R
			Polynomial	I, R
Group-wise	Deterministic	Dyn. Key Gen.	Polynomial	B, R

The papers are: A[Blom 1985], B[Blundo et al. 1992], C[Eschenauer and Gligor 2002], D[Lai et al. 2002], E[Chan et al. 2003], F[Pietro et al. 2003], G[Liu and Ning 2003c], H[Du et al. 2003], I[Liu and Ning 2003b], J[Zhu et al. 2003], K[Du et al. 2004], L[Dutertre et al. 2004], M[Lee and Stinson 2004b], N[Hwang et al. 2004], P[Camtepe and Yener 2004], Q[Lee and Stinson 2004a], R[Huang et al. 2004], S[Hwang and Kim 2004].