# Efficient Key Predistribution for Grid-Based Wireless Sensor Networks*

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

[1] Department of Mathematics
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, U.K.
{s.blackburn,keith.martin,m.b.paterson}@rhul.ac.uk
[2] Technion -Israel Institute of Technology
Department of Computer Science
Technion City, Haifa 32000, Israel
etzion@cs.technion.ac.il

**Abstract.** In this paper we propose a new key predistribution scheme for wireless sensor networks in which the sensors are arranged in a square grid. We describe how Costas arrays can be used for key predistribution in these networks, then define *distinct difference configurations*, a more general structure that provides a flexible choice of parameters in such schemes. We give examples of distinct difference configurations with good properties for key distribution, and demonstrate that the resulting schemes provide more efficient key predistribution on square grid networks than other schemes appearing in the literature.

**Keywords:** wireless sensor networks, key predistribution, costas arrays.

## 1 Introduction

Wireless sensors are small, battery-powered devices with the ability to take measurements of quantities such as temperature or pressure, and to engage in wireless communication. When a collection of sensors is deployed the sensors can communicate with each other and thus form an ad hoc network, known as a *wireless sensor network* (WSN), in order to facilitate the transmission and manipulation of data by the sensors. Such networks have a wide range of potential applications, including wildlife monitoring or pollution detection (see Römer and Mattern [33] for some examples of how they have been used in practice).

For many applications it is desirable to encrypt communications within the network, since wireless communication is highly vulnerable to interception. The limited memory and battery power of sensors means that for many purposes symmetric techniques are preferred to more computationally intensive public

* This research was partly carried out under EPSRC grant EP/E034632/1.
** This author was supported by EPSRC grant EP/D053285/1.

key operations. Thus sensors must share secret keys, in order to provide authentication, confidentiality, or data integrity. One method for enabling this is for the sensors' keys to be preloaded prior to deployment. This technique is known as *key predistribution*.

Much of the literature on key predistribution in wireless sensor networks deals with the case where the physical topology of the network is completely unknown prior to deployment [3–7, 9, 10, 12, 13, 18, 19, 21–24, 26, 28–31, 35]. In practice, however, many sensor network applications involve networks for which there is some degree of control (indeed, often complete control) over the sensors' locations. Key predistribution is particularly effective in such networks, as the location knowledge can be harnessed to develop more efficient schemes. For instance, it may be possible to reduce the number of keys shared by pairs of nodes that cannot physically communicate. Not only does this reduce the amount of keying material that must be stored, but it improves the resiliency of the network: an adversary learns fewer keys when capturing a given number of nodes, and those keys it does learn tend to be shared only by nodes in a restricted neighbourhood of those captured nodes. Also, a priori knowledge of location reduces the need for nodes to undergo location discovery or neighbour discovery; this may reduce or even eliminate any communication overheads in the key setup process, particularly in the case where there is some regularity or symmetry to the sensors' distribution.

While there are several examples of location-based schemes appearing in the literature [8–11, 17, 20, 25, 34], in the majority of cases the networks consist of randomly distributed nodes whose approximate location is known. In [27], Martin and Paterson give an indication of the types of networks that have been considered in the WSN key predistribution literature, and suggest that there is considerable scope for the development of schemes suited to specific network topologies, in situations where the topology is known before sensor deployment.

In this paper we consider the particular case of a network where the sensors are arranged in a square grid. There are many potential applications in which such a pattern may be useful: monitoring vines in a vineyard or trees in a commercial plantation or reforestation project, studying traffic or pollution levels on city streets, measuring humidity and temperature at regular intervals on library shelves, performing acoustic testing at each of the seats in a theatre, monitoring goods in a warehouse, indeed any application where the objects being studied are naturally distributed in a grid. For purposes of commercial confidentiality or for protecting the integrity of scientific data it is necessary to secure communication between sensors, and thus it is important to have efficient methods of distributing keying material in such networks. The goal of this paper is to provide some practical key predistribution schemes designed specifically for square grids. We show that the highly structured topology of these networks can be exploited to develop schemes that perform significantly better for this application than more general techniques, such as those of Eschenauer and Gligor [13]. Our schemes are designed for *homogeneous* networks in which all sensors have the same capabilities. We assume the nodes have no access to an external trusted authority (such as a base station) for the purposes of establishing keys once they have been deployed. We assume that the location of each node within the grid is known prior to deployment, and consider the problem of establishing pairwise keys between nodes within communication distance of one another. This setting can be described in the language of [27] as that of a locally 2-complete scheme for a network with fixed sensors and full location control.

In the following section, we discuss the desirable properties for key predistribution schemes based on square grids. In Sect. 4 we describe a key predistribution scheme based on *Costas arrays*, and we introduce the concept of *distinct-difference configurations* and use them to generalise our scheme. In Sect. 5 we discuss certain important properties of KPSs, and in Sect. 6 we compare the behaviour of our schemes to that of several schemes from the literature. We show that our schemes outperform these previously studied schemes under our network model.

## 2 The Network Model

We say that a wireless sensor network is *grid based* if it consists of a (potentially unbounded) number of identical sensors arranged in a square grid.

If each sensor has a maximum transmission range $r$ then a sensor is able to communicate directly with all nodes within the circle of radius $r$ that surrounds it. (We say that two squares occur at distance $r$ if the Euclidean distance between the centres of the squares is $r$.) Without loss of generality we can scale our unit of distance so that adjacent nodes in the grid are at distance 1 from each other; we will adopt this convention throughout this paper as it removes unnecessary complications from our discussions.

We refer to nodes within the circle of radius $r$ centred at some node $\Psi$ as *$r$-neighbours* of $\Psi$. For most applications it is useful for any two neighbouring nodes in a sensor network to be able communicate securely. In designing a KPS, however, we are restricted by the limited storage capacity of the sensors: if a node has many neighbours, it may be unable to store enough keys to share a distinct key with each neighbour. We would like to design key predistribution schemes in which each node shares a key with as many of its $r$-neighbours as possible, while taking storage constraints into account. (Note that we only require keys to be shared by nodes that are $r$-neighbours, in contrast to a randomly distributed sensor network which potentially requires all pairs of nodes to share keys.) One way of achieving this is for each key to be shared by several different nodes; however, it is necessary to restrict the extent to which each key is shared, to protect the network against key compromise through node capture.

In Sect. 4 we propose a construction for KPSs that seek to balance the competing requirements discussed in this section. First, however, we describe a combinatorial structure that we will use in this construction.

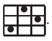
## 3 Costas Arrays

Costas arrays were first introduced for use in the detection of sonar signals (see [16]), and have received much attention for this and other applications (an extensive bibliography can be found at [32]). To the best of our knowledge, the KPS we propose in Sect. 4 represents the first time these structures have been used for key distribution. In this section we provide basic definitions and properties of these arrays, and briefly describe some known constructions.

**Definition 1.** *A* Costas array of order $n$ *is an* $n \times n$ *matrix with the following properties:*

- *each position is either blank or contains a dot,*
- *each row and each column contains exactly one dot,*
- *all* $n(n-1)$ *vectors connecting pairs of dots are distinct as vectors (any two vectors are different in either length or direction).*

*Example 1.*



This is an example of a Costas array of order 3. It is easily seen that the six vectors connecting pairs of dots are distinct.

The application of Costas arrays in sonar or radar relies on the fact that if a translation is applied to a copy of a Costas array then at most one dot of the translated array coincides with a dot of the original array, unless the two are exactly superimposed. It is this property that motivates our use of Costas arrays in constructing KPSs. We formalise it as follows.

**Lemma 1.** *Let* $S = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n\}$ *be the set of positions of the dots in a Costas array* $\mathcal{A}$*. Suppose the array* $\mathcal{A}$ *is translated by a vector* $\mathbf{v}$ *in the lattice* $\mathbb{Z}^2$ *and let* $S' = \{\mathbf{d}_1 + \mathbf{v}, \mathbf{d}_2 + \mathbf{v}, \ldots, \mathbf{d}_n + \mathbf{v}\}$ *be the set of positions of the dots in the translated array. Then if* $\mathbf{v} \neq \mathbf{0}$*, we have* $|S \cap S'| \leq 1$*.*

*Proof.* Suppose there exists a vector $\mathbf{v}$ and dot positions $\mathbf{d}_i, \mathbf{d}_j, \mathbf{d}_k, \mathbf{d}_l$ such that $\mathbf{d}_i = \mathbf{d}_j + \mathbf{v}$ and $\mathbf{d}_k = \mathbf{d}_l + \mathbf{v}$. Then $\mathbf{d}_i - \mathbf{d}_k = \mathbf{d}_j - \mathbf{d}_l$. As $\mathcal{A}$ is a Costas array, this implies that $\mathbf{d}_i = \mathbf{d}_j$ and $\mathbf{d}_k = \mathbf{d}_l$, and hence $\mathbf{v} = \mathbf{0}$. □

Two main constructions for Costas arrays are known (see [14–16] for further discussion). Let $p$ be an odd prime. An integer $\alpha$ is a *primitive root modulo* $p$ if the powers $\alpha^1, \alpha^2, \ldots, \alpha^{p-1}$ are all distinct modulo $p$; such integers exist for all odd primes $p$.

**The Welch Construction** Let $\alpha$ be a primitive root modulo $p$ and let $\mathcal{A}$ be a $(p-1) \times (p-1)$ array. For $1 \leq i \leq p-1$ and $1 \leq j \leq p-1$ we put a dot in $\mathcal{A}(i,j)$ if and only if $\alpha^i \equiv j \pmod{p}$.

**The Golomb Construction** Let $q$ be a power of a prime and let $\alpha$ and $\beta$ be two primitive elements in GF($q$), i.e. elements that generate the multiplicative group of GF($q$). We define $\mathcal{A}$ to be a $(q-2) \times (q-2)$ array. For $1 \leq i \leq q-2$ and $1 \leq j \leq q-2$ we put a dot in $\mathcal{A}(i,j)$ if and only if $\alpha^i + \beta^j = 1$. We remark that when $\alpha = \beta$ the construction is called the Lempel Construction.

There are several variants for these two constructions resulting in Costas arrays with orders slightly smaller (by 1, 2, 3, or 4) than the orders of these two constructions.

## 4 Construction of Key Predistribution Schemes for Grid-Based Networks

In this section we provide basic definitions relating to key predistribution, and examine certain properties that must be considered when designing such schemes, before proposing constructions of KPSs that are specifically adapted to grid-based networks.

Let $\mathcal{K}$ be a finite set whose elements we refer to as keys (whether they be either actual secret keys, or quantities from which such keys may be derived). We consider a set $U$ of wireless sensors, each of which has sufficient memory to store $m$ keys; after deployment the nodes $U$ form a wireless sensor network $W$.

**Definition 2.** *A* key predistribution scheme (KPS) *for $W$ is a map $U \rightarrow \mathcal{K}^m$ that assigns up to $m$ keys from $\mathcal{K}$ to each node in $U$.*

Each node stores the keys assigned to it in its memory prior to deployment. Once the nodes are deployed we have the following possible situations.

- Two nodes that share one or more common elements of $\mathcal{K}$ can use them to derive a common key.
- Two nodes that do not share a key may rely on an intermediate node with which they both share a key in order to communicate securely; this is referred to as a *two-hop path*.

If each $k \in \mathcal{K}$ is assigned to a set $S_k \subset U$ of at most $\alpha$ nodes we refer to the KPS as an *$[m, \alpha]$-KPS*. As mentioned in Sect. 2, one of the goals when designing an $[m, \alpha]$-KPS is to enable each node to communicate directly with as many nodes as possible, hence we would like to maximise the expected number of neighbouring nodes that share at least one key with a given $\Psi$. We note that when evaluating properties of a grid-based network in which the network does not extend infinitely in all directions, complications may arise due to nodes on the edge of the network having a reduced number of neighbours. This can be avoided by restricting attention to properties of nodes on the *interior* of the network (nodes $\Psi$ such that each grid position that is within range of $\Psi$ contains a node of the network). This is a reasonable restriction to make as it greatly simplifies analysis and comparison of KPSs, especially since for a grid-based network of any size the edge nodes will only represent a small proportion of the network.

**Theorem 1.** *When an $[m, \alpha]$-KPS is used to distribute keys to nodes in a square grid network, the expected number of $r$-neighbours of a node $\psi$ in the interior of the network that share at least one key with $\Psi$ is at most $m(\alpha - 1)$. The value $m(\alpha - 1)$ is achieved precisely when the following conditions are met.*

1. *Each interior node stores exactly $m$ keys, each of which are shared by exactly $\alpha$ nodes.*
2. *No pair of nodes shares two or more keys.*
3. *The distance between any two nodes sharing a key is at most $r$.*

*Proof.* The maximum number of keys allocated to an interior node $\Psi$ by an $[m,\alpha]$-KPS is $m$; each of these keys is shared by at most $\alpha$ nodes (which may or may not be $r$-neighbours of $\Psi$). Hence a given interior node shares keys with at most $\alpha - 1$ of its $r$-neighbours, and this maximum value is achieved if and only if no two nodes share more than one key with $\Psi$, and every node with which $\Psi$ shares a key is an $r$-neighbour of $\Psi$. The result follows directly. □

This result indicates that when distributing keys according to an $[m,\alpha]$-KPS, limiting the number of keys shared by each pair of nodes to at most one increases the number of pairs of neighbouring nodes that share keys, hence this is desirable from the point of view of efficiency. This restriction will be further exploited in the analysis of Sect. 5. In the following section we describe a method of constructing $[m,\alpha]$-KPSs with this property.

### 4.1 Key Predistribution Using Costas Arrays

We now propose a KPS for a grid-based network, in which the pattern of nodes that share a particular key is determined by a Costas array. The result is a $[n,n]$-KPS in which any two nodes have at most one key in common.

**Construction 2.** *Let $\mathcal{A}$ be a $n \times n$ Costas array. We can use $\mathcal{A}$ to distribute keys from a keypool $\mathcal{K}$ to a set $U$ of nodes arranged in a grid-based network as follows.*

- *Arbitrarily choose one square of the grid to be the origin, and superimpose $\mathcal{A}$ on the grid, with its lower left-hand square over the origin. Select a key $k_{00}$ from $\mathcal{K}$, and distribute it to nodes occurring in squares coinciding with a dot of $\mathcal{A}$ (so $n$ nodes receive the key $k_{00}$).*
- *Similarly, for each square occurring at a position $(i,j)$ in the grid, we place the lower left-hand square of $\mathcal{A}$ over that square, then assign a key $k_{ij} \in \mathcal{K}$ to the squares that are now covered by dots of $\mathcal{A}$.*

If the dots of the Costas array occur in squares $(0,a_0)$, $(1,a_1),\ldots,(n-1,a_{n-1})$ of the array then the above scheme associates a key $k_{ij}$ with the nodes in squares $(i,j+a_0)$, $(i+1,j+a_1)$, $\ldots$, $(i+n-1,j+a_{n-1})$ (where such nodes exist). We observe that the deterministic nature of this key allocation, together with the structured topology of a square grid, means that nodes can simply store the coordinates in the grid of those nodes with which they share keys, thus obviating the need for a shared-key discovery process with ensuing communication overheads.

*Example 2.* Consider the $3 \times 3$ Costas array of Example 1. If we use this array for key distribution as described above, each node stores three keys. Figure 1 illustrates this key distribution: each square in the grid represents a node, and each symbol contained in a square represents a key possessed by that node. The central square stores keys marked by the letters $A$, $B$ and $C$; two further nodes share each of these keys, which are marked in bold. Letters in standard type represent keys used to connect the central node to one of its neighbours via a two-hop path, other keys are marked in grey. Note that we have only illustrated some of the keys; the pattern of key sharing extends in a similar manner throughout the entire network.

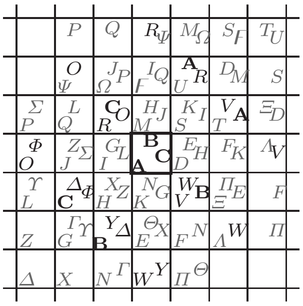

P Q R Ψ M Ω S F T U
O Ψ J P Ω I Q F A R U D M S
Σ P L Q C O R H J M K I S V A T Ξ D
Φ O Z Σ J G L I B C A E H D F K Λ V
ϒ L Δ Φ C X Z H N G K W B V Π E Ξ F
Z Γ ϒ G Y Δ B Θ X E F N Λ W Π
Δ X N Γ W Y Π Θ

**Fig. 1.** Key distribution using a 3 × 3 Costas array.

**Theorem 3.** *The key predistribution scheme in Construction* 2 *has the following properties:*

1. *Each sensor is assigned $n$ different keys.*
2. *Each key is assigned to $n$ sensors.*
3. *Any two sensors have at most one key in common.*
4. *The distance between two sensors which have a common key is at most $\sqrt{2}(n-1)$.*

*Proof.*

1. There are $n$ dots in $\mathcal{A}$. For each dot in turn, if we position $\mathcal{A}$ so that dot lies over a given node $\Psi$, this determines a positioning of $\mathcal{A}$ for which the corresponding key is allocated to $\Psi$. Hence $\Psi$ stores $n$ keys in total.
2. A key $k_{ij}$ is assigned to $n$ positions in the square grid, namely those that coincide with the $n$ dots of a fixed shift of $\mathcal{A}$.
3. Suppose there exist two sensors $A$ and $B$ sharing (at least) two keys. These keys correspond to different translations of the array $\mathcal{A}$, hence there exist two translations of $\mathcal{A}$ in which dots occur at the positions of both $A$ and $B$. However, by Lemma 1, two copies of $\mathcal{A}$ coincide in at most one dot, thus contradicting the original assumption.
4. The two most distant sensors which have a key in common must correspond to two dots in the same translation of $\mathcal{A}$. The largest distance between two dots in $\mathcal{A}$ occurs if they are in two opposite corners of the array, i.e. at distance $\sqrt{2}(n-1)$.

□

**Corollary 1.** *When the $[n,n]$-KPS of Construction 2 is applied to a grid-based network then a node on the interior of the network shares keys with $n(n-1)$ other nodes, the maximum possible for a $[n,n]$-KPS.*

### 4.2 Distinct-Difference Configurations in Key Predistribution

The proof of Part 3 of Theorem 3 relies on the property that the vectors connecting pairs of dots in a Costas array are pairwise distinct. We do not, however, make use of the requirement that each row and column have exactly one dot. This suggests that we can relax this condition in order to explore other structures for use in key predistribution. This leads us to the following definition.

**Definition 3.** *A* distinct-difference configuration *$DD(m, r)$ consists of a set of $m$ dots placed in a square grid such that*

- *any two of the dots in the configuration are at distance at most $r$ apart,*
- *all $m(m-1)$ differences between pairs of dots are distinct as vectors (any two vectors differ either in length or direction).*

A Costas array is an example of a DD$(n, r)$, for some $r \leq \sqrt{2}(n-1)$. Like Costas arrays, a DD$(m, r)$ can be used for key predistribution:

**Construction 4.** *For a given $DD(m, r)$ we distribute keys as in Construction 2, using the $DD(m, r)$ in place of a Costas array.*

**Theorem 5.** *If a $DD(m, r)$ is used for key predistribution as described in Construction 4 the resulting KPS has the following properties:*

1. *Each sensor is assigned $m$ different keys.*
2. *Each key is assigned to $m$ sensors.*
3. *Any two sensors have at most one key in common.*
4. *The distance between two sensors which have a common key is at most $r$.*

*Proof.* As in the case of the Costas arrays, the fact that differences between pairs of dots are distinct imply that two nodes share at most one key. The limit on the distance between nodes sharing keys are a distance of at most $r$ apart follows directly from the restriction on the distances between dots in the DD$(m, r)$. □

*Example 3.*

This is an example of a DD$(3, 2)$. If used in a KPS each node stores 3 keys. Figure 2 illustrates (part of) the pattern of key sharing that results. As in Fig. 1, each square in the grid represents a node, and each letter represents a key possessed by that node. This key distribution has an advantage over that of Example *2* in that each node still shares keys with six other nodes, but these nodes are all 2-neighbours, rather than 3-neighbours.

This construction provides $[m, m]$-KPSs in which interior nodes share keys with an optimal number $m(m-1)$ of neighbouring nodes. We have greater flexibility than Construction 3.5 because we consider a more general class of configurations. So we are better able to choose a configuration whose properties match the application requirements. The use of a DD$(m, r)$ enables the construction of

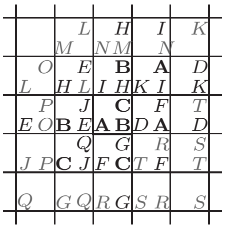


**Fig. 2.** Key predistribution using a DD(3, 2)

a KPS suitable for the specific radius $r$ and maximum storage $m$ of a given network[3], whereas in the case of Costas arrays the number of dots and the maximal distance between them are directly linked.

We have noted that the use of a DD$(m, r)$ maximises the number of $r$-neighbours that share keys with a given node. Additionally, it is desirable to maximise the number of $r$-neighbours that can communicate securely with a given node $\Psi$ via a one-hop or two-hop path. We refer to this quantity as the *two-hop $r$-coverage* of a KPS. In the case of our scheme based on distinct-difference configurations we refer to the two-hop $r$-coverage of a DD$(m, r)$ to indicate the two-hop $r$-coverage obtained by a KPS constructed from that configuration. Table 1 shows the maximum possible values for the two-hop $r$-coverage of a DD$(m, r)$ for $r = 1, 2, \ldots, 12$. The empty positions in the table represent combinations of $m$ and $r$ for which no DD$(m, r)$ exists. In Fig. 3 we illustrate DD$(m, r)$ achieving the maximal two-hop $r$-coverage values shown in Table 1, for those cases where the corresponding two-hop $r$-coverage cannot be obtained

**Table 1.** The maximum two-hop $r$-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 | ≥432 |
| 9 | - | - | - | - | - | - | 148 | 196 | 252 | 316 | 376 | 440 |
| 10 | - | - | - | - | - | - | - | 196 | 252 | 316 | 376 | 440 |
| 11 | - | - | - | - | - | - | - | - | 252 | 316 | 376 | 440 |
| 12 | - | - | - | - | - | - | - | - | - | 316 | 376 | 440 |

[3] provided a suitable DD$(m, r)$ can be found. For a given $r$ there is evidently an upper limit on the value of $m$ for which a DD$(m, r)$ exists. If the potential storage $m$ exceeds this value a DD$(m', r)$ could be employed with $m'$ equal to the maximum number of dots possible in a distance $r$ distinct-difference configuration.

by a configuration with smaller $m$ (without increasing $r$) or smaller $r$ (without increasing $m$). (For a given radius $r$ the number of two-hop $r$-neighbours is evidently bounded by the total number of $r$-neighbours; these totals correspond to the numbers in bold in Fig. 3. Similarly, for a given $m$ there is a maximum number of two-hop $r$-neighbours that can be achieved by a $\mathrm{DD}(m,r)$; these values appear in italics. Both trends are apparent in Table 1.) In the case of $m = 8, r = 12$ the best known two-hop $r$-coverage is 432.

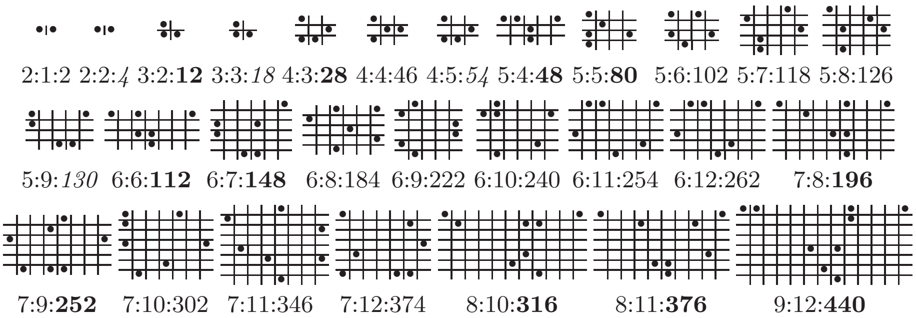


**Fig. 3.** Distinct-difference configurations with maximal two-hop $r$-coverage $\gamma$. The labels indicate the corresponding $m$:$r$:$\gamma$. The value of $\gamma$ is given in bold if it is the maximum possible for the given $r$, and in italics if it is the maximum given $m$.

## 5 Evaluation of Key Predistribution Schemes for Grid-Based Networks

In Sect. 4 we indicated some desirable properties of key predistribution schemes in order to motivate our constructions. We now provide a wider analysis of the properties of these schemes. There are no standard metrics for evaluating KPSs, as desirable properties depend on the particular application environment; authors tend to devise their own metrics for evaluating the schemes they propose. Nevertheless the basic goals of these schemes remain the same: it is beneficial to restrict the amount of storage and communication overheads required, while maximising the number of secure communication links between nodes, even in the case when nodes are subject to adversarial compromise. In this section we consider each of these aspects, in the context of grid-based networks, and define the precise quantities we use in Sect. 6 to compare our schemes with previous schemes.

**Storage** There is no established consensus on the number of symmetric keys that a sensor can feasibly store in practice. Estimations in the literature range from “perhaps 30-50” [23] to more than 200 [6]. As sensor technology improves, the amount of memory available is increasing. However, there is

**Table 2.** A comparison of key predistribution schemes for a 100 × 100 grid-based network. (*Entries represent the mean over 10000 trials, with the sample standard deviation given in brackets.*)

| **Scheme** | **m** | $\alpha$ | **One-hop** | **Two-hop** | **Resilience** | **L. Resilience** |
|---|---|---|---|---|---|---|
| Costas | 8 | 8 | 56 | 366 | 331 (86) | 59 (53) |
| DD(8, 11) | 8 | 8 | 56 | 376 | 336 (86) | 59 (53) |
| Liu & Ning | 8 | 2 | 8 | 24 | 23.87 (1.48) | 20.3 (7.0) |
| Eschenauer & Gligor | 8 | $\approx 200$ | 56.2 (7.0) | 370.0 (3.8) | 36 (38) | 36 (38) |
| Ito *et al.* | 8 | $\approx 8$ | 36.2 (6.4) | 319.6 (20.1) | 259 (97) | 52 (47) |

always a tradeoff between the amount of memory used for cryptographic purposes and the amount available for the rest of the sensor's functionality. Also, the development of smaller, less power-hungry sensors will continue to place limits on memory capacity in the future. It is common for the storage requirement to be a parameter of a KPS, and for other properties to be described in terms of this parameter. When choosing parameters for the schemes we compare in Sect. 6, we fix an upper bound for the storage and consider only schemes whose storage requirements do not exceed this bound.

**Cost of shared key discovery** The deterministic nature of our scheme means that no communication is required either for neighbour discovery, or for shared key discovery.

**One-hop and two-hop coverage** As discussed in Sect. 4, our schemes ensure nodes have the maximum number $m(m-1)$ of one-hop $r$-neighbours that is possible for a $[m, m]$-KPS. Thus the number of secure communication links is maximised by choosing $m$ to be as large as possible. Note that there are two factors constraining the size of $m$: the memory capacity of nodes, and the combinatorial limits on the size of $m$ for a fixed value of $r$. In order to assess the connectivity of a scheme, it is also desirable to take into account the two-hop $r$-coverage. Table 1 illustrates that if the storage $m$ is sufficient, it is possible to find distinct difference configurations for use in Construction 4 that ensure that every node on the interior of the network can communicate with each of its $r$-neighbours by either a one-hop or a two-hop path.

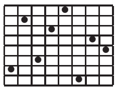
**Resilience** Informally speaking, the resilience of a KPS is the extent to which secure communication can be maintained within the network when an adversary compromises a certain number of nodes and extracts the keys that they store. In Sect. 6 we will measure the resilience of a scheme by the expected number of $r$-neighbours of a node $\Psi$ that can still communicate securely (i.e. by using keys unknown to the adversary) with $\Psi$ by either a one-hop or two-hop path, after a fixed number of nodes have been compromised. We will consider both the case in which the compromised nodes are chosen uniformly throughout the network, and the case where the nodes are drawn uniformly from the $r$-neighbourhood of $\Psi$ (we assume that $\Psi$ itself is not compromised.) We refer to the quantity arising from the latter case as the *local resilience*.

## 6 Concrete Comparison of Existing Schemes

In order to illustrate the performance of the KPSs proposed in this paper we select some concrete values for the network parameters, which allows us to compare the performance of our schemes explicitly with other schemes appearing in the literature. Our schemes are shown to perform better than previously known schemes in our network model. We will consider a grid-based network with 10000 nodes arranged in a square, in which each node can store up to 8 keys and has a communication range $r = 11$. The results of our analysis of several schemes are summarised in Table 2. For each scheme we are interested in the values of $m$, $\alpha$ and the expected number of one-hop 11-neighbours *(One-hop)* and two-hop 11-neighbours *(Two-hop)*. We also measure the number of a node's two-hop links that remain secure after an adversary compromises five nodes, either uniformly throughout the network *(Resilience)*, or uniformly from among that node's 11-neighbours *(L. Resilience)*. These values for each scheme are displayed in Table 2, and represent the mean value over 10000 randomly generated instances. The corresponding sample standard deviation is given in brackets. In each case the parameters for the schemes have been chosen so that the storage requirement is at most 8 keys, and so that all schemes have (where possible) a similar number of one-hop 11-neighbours. We now give a brief description of the schemes we are considering, as well as an explanation of the parameter choices involved.

**Construction 4** The 11-neighbourhood of a node contains 376 other nodes. If the storage limit is 8, then Construction 4 results in a KPS in which each node has 56 one-hop neighbours. Using the DD$(8, 11)$ shown in Fig. 3 means that all 376 11-neighbours of a given node can communicate with that node via a one-hop or two-hop path.

**Construction 2** This construction also results in nodes having 56 one-hop neighbours, however the best two-hop 11-coverage that results from an $8 \times 8$ Costas array is 366, achieved by the following array.



**Eschenauer and Gligor [13]** In Eschenauer and Gligor's KPS, each node is assigned $m$ keys drawn uniformly without replacement from a key pool of a fixed size. By taking $m = 8$ and a keypool of size 400 for this network we obtain a KPS in which the number of one and two-hop 11-neighbours is similar to that of our schemes.

**Liu and Ning [25]** Liu and Ning's 'closest pairwise scheme' is a location-based scheme in which each node shares keys with its $c$ closest neighbours. Since we are working with a square grid, we can consider a scheme in which each node shares pairwise keys with the 8 nodes surrounding it.

**Ito, Ohta, Matsuda and Yoneda [20]** The scheme of Ito *et al.* is a location-based, probabilistic scheme. They propose associating keys with points in the target area, then for each node they randomly choose $m$ points that are expected to lie within its communication range after deployment, and assign the corresponding keys to that node. To deploy this scheme in our grid-based network we associate a key with each grid point, then for each node randomly choose 8 points within distance 11 of that node.

**Other location based schemes** Most of the location-based KPSs in the literature do not assume a precise knowledge of sensor locations, but instead divide the target area into regions (square, rectangular, hexagonal and triangular regions have all been proposed) and suppose that the region in which each sensor will be deployed is known *a priori*. Schemes such as those in [9, 10, 17, 34, 25] involve all nodes in each region being given shares in a threshold key establishment scheme such as those of [1, 2] with nodes receiving shares corresponding to each of the neighbouring regions. The storage constraints of the specific network environment we are considering mean that most of these scheme either cannot be employed, or else could only be employed with such low thresholds as to severely compromised their resilience.

The scheme of Du, Deng, Han, Chen and Varshney [11] similarly divides the target area into regions, and then modifies Eschenauer and Gligor's basic scheme by letting the pool from which nodes draw keys depend on the region in which they are to be deployed. However, Ito *et al.* argue that this does not provide sufficient granularity [20], as a rectangular region does not adequately model the circle throughout which a node is supposed to be able to communicate.

In Table 2 we compare our Costas array and DD$(8, 11)$ schemes, Liu and Ning's closest pairwise scheme, Eschenauer and Gligor's scheme, and the scheme of Ito *et al.* for the choices of parameters discussed above. This data highlights several differences in the behaviour of the various schemes in this environment; in particular we note the following.

The local resilience of Eschenauer and Gligor's scheme is less than that of our schemes, and the resilience is substantially less (as their scheme does not take account of the nodes' locations, the resilience matches the local resilience). This is essentially due to the large value of $\alpha$ that is required in order for their scheme to give adequate one-hop or two-hop coverage. The use of location knowledge in the scheme of Ito *et al.* results in an improvement in resilience, although it is still significantly less than that of our schemes, and the one-hop and two-hop coverage is lower too. A change of parameters could increase the coverage, but at the cost of increasing $\alpha$, so that any increase in resilience would be curtailed. Furthermore, even though [20] is location based, the fact that its key distribution is probabilistic means that it incurs the same shared-key-discovery cost as [13], whereas our deterministic schemes involve no key-discovery overheads.

The coverage of Liu and Ning's scheme is very low. The resilience is high in proportion to the coverage, in that most of the links are expected to remain unaffected after node compromise. However since the number of links existing prior to node compromise is small, then in absolute terms the resilience and local resilience are even lower than that of [13].

Thus we see that both Construction 2 and Construction 4 yield KPSs that provide good one-hop and two-hop coverage in grid-based networks with restricted storage, and that the resulting KPSs are demonstrably more resilient in the fact of node compromise than previously proposed schemes. They therefore represent a good solution whenever a very lightweight yet resilient KPS is required for a grid-based network.

## References

1. Blom, R.: An Optimal Class of Symmetric Key Generation Systems. In Beth, T., Cot, N., Ingemarsson, I., (eds.) Eurocrypt 1984. LNCS, vol. 209, pp. 334–338. Springer, Heidelberg (1985)
2. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-Secure Key Distribution for Dynamic Conferences. In: Brickell, E.F. (ed.) Crypto 1992. LNCS, vol. 740, pp. 471–338. Springer, Heidelberg (1992)
3. Çamtepe, S.A., Yener, B.: Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks. In: Samarati, P., Ryan, P.Y.A., Gollman, D., Molva, R. (eds.) Esorics 2004. LNCS, vol. 3193, pp. 293–308. Springer, Heidelberg (2004)
4. Çamtepe, S.A., Yener, B., Yung, M.: Expander Graph Based Key Distribution Mechanisms in Wireless Sensor Networks. In: IEEE International Conference on Communications, vol. 5., pp. 2262–2267. IEEE press, New York (2006)
5. Chakrabarti, D., Maitra, S., Roy, B.K.: A Hybrid Design of Key Pre-distribution Scheme for Wireless Sensor Networks. In: ICISS. pp. 228–238 (2005)
6. Chakrabarti, D., Maitra, S., Roy, B.K.: A Key Pre-distribution Scheme for Wireless Sensor Networks: Merging Blocks in Combinatorial Design. In: ISC. pp. 89–103 (2005)
7. Chan, H., Perrig, A., Song, D.: Random Key Predistribution Schemes for Sensor Networks. In: IEEE Symposium on Security and Privacy, pp. 197- . IEEE press, New York (2003)
8. Chan, S.P., Poovendran, R., Sun, M.T.: A Key Management Scheme in Distributed Sensor Networks Using Attack Probabilities. In: IEEE GLOBECOM '05, vol. 2. (2005)
9. Delgosha, F., Fekri, F.: Key Pre-distribution in Wireless Sensor Networks Using Multivariate Polynomials. In: IEEE Commun. Soc. Conf. Sensor and Ad Hoc Commun. and Networks - SECON05. (2005)
10. Delgosha, F., Fekri, F.: Threshold Key-Establishment in Distributed Sensor Networks Using a Multivariate Scheme. In: Infocom 2006. (2006)
11. Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K.: A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. In: INFOCOM. (2004)
12. Du, W., Deng, J., Han, Y.S., Varshney, P.K.: A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) CCS '03. pp. 42–51. ACM Press, New York (2003)
13. Eschenauer, L., Gligor, V.D.: A Key-Management Scheme for Distributed Sensor Networks. In: Atluri, V. (ed.) CCS 2002. pp. 41–47. ACM Press, New York (2002)
14. Golomb, S.W.: Algebraic Constructions for Costas Arrays. J. Comb. Theory A. 37, 13–21 (1984)
15. Golomb, S.W., Taylor, H.: Constructions and Properties of Costas Arrays. P. IEEE. 72, 1143–1163 (1984)

16. Golomb, S.W., Taylor, H.: Two-Dimensional Synchronization Patterns for Minimum Ambiguity. IEEE T. Inform. Theory. 28, 600–604 (1982)
17. Huang, D., Mehta, M., Medhi, D., Harn, L.: Location-Aware Key Management Scheme for Wireless Sensor Networks. In: Setia, S., Swarup, V. (eds.) SASN 2004. pp.29–42. ACM Press, New York (2004)
18. Hwang, D., Lai, B.C., Verbauwhede, I.: Energy-Memory-Security Tradeoffs in Distributed Sensor Networks. In: ADHOC-NOW. pp. 70–81 (2004)
19. Hwang, J., Kim, Y.: Revisiting Random Key Pre-distribution Schemes for Wireless Sensor Networks. In: Setia, S., Swarup, V. (eds.) SASN 2004. pp.43–52. ACM Press, New York (2004)
20. Ito, T., Ohta, H., Matsuda, N., Yoneda, T.: A Key Pre-distribution Scheme for Secure Sensor Networks Using Probability Density Function of Node Deployment. In: Atluri, V., Ning, P., Du, W. (eds.) SASN 2005. pp.69–75. ACM Press, New York (2005)
21. Lee, J., Stinson, D.R.: A Combinatorial Approach to Key Predistribution for Distributed Sensor Networks. IEEE Wireless Communications and Networking Conference, CD-ROM, 2005, paper PHY53-06, 6 pp (2005)
22. Lee, J., Stinson, D.R.: Deterministic Key Predistribution Schemes for Distributed Sensor Networks. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 294–307. (2004)
23. Lee, J., Stinson, D.R.: On the Construction of Practical Key Predistribution Schemes for Distributed Sensor Networks Using Combinatorial Designs. ACM Trans. Inf. Syst. Secur. 11(2), 1–35 (2008)
24. Liu, D., Ning, P.: Establishing Pairwise Keys in Distributed Sensor Networks. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) CCS '03. pp. 52–61. ACM Press, New York (2003)
25. Liu, D., Ning, P.: Location-Based Pairwise Key Establishments for Static Sensor Networks. In: Setia, S., Swarup, V. (eds.) SASN 2003. pp.72–82. ACM Press, New York (2003)
26. Liu, D., Ning, P., Li, R.: Establishing Pairwise Keys in Distributed Sensor Networks. ACM Trans. Inf. Syst. Secur. 8(1), 41–77 (2005)
27. Martin, K.M., Paterson, M.B.: An Application-Oriented Framework for Wireless Sensor Network Key Establishment. In: WCAN 2007. ENTCS, (to appear)
28. Mohaisen, A., Maeng, Y., Nyang, D.: On Grid-Based Key Pre-distribution: Toward a Better Connectivity in Wireless Sensor Network. In: SSDU-07. (2007)
29. Mohaisen, A., Nyang, D.: Hierarchical Grid-Based Pairwise Key Predistribution Scheme for Wireless Sensor Networks. In: EWSN. 83–98 (2006)
30. Pietro, R.D., Mancini, L.V., Mei, A.: Random Key-Assignment for Secure Wireless Sensor Networks. In: Setia, S., Swarup, V. (eds.) SASN 2003. pp.62–71. ACM Press, New York (2003)
31. Ramkumar, M., Memon, N.: An Efficient Key Predistribution Scheme for Ad Hoc Network Security. IEEE J. Sel. Area. Comm. 23, 611-621 (2005)
32. Rickard, S.: CostasArrays.org, www.costasarrays.org
33. Römer, K., Mattern, F.: The Design Space of Wireless Sensor Networks. Wirel. Commun. 11(6), 54–61. (2004)
34. Zhou, Y., Zhang, Y., Fang, Y.: Key Establishment in Sensor Networks Based on Triangle Grid Deployment Model. In: MILCOM 2005, vol. 3, pp. 1450–1455 (2005)
35. Zhu, S., Xu, S., Setia, S., Jajodia, S.: Establishing Pairwise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach. In: ICNP. 326–335 (2003)