Efficient Key Predistribution for Grid-Based Wireless Sensor Networks

Simon R. Blackburn[1] Keith M. Martin[1] Tuvi Etzion[2]
Maura B. Paterson[1]

[1]Information Security Group
Royal Holloway, University of London

[2]Technion -Israel Institute of Technology
Department of Computer Science

ICITS 2008

Outline

Grid-Based Networks

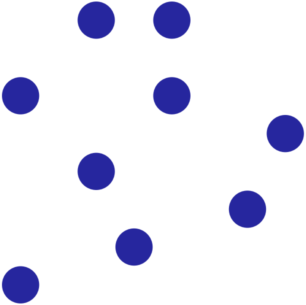
Costas Arrays

A New KPS for Grid-Based Sensor Networks

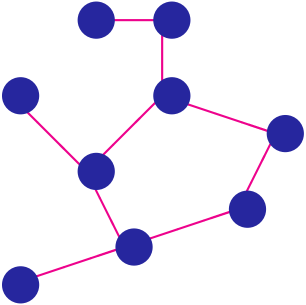
Evaluation of KPSs for Grid-Based Networks

Wireless Sensor Networks

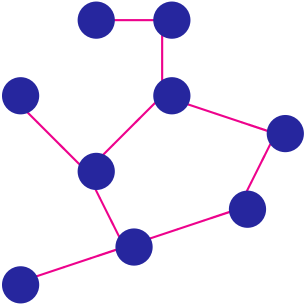
Wireless Sensor Networks



Wireless Sensor Networks

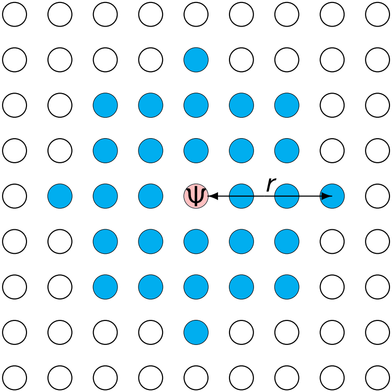


Wireless Sensor Networks

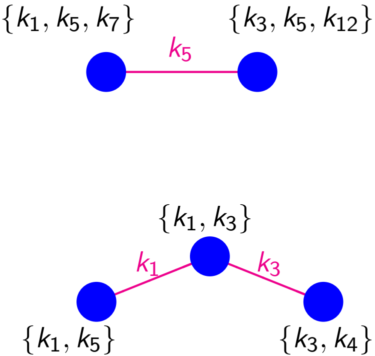


- restricted memory
- restricted battery power
- restricted computational ability
- vulnerable to compromise

Grid-Based Wireless Sensor Networks



Key Predistribution



key predistribution scheme (KPS)

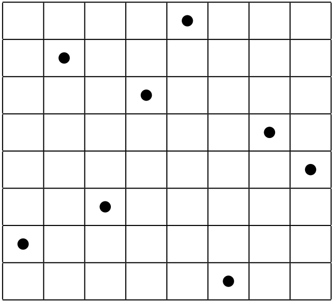
- nodes are assigned keys before deployment
- nodes that share keys can communicate securely
- two-hop path: nodes communicate via intermediate node

Goals for a KPS in a Grid-Based Network

- enable any two neighbours to communicate securely (directly or using a two-hop path)
- minimise storage
- be resilient against node compromise

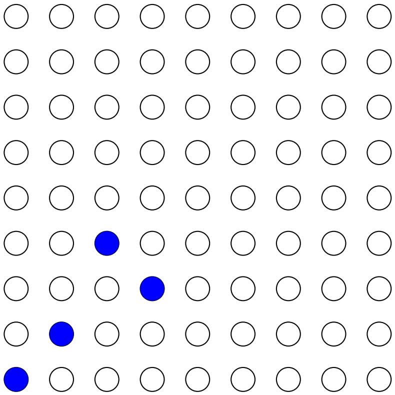
Observation: it is not necessary for two nodes to share more than one key

Costas Arrays

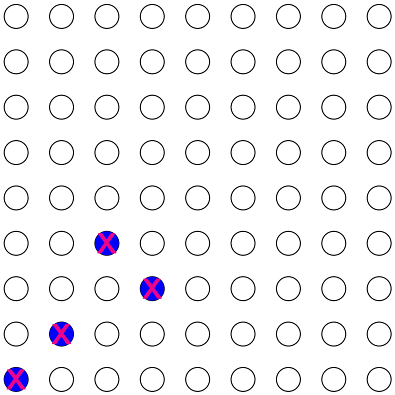


- one dot per row/column
- vector differences between dots are distinct
- applications to sonar, radar
- known constructions are based on finite fields

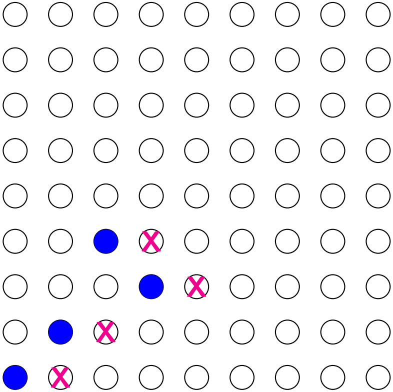
Translated Costas Arrays Overlap in at Most One Point



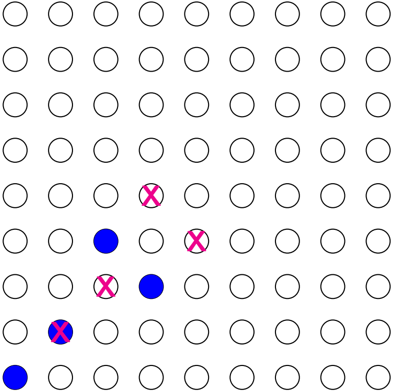
Translated Costas Arrays Overlap in at Most One Point



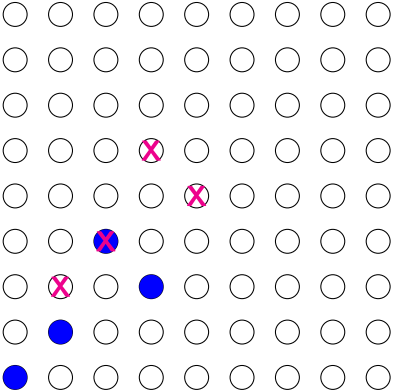
Translated Costas Arrays Overlap in at Most One Point



Translated Costas Arrays Overlap in at Most One Point

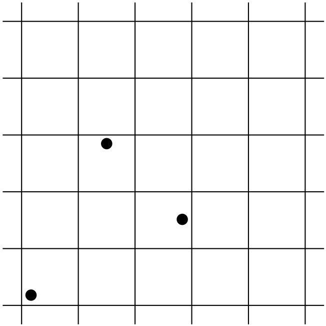


Translated Costas Arrays Overlap in at Most One Point



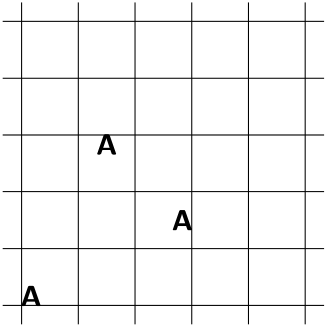
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



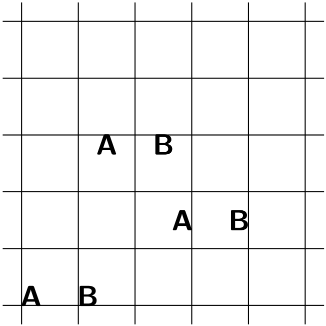
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



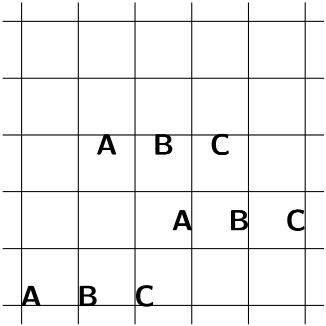
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



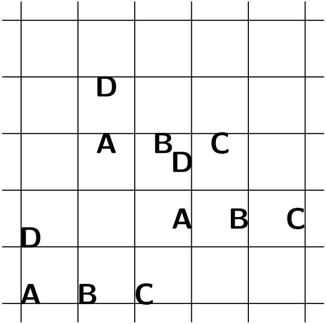
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



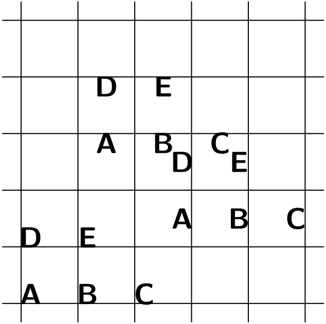
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



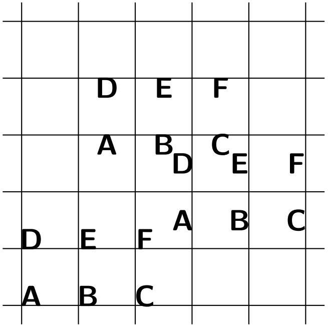
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



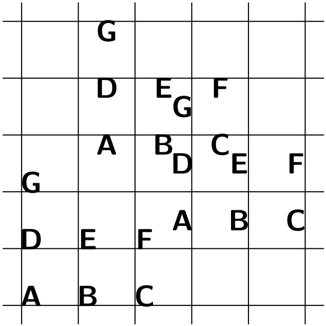
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



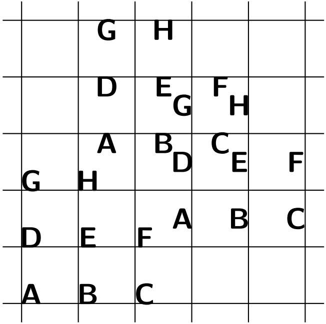
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



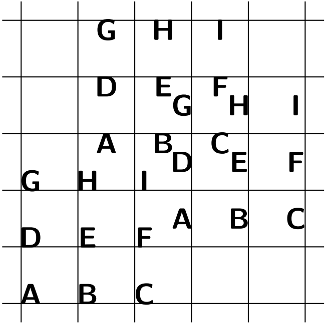
Key Predistribution Using Costas Arrays

- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



Key Predistribution Using Costas Arrays

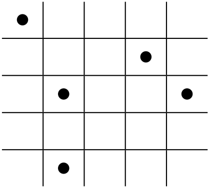
- uses an $n \times n$ Costas array
- each sensor stores n keys
- each key is assigned to n sensors
- two sensors share at most one key
- the distance between two sensors that share a key is at most $\sqrt{2}(n-1)$



Distinct-Difference Configurations

Definition (Distinct-Difference Configuration DD(m, r))

- m dots are placed in a square grid
- the distance between any two dots is at most r
- vector differences between dots are all distinct



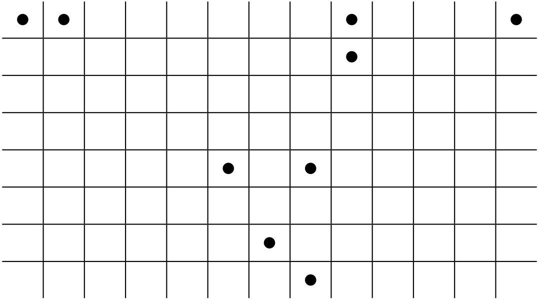
DD$(5, 8)$

- can be used for key predistribution in the same way as a Costas array
- more general than a Costas array $\Rightarrow$ more flexible choice of parameters

Maximum Two-Hop Coverage of a DD(m, r)

$m \backslash r$	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	4	4	4	4	4	4	4	4	4	4
3	-	12	18	18	18	18	18	18	18	18	18	18
4	-	-	28	46	54	54	54	54	54	54	54	54
5	-	-	28	48	80	102	118	126	130	130	130	130
6	-	-	-	48	80	112	148	184	222	240	254	262
7	-	-	-	-	80	112	148	196	252	302	346	374
8	-	-	-	-	-	112	148	196	252	316	376	$\geq$432
9	-	-	-	-	-	-	148	196	252	316	376	440
10	-	-	-	-	-	-	-	196	252	316	376	440
11	-	-	-	-	-	-	-	-	252	316	376	440
12	-	-	-	-	-	-	-	-	-	316	376	440

An optimal DD$(9, 12)$



- When $r = 12$ a node has 440 neighbours.
- A KPS based on this array ensures nodes can communicate securely with all 440 neighbours via a one-hop or two-hop path.
- This scheme requires each node to store 9 keys.

Comparison with Existing Schemes -Connectivity

Scheme	m	α	One-hop	Two-hop
Costas	8	8	56	366
DD(8, 11)	8	8	56	376
Liu & Ning	8	2	8	24
Eschenauer & Gligor	8	≈ 200	56.2	370.0
Ito *et al.*	8	≈ 8	36.2	319.6

values are averaged over 10000 trials on a 100×100 square grid

Comparison with Existing Schemes -Resilience

Scheme	m	α	Resilience	Local Res.
Costas	8	8	331	59
DD$(8, 11)$	8	8	336	59
Liu & Ning	8	2	23.87	20.3
Eschenauer & Gligor	8	≈ 200	36	36
Ito *et al.*	8	≈ 8	259	52

values are averaged over 10000 trials on a 100×100 square grid