Location-Aware Key Distribution in WSNs

Dr. Ashok Kumar Das

IEEE Senior Member
Associate Professor
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: *ashok.das@iiit.ac.in*
URL: `http://www.iiit.ac.in/people/faculty/ashokkdas`
`https://sites.google.com/view/iitkgpakdas/`

Location-Aware Key Distribution in WSNs

The closest pairwise keys scheme (CPKS)

- The closest pairwise keys scheme (CPKS) proposed by Liu and Ning is a key pre-distribution scheme of location-awareness in nature.
- In the key pre-distribution phase, for each sensor node u to be deployed in the target field, the key setup server first determines a set S of m nodes whose expected locations of deployment are closest to that of u.
- For every node $v \in S$, for which a pairwise key between u and v has not already been assigned by the setup server, a new random symmetric key k_{uv} is generated.
- The key-plus-id combination (k_{uv}, v) is loaded to u's key ring, whereas the pair (k_{uv}, u) is loaded to v's key ring.

The closest pairwise keys scheme (CPKS)

- In direct key establishment phase, two neighboring nodes, say, *u* and *v* can establish a secure communication link, if they have a pre-distributed pairwise key.
- To identify a common key is trivial, because each pairwise key in a particular node is accompanied by the id of the other nodes holding the key.
- A cryptographic handshake may be then performed by the nodes *u* and *v* for mutual verification of the common key shared between them.

Network connectivity of CPKS

- Assume that the deployment field is two dimensional. Let u be a sensor node whose expected location be (u_x, u_y), whereas its actual location be (u'_x, u'_y). This corresponds to a deployment error $e = (u'_x - u_x, u'_y - u_y)$.
- Liu and Ning showed that the network connectivity of CPKS depends upon the deployment error e. If the maximum deployment error e is small, CPKS provides significantly better connectivity than the random schemes.
- They have also shown that for sufficiently large errors, CPKS essentially degrades to the random pairwise keys scheme (EG scheme) which has very poor connectivity when the network size is larger.

Network connectivity derivation

- For the sake of simplicity, we assume that the target field is two-dimensional, so that every point in that region is expressed by two co-ordinates x and y.
- Assume that u is a sensor node whose expected location is (u_x, u_y) whereas its actual location is (u'_x, u'_y). This corresponds to a deployment error of $e_u = (u'_x - u_x, u'_y - u_y)$.
- The actual location (or equivalently the error e_u) can be modeled as a continuous random variable that can assume values in R^2.
- The probability density function $f_u(u'_x, u'_y)$ of (u'_x, u'_y) characterizes the pattern of deployment error.
- Let (u'_x, u'_y) is uniformly distributed within a circle with center at (u_x, u_y) and radius e called the *maximum deployment error*. We have:

$$f_u(u'_x, u'_y) = \begin{cases} \frac{1}{\pi e^2} & \text{if } (u'_x - u_x)^2 + (u'_y - u_y)^2 \leq e^2 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Network connectivity derivation

- Another strategy is to model (u'_x, u'_y) as a random variable following the two-dimensional normal (Gaussian) distribution with mean (u_x, u_y) and variance σ^2. The corresponding probability density function is:

$$f_U(u'_x, u'_y) = \frac{1}{2\pi\sigma^2} e^{-[(u'_x - u_x)^2 + (u'_y - u_y)^2]/(2\sigma^2)}. \quad (2)$$

- However, for the sake of simplicity, we only consider the uniform distribution given in Equation (1).
- Two nodes are called *physical neighbors* if they lie in each other's communication range. They are called *key neighbors* if they possess shares of a common key. They are called *direct neighbors* if they are both physical and key neighbors.
- Let u and v be two deployed nodes. Assume that each node has a communication range ρ and that the different nodes are deployed independently i.e., (u'_x, u'_y) and (v'_x, v'_y) are independent random variables.

Network connectivity derivation

- The probability that u and v are in each other's communication range can be calculated by

$$p(u,v) = \iiiint_C f_u(u'_x, u'_y)\, f_v(v'_x, v'_y)\, du'_x\, du'_y\, dv'_x\, dv'_y \qquad (3)$$

  where C is the region $(u'_x - v'_x)^2 + (u'_y - v'_y)^2 \leq \rho^2$.
- Since u can share pairwise keys with c nodes, the expected value of ρ' is given by $\rho' = \rho \times \sqrt{\frac{c}{d+1}}$.
- Let v be a key neighbor of u. Then, the probability that v lies in the physical neighborhood of u is given by

$$p(u) = \frac{1}{\pi \rho'^2} \iint_C p(u,v)\, dx\, dy \qquad (4)$$

Network connectivity derivation

- Again, since u is expected to have $c \times p(u)$ direct neighbors, the probability that u can establish a pairwise key with one of its physical neighbor is given by

$$p = \frac{p(u) \times c}{d} \approx p(u) \times \lambda \tag{5}$$

  where $\lambda = \frac{c}{d+1}$.
- We take the communication range ρ as the basic unit of distance measurement, i.e., $\rho = 1$. One can compute the probability p for the density function given above and establish that $p \approx 1$ for small deployment errors.

Resilience against node capture attack of CPKS

- We note that each predistributed pairwise key $k_{u,v}$ between two neighbor nodes u and v is randomly generated.
- Thus, no matter how many nodes are captured, the pairwise keys between non-compromised sensor nodes remain still secure.
- This means that no matter how many sensor nodes are captured, the non-compromised nodes can communicate with each other with 100% secrecy.
- In this way, CPKS provides unconditional security against node capture attacks.

The closest polynomials pre-distribution scheme (CPPS)

- The closest polynomials pre-distribution scheme (CPPS) proposed by Liu and Ning is a location-aware scheme.
- It is based on the polynomial-pool based scheme which uses priori deployment knowledge of the deployed sensor nodes.
- The deployment field is partitioned into equal-sized square cells $C_{i,j}$ containing R rows and C columns.
- The four adjacent neighboring cells of a cell $C_{i,j}$ are considered as the cells $C_{i,j-1}$, $C_{i-1,j}$, $C_{i,j+1}$, and $C_{i+1,j}$. For example, if a node u is expected to locate in cell $C_{2,2}$, the four adjacent neighboring cells of its home cell are $C_{2,1}$, $C_{1,2}$, $C_{2,3}$, and $C_{3,2}$.

The closest polynomials pre-distribution scheme (CPPS)

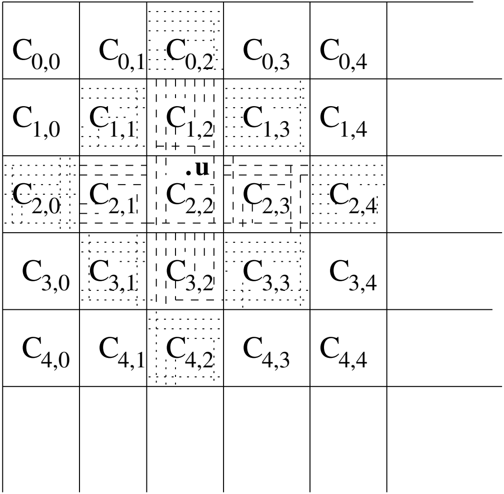

Figure: Partition of a target field.

The closest polynomials pre-distribution scheme (CPPS)

- In the key predistribution phase, the key setup server chooses $s = R \times C$ random symmetric t-degree bivariate polynomials $f_{i,j}(x, y) \in F_q[x, y]$, $i = 1, 2, \ldots, R$, $j = 1, 2, \ldots, C$, where $F_q = GF(q)$ is a finite field with q large enough to accommodate a cryptographic key.
- Here $GF(q)$ stands for the Galois field of order q, where q is either a prime or of the form 2^m for some positive integer m.
- Assume that the expected deployment location of a node u lies in the cell $C_{i,j}$ called the *home cell* of u. The key ring of u is loaded with the shares of the five polynomials corresponding to the home cell and the four neighboring cells.
- That is, u is given the five polynomial shares: $f_{i,j}(u, y)$, $f_{i-1,j}(u, y)$, $f_{i+1,j}(u, y)$, $f_{i,j-1}(u, y)$, $f_{i,j+1}(u, y)$. The key set-up server also stores in u's memory the id (i, j) of its home cell.

The closest polynomials pre-distribution scheme (CPPS)

- Thus, we note that each sensor node's key ring contains five t-degree polynomial shares before its deployment in the target field.
- Since each t-degree polynomial share consists of $(t+1)$ coefficients which are from F_q, so the storage requirement for this polynomial share is $(t+1)\log_2(q)$ bits.
- Hence, the storage requirement for each sensor node is $5(t+1)$ $\log_2(q)$ bits in order to store its keying informations.
- If each symmetric key is taken of q bits, the storage requirement is equivalent to storing $5(t+1)$ symmetric keys.

The closest polynomials pre-distribution scheme (CPPS)

- In direct key establishment phase, each node u broadcasts its home cell's id (i, j) (or some messages encrypted by potential pairwise keys) to its immediate physical neighbors. Suppose u and v are two neighbors.
- From the co-ordinates of the home cell (i, j) of the source node u, the destination node v can immediately determine the ids of the polynomial shares held by the source node u.
- Let u and v find a common polynomial, say, f. Then the source node u computes a common pairwise key shared with the destination node v as $f(u, v)$. Similarly, the destination node v also computes a common pairwise key shared with the source node u as $f(v, u)$.
- Since $f(u, v) = f(v, u)$, they store this value $f(u, v)$ as the secret key shared between them for their future communication. A cryptographic handshake may be performed between neighbor nodes u and v for mutual verification of the common key $f(u, v)$.

Network connectivity of CPPS

- Similar to the analysis of CPKS, the network connectivity of CPPS also depends on the deployment error.
- Larger error leads to poorer connectivity. Moreover, the network connectivity of CPPS depends on the length L of cell side.
- Liu and Ning showed that if L is chosen larger, there is a higher probability of establishing a direct key between two neighbors. Thus, smaller L also leads to poorer connectivity.
- If the maximum deployment error is small, CPPS also provides significantly better connectivity than the random schemes.

Resilience against node capture attack of CPPS

- As long as no more than t polynomial shares of a bivariate polynomial are disclosed, an attacker can know nothing about the non-compromised pairwise keys established using this polynomial.
- Thus, the security of this scheme depends on the average number of nodes sharing the same polynomial, or equivalently on the number of nodes that are expected to be located in each cell and its four adjacent cells.
- If that number is larger than t, CPPS is not unconditionally secure.
- Again, in CPPS, if the length L of cell side is larger, it leads to compromise of a larger number of sensor nodes sharing the same bivariate polynomial, which in turn degrades the security performance.
- As a result, CPPS is unconditionally secure and t-collusion resistant.