NC STATE UNIVERSITY Computer Science

CSC 774 Network Security

Topic 9.1 Key Predistribution in Wireless Sensor Networks (2)

Outline

- Background
  - Polynomial based key predistribution
- A framework for key predistribution in sensor networks
  - Polynomial pool based key predistribution
- Two efficient key predistribution schemes
  - Random subset assignment
  - Grid based key predistribution
- Efficient implementation in sensor networks
- Conclusion and future work

Polynomial Based Key Predistribution

- By Blundo et al. [CRYPTO '92]
  - Developed for group key predistribution
  - We consider the special case of pairwise key predistribution
- Predistribution:
  - The setup server randomly generates $f(x,y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$, where $f(x,y) = f(y, x)$
  - Each sensor i is given a *polynomial share* $f(i, y)$
- Key establishment:
  - Node i computes $f(i, y = j) = f(i, j)$
  - Node j computes $f(j, y = i) = f(j, i) = f(i, j)$

Polynomial Based Key Predistribution (Cont'd)

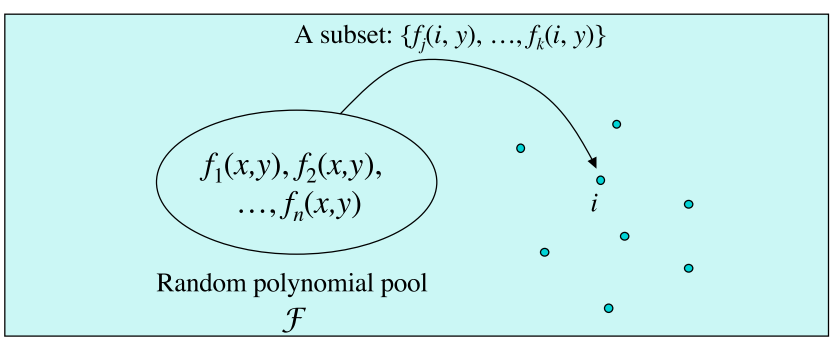
- Security properties (by Blundo et al.)
  - Unconditionally secure for up to t compromised nodes
- Performance
  - Storage overhead at sensors: $(t+1)\log q$ bits
  - Computational overhead at sensors: t modular multiplications and t modular additions
  - No communication overhead
- Limitation
  - Insecure when more than t sensors are compromised
  - An invitation for node compromise attacks

Polynomial Pool Based Key Predistribution

- A general framework for key predistribution based on bivariate polynomials
  - Let us use multiple polynomials
    - A pool of randomly generated bivariate polynomials
- Two special cases
  - One polynomial in the polynomial pool
    - Polynomial based key predistribution
  - All polynomials are 0-degree ones
    - Key pool by Eschenauer and Gligor

Polynomial Pool Based Key Predistribution (Cont'd)

- Phase 1: Setup
  - Randomly generates a set $\mathcal{F}$ of bivariate t-degree polynomials
  - Subset assignment: Assign a subset of polynomials in $\mathcal{F}$ to each sensor



Polynomial Pool Based Key Predistribution (Cont'd)

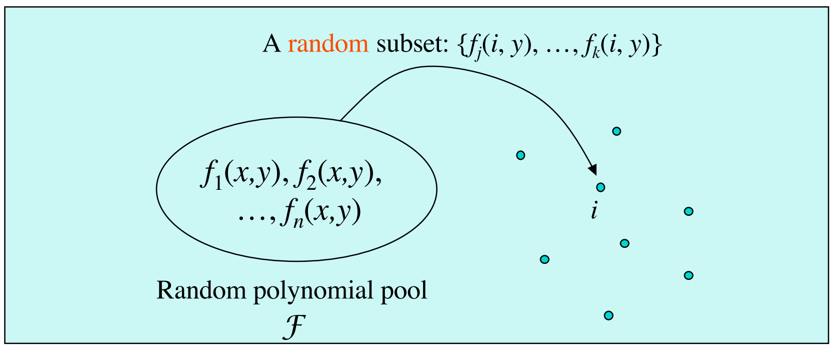
- Phase 2: Direct Key Establishment
  - Polynomial share discovery: Communicating sensors discover if they share a common polynomial
    - Pairwise keys can be derived if they share a common polynomial.
  - Two approaches:
    - Predistribution:
      - Given predistributed information, a sensor can decide if it can establish a direct pairwise key with another sensor.
    - Real-time discovery:
      - Sensors discover on the fly if they can establish a direct pairwise key.

Polynomial Pool Based Key Predistribution (Cont'd)

- Phase 3: Path Key Establishment
  - Establish pairwise keys through other sensors if two sensors cannot establish a common key directly
  - Path discovery
    - Node *i* finds a sequence of nodes between itself and node *j* such that two adjacent nodes can establish a key directly
    - Key path: the above sequence of nodes between *i* and *j*
  - Two approaches
    - Predistribution
      - Node *i* can find a key path to node *j* based on predistributed information
    - Real-time discovery
      - Node *i* discover a key path to node *j* on the fly

Random Subset Assignment Scheme

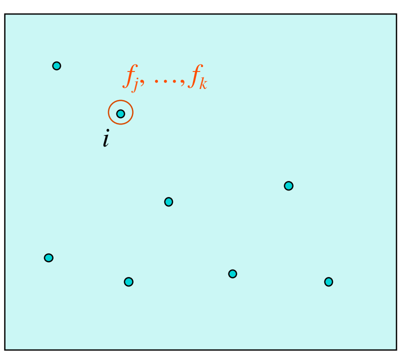
- An instantiation of the polynomial pool-based key predistribution.
- Subset assignment: random



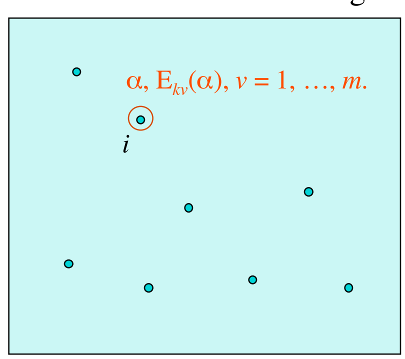
Random Subset Assignment (Cont'd)

- Polynomial share discovery
  - Real-time discovery

Broadcast IDs in clear text.

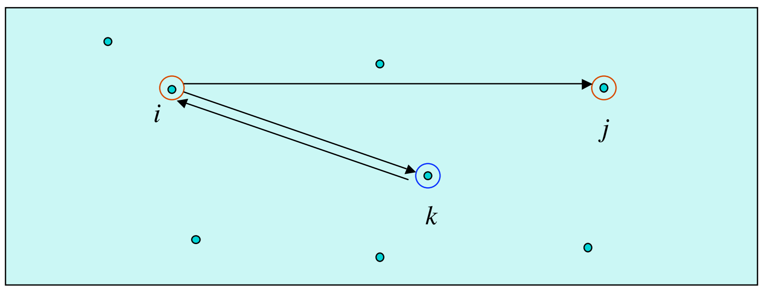


Broadcast a list of challenges.

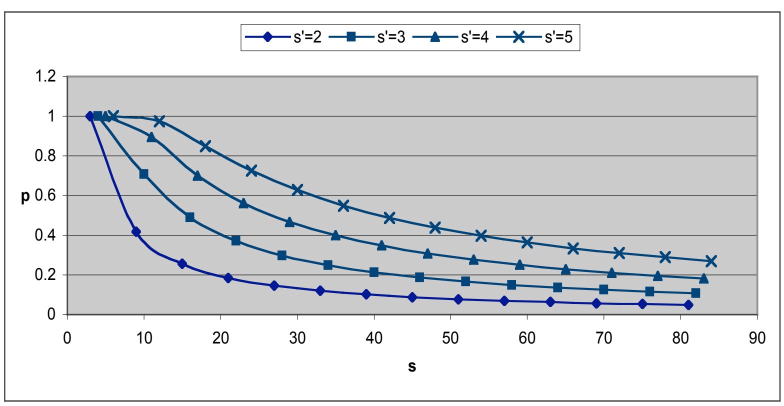


Random Subset Assignment (Cont'd)

- Path discovery
  - *i* and *j* use *k* as a KDC
  - Alternatively, *i* contacts nodes with which it shares a key; any node that also shares a key with *j* replies.
  - Each key path has 2 hops

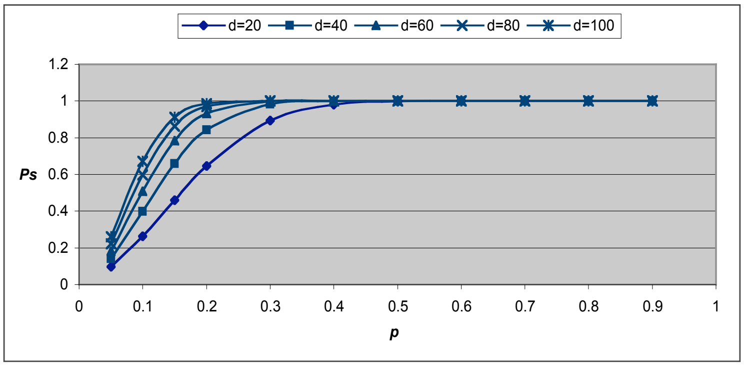


Probability of Sharing Direct Keys between Sensors



- *s*: polynomial pool size
- *s'*: number of polynomial shares for each sensor
- *p*: probability of sharing a polynomial between two sensors

Probability of Sharing Keys between Sensors

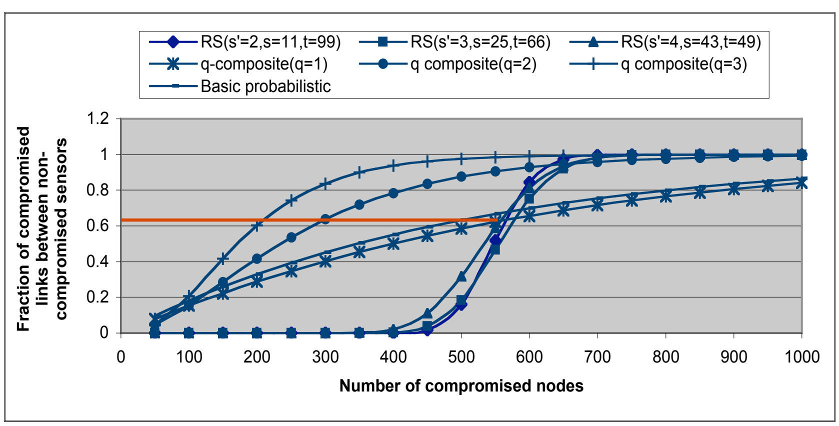


- *d*: number of neighbors
- *p*: probability that two sensors share a polynomial
- p_s: probability of sharing a common key

Note: each key path is at most two hops

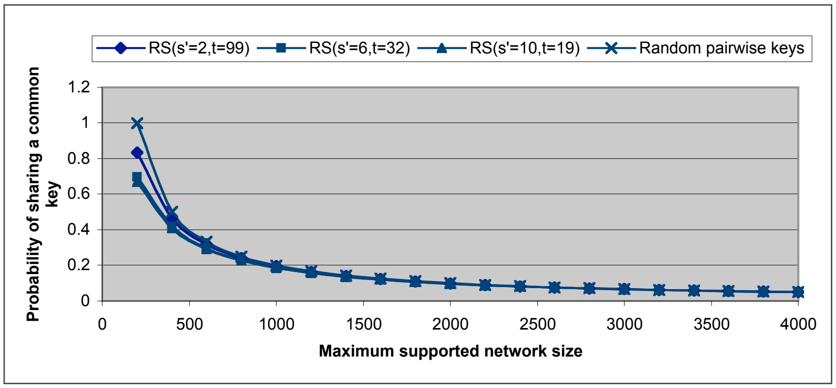
Dealing with Compromised Sensors

- Comparison with basic probability and *q*-composite schemes
  - Probability to establish direct keys $p = 0.33$
  - Each sensor has storage equivalent to 200 keys



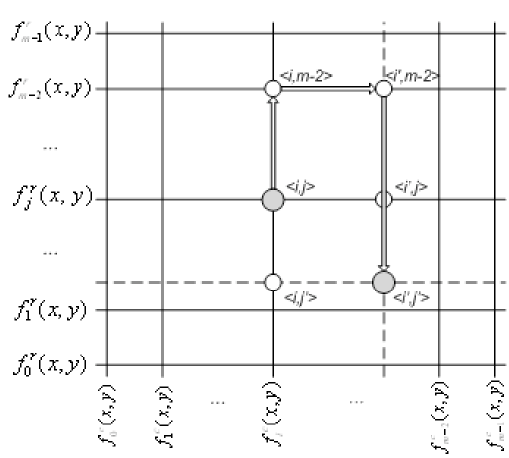
Dealing with Compromised Sensors (Cont'd)

- Comparison with random pairwise keys scheme
  - Assume perfect security against node compromises
    - Each polynomial is used at most *t* times in our scheme
  - Each sensor has storage equivalent to 200 keys



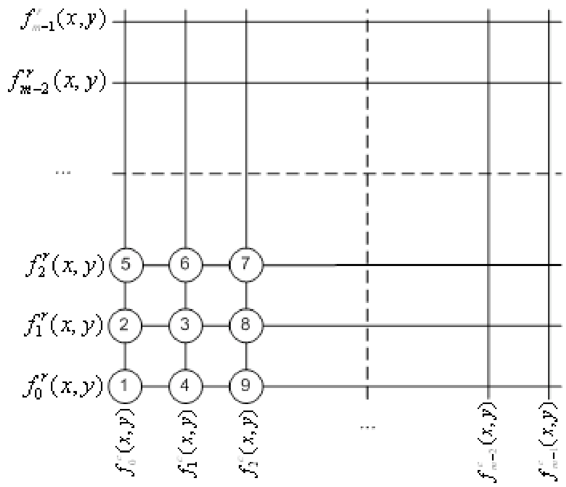
Grid Based Key Predistribution

- Create a $m \times m$ grid
- Each row or column is assigned a polynomial
- Assign each sensor to an interaction
- Assign each sensor the polynomials for the row and the column of its intersection
  - Sensor ID: coordinate
- There are multiple ways for any two sensors to establish a pairwise key



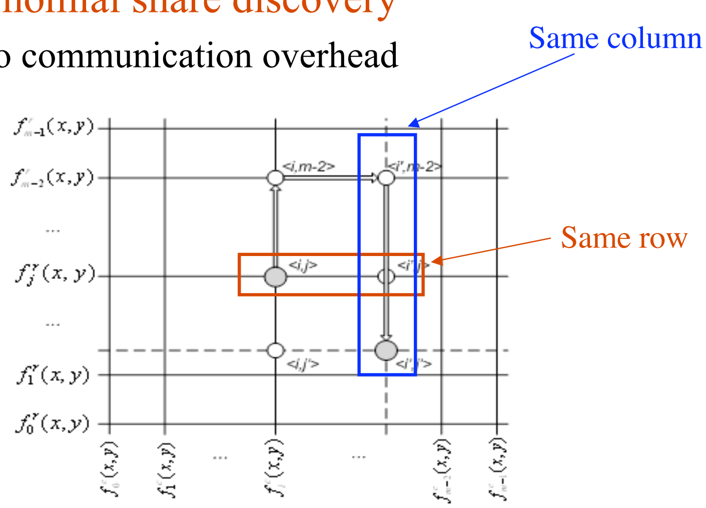
Grid Based Key Predistribution (Cont'd)

- Order of node assignment



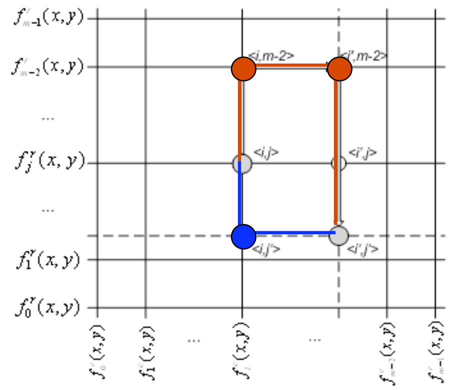
Grid Based Key Predistribution (Cont'd)

- Polynomial share discovery
  - No communication overhead



Grid Key Predistribution (Cont'd)

- Path discovery
  - Real-time discovery
  - Paths with one intermediate node
  - Paths with two intermediate nodes
  - They know who to contact!

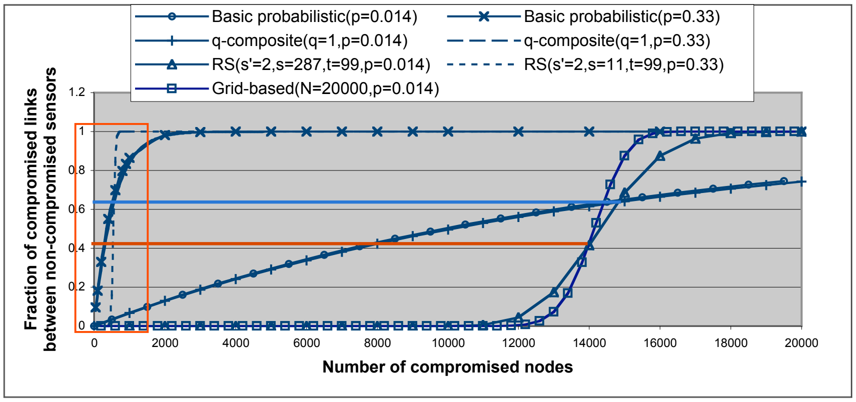


Properties

1. Any two sensors can establish a pairwise key when there is no compromised node;
2. Even if some sensors are compromised, there is still a high probability to establish a pairwise key between non-compromised sensors;
3. A sensor can directly determine whether it can establish a pairwise key with another node.

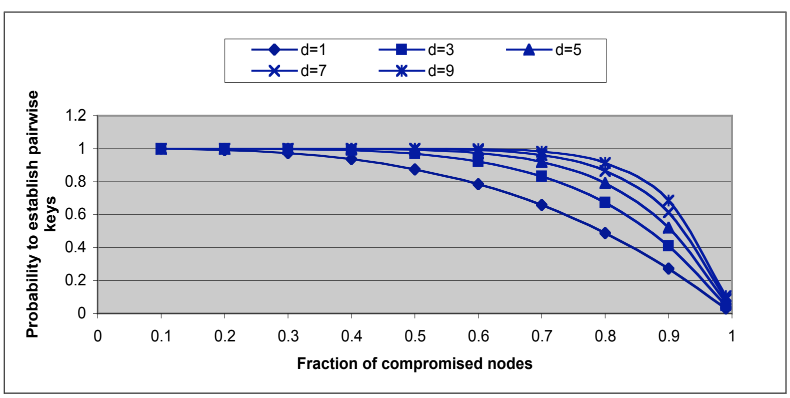
Dealing with Compromised Sensors

- Comparison with basic probabilistic scheme, *q*-composite scheme, and random subset assignment scheme
  - Assume each sensor has storage equivalent to 200 keys



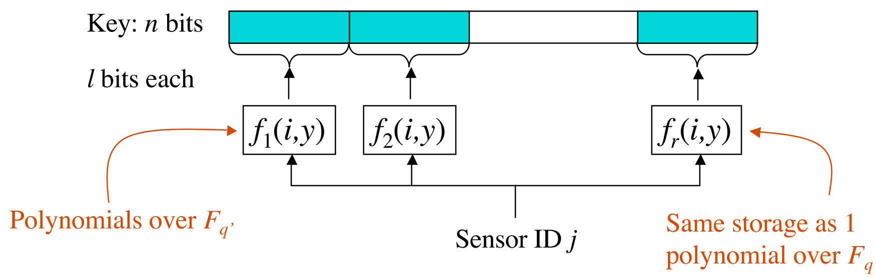
Dealing with Compromised Sensors (Cont'd)

- Probability to establish pairwise keys when there are compromised sensors
  - *d*: number of non-compromised sensors to contact
  - Assume each sensor has storage equivalent to 200 keys



Implementation

- Observations
  - Sensor IDs are chosen from a field much smaller than cryptographic keys
    - Field for cryptographic keys: F_q
    - Field for sensor IDs: $F_{q'}$
  - Special fields: $q'=2^{16}+1$, $q'=2^8+1$
    - No division operation is needed for modular multiplications



Implementation (Cont'd)

- Lemma 1. In this implementation, the entropy of the key for a coalition of no more than t other sensors is

$$r \cdot [\log_2 q' - (2 - \frac{2^{l+1}}{q'})]$$

where $l = \lfloor \log_2 q' \rfloor$ and $r = \left\lceil \frac{n}{l} \right\rceil$.

- Examples
  - 64 bit keys
  - When $q'=2^{16}+1$, the above entropy is 63.9997 bits
  - When $q'=2^8+1$, the above entropy is 63.983 bits