Efficient Implementation of Lightweight Key Predistribution Techniques for Grid-Based Wireless Sensor Networks

Abstract

The key predistribution scheme (KPS) of Blackburn *et al.* for grid-based wireless sensor networks makes use of distinct-difference configurations (DDCs) to achieve a lightweight and resilient distribution of symmetric keys. Significant theoretical progress on DDCs has been made recently. This paper examines the implications of this research for the construction of KPSs for grid-based wireless sensor networks. We explore the connectivity and resilience requirements of such schemes and give explicit algorithms for efficiently constructing DDCs that lead to schemes with the desired properties for a large range of parameters.

Keywords: Key predistribution, wireless sensor networks, symmetric key management, distinct-difference configurations

1 Introduction

A wireless sensor network (WSN) is an ad hoc network formed from a large collection of low-powered sensor nodes that gather data and use wireless communication to transmit the information they collect. Due to the wireless nature of the communication and the potential commercial sensitivity of the data they measure, there is a requirement for cryptographic techniques to provide authentication, data integrity and/or confidentiality. The limited processing power and memory of the sensors means that in many circumstances the use of symmetric cryptographic primitives may be preferred to more computationally intensive public-key operations[1]. This creates a requirement for the sensors to share keys. One effective method of distributing keys to the sensors is a *key predistribution scheme* (KPS), which allocates keys to be stored in the sensors' memories prior to deployment. The design of a KPS involves a trade-off between the number of keys each node must store *(storage)*, the number of secure links between nodes in the resulting network *(connectivity)*, and the vulnerability of the scheme to adversaries that capture nodes and extract the keys they contain *(resilience)*. Many KPSs have been proposed in the literature, but most of them have been designed for networks in which the location of the sensors is not known before deployment (see [9, 14, 18] for surveys of this field). However, in many instances the demands of the application lead directly to networks in which there is prior knowledge of sensor locations [14, 17]. When this occurs, this location knowledge may be exploited for the development of KPSs that provide a more efficient trade-off between storage, connectivity and resilience.

One natural scenario in which there is complete knowledge of the sensors' locations is that in which the sensors are located in a grid formation. The use of a grid-based network is generally motivated by applications that require measurements to be taken at regularly-spaced intervals. In some cases, the use of a hexagonal grid (as opposed to a square grid) may be desirable, as it permits a particularly efficient packing of sensors into a target region. Grid-based networks can arise in many applications, with recent instantiations including soil moisture sensing [2], monitoring conditions in a nectarine orchard [1], and measuring the efficiency of water use during irrigation [15]).

Blackburn *et al.* showed that effective KPSs for networks based on square grids can be achieved through the use of combinatorial objects known as *distinct-difference configurations* (DDCs): sets of points in a square grid such that the vectors joining any two pairs of points differ in either length or direction [4]. This scheme makes use of the knowledge of the nodes'

[1] Recent progress in efficient implementation of public-key techniques indicate it is possible for a sensor node to perform the necessary computations. However, such techniques still require a substantial amount of memory to store the necessary code, and the operations are costly in terms of the time and energy required to perform them. Additionally, the practical advantages of public-key schemes are less substantial in the case of a grid-based network where the location knowledge can be exploited to achieve particularly effective key predistribution.

locations to ensure that keys are only shared by nodes that are within communication range. For a network with a given communication range and distance between neighbouring nodes, this permits the total number of nodes in the network to be made arbitrarily large without affecting either connectivity or resilience. This is not the case for KPSs such as [10] that do not exploit location knowledge: in such schemes, the storage/connectivity/resilience trade-off becomes worse as the number of nodes is increased, as maintaining connectivity with a given amount of storage inevitably leads to decreased resilience as the network size increases. The trade-off provided by the scheme of [4] is particularly efficient, as it ensures that any two nodes share at most one key, thus maximising the number of communication links secured by a given number of shared keys. Additionally, the fact that this scheme is deterministic implies that no communication is required for pairs of nodes to determine which keys they share (the shared-key discovery process represents a substantial overhead for networks in which the keys are allocated randomly).

A suitable choice of DDC in the scheme of Blackburn *et al.* leads to KPSs that perform favourably compared with other schemes in the literature, as demonstrated in [4]. However, in [4] a computer search was required in order to find DDCs with good properties for use in key predistribution; this quickly becomes infeasible if large DDCs are required.

Certain combinatorial properties of DDCs have recently been explored in [3,5]. The authors give upper bounds on the number of points possible in a configuration with bounded radius, and provide constructions of periodic configurations that can be used to obtain DDCs whose number of points is close to optimal (asymptotically) in both the square and hexagonal grid [5]. Inspired by the scheme of [4] they consider bounds on the possible *two-hop coverage* of a DDC, and give constructions of DDCs that attain the maximum possible two-hop coverage given the number of points they contain, as well as others that have provably complete two-hop connectivity over a specific region [3]. However, the papers [3, 5] focus primarily on the combinatorics of DDCs, and the specific implications of their results for the design of practical KPSs are not explored.

In order to design explicit KPSs using these techniques for a given network environment, two further questions must be addressed:

1. What is the best way to choose a DDC on which to base a grid-based KPS?
2. How should the chosen DDC be instantiated by an explicit construction?

We extend the previous research by providing answers to these questions.

1.1 Our Contributions

The previous work on DDCs in the literature is very mathematical, and does not directly consider practicalities that are of key importance in the design of an effective KPS [3, 5]. As we shall see, the selection of appropriate parameters turns out to be delicate, involving a three-way trade-off between storage, connectivity and resilience. This paper addresses these issues in order to provide practical key distribution solutions for a grid based environment.

- We provide an explicit algorithmic description of the the scheme presented in [4]. We extend this scheme to hexagonal grids, and we indicate that it can be combined with Blom's scheme to increase its resilience.
- We give a further analysis of the connectivity and resilience requirements of KPSs for a grid-based environment, and we discuss how this influences the selection of appropriate DDCs for use in the scheme of [4].
- We discuss how the bounds given in [3, 5] affect the possible ranges of parameters of DDCs for use in such schemes.
- We demonstrate how various constructions from [3, 5] can be combined to yield DDCs with the properties we desire for use in the scheme of [4]. We give explicit algorithms describing how these DDCs can be efficiently constructed.

After preliminaries in Section 2, we provide an algorithmic description of the KPS proposed in [4] in Section 3, extending this KPS to suit networks based on hexagonal grids, as well as square grids. In Section 4 we observe that connectivity provides a good criterion for choosing a DDC for use in a KPS, and in Section 5 we discuss connectivity properties that can be used for this purpose, namely the one-hop coverage and two-hop coverage. Efficient constructions for DDCs with good one-hop coverage in both the square and hexagonal grids are given in Section 5.1. Finally, in Section 5.2 we provide an algorithm for the construction of a DDC that gives complete 2-hop coverage over specified regions in the square or hexagonal grid.

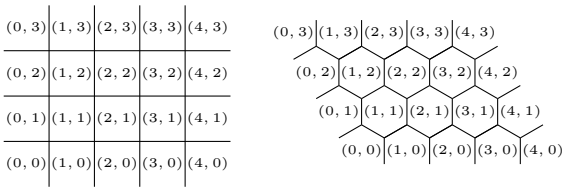


Figure 1: Coordinates for nodes in the square and hexagonal grids

2 Grid-Based Networks and Key Predistribution

2.1 Assumptions

In this paper we consider sensor networks in which the nodes are located at the centres of the squares in a square grid, or the hexagons in a hexagonal grid. Individual nodes in the network may be identified through the use of coordinates, which we assign as shown in Fig. 1.

We suppose that any node is able to communicate with all nodes that lie within distance r of it, and we refer to r as the *communication range* of the nodes. Two nodes are considered to be able to communicate securely if they are within range of each other and share a key; we refer to this as a *one-hop path* between the nodes. Nodes that do not share a key may be able to communicate with the aid of an intermediate node with which they can both form one-hop paths: this is referred to as a *two-hop path*.

The goal of a KPS is to facilitate secure communication between neighbouring nodes. As communication between nodes is costly in terms of the energy expenditure that is required, it is desirable for nearby nodes to be able to communicate as directly as possible. Useful parameters for measuring the performance of a KPS are the *one-hop coverage*, which we define to be the expected number of neighbours that share keys with a node, and the *two-hop coverage*, which we define to be the expected number of neighbours with which a node can communicate securely via either a one-hop or a two-hop path.

Finally, we assume there is an upper bound m on the number of keys each node can feasibly store.

2.2 Adversary Model

We assume the presence of an adversary that can eavesdrop on all unencrypted traffic in the network. In addition, we suppose the adversary has the ability to physically compromise nodes and extract any keys that they store.

2.3 Design Requirements

Ideally we would like a KPS to provide good connectivity with strong resilience, without requiring nodes to store too many keys. As these properties are in opposition to each other, the design of a KPS involves finding an appropriate trade-off between them. Certain trivial schemes may seem obvious candidates for KPSs in grid based networks. However, they have inherent limitations that affect their applicability:

single key scheme Perhaps the simplest KPS is that in which a single key is stored by all nodes in the nextwork. This provides perfect connectivity with extremely low storage overheads. Unfortunately, it has very poor resilience, since the capture of even a single node by the adversary leads to the compromise of all communication links within the network.

immediate neighbours scheme A second possibility would be for a node in a square (hexagonal) grid to share keys with its four (six) closest neighbours. This leads to very low storage and ensures the network is connected. However, the one-hop coverage is just four (six), and the two-hop coverage is only twelve (eighteen), which can lead to communication bottlenecks in the network, and would result in nodes becoming isolated from the rest of the network if their immediate neighbours were to fail or be compromised.

locally-complete pairwise scheme The fragility of the previous scheme could be overcome by allowing each pair of nodes that are within communication range to store a distinct key. This scheme has excellent one-hop coverage (since any pair of nodes that is within range can communicate securely) and resilience (since the compromise of a node does not affect the security of any keys shared by uncompromised nodes). However, the number of nodes within range of any given node grows quadratically with the communication range, which quickly results in nodes being required to devote unfeasible amounts of memory to the storage of keys.

The inflexibility of these schemes makes it impossible to vary the trade-off between storage, connectivity

and resilience to suit application requirements. In Section 3 we describe an example of a scheme designed specifically for a grid-based environment that can provide a flexible and efficient trade-off between these three properties.

3 A Practical KPS for Grid-Based Networks

In [4], Blackburn *et al.* describe a KPS for a grid-based network in which any pair of nodes shares at most one key, and only nodes that are within communication range share keys. This first property leads to schemes with high connectivity given a particular amount of storage: since there is no duplication of shared keys, each shared key secures a new link. The second property ensures that no shared key is wasted on a pair of nodes who are too far apart to be able to communicate, and also has the advantage that if an adversary extracts keys from a node then the only links affected will be local, with the rest of the network remaining unaffected. These properties are achieved by basing the scheme on a distinct-difference configuration DD(m, r): a set of points in the square grid such that the difference between any two points is at most r, and the vectors connecting any two pairs of points differ either in length or direction. It is demonstrated in [4] that for networks based on square grids, this scheme outperforms other schemes from the literature [10,11,13], achieving good two-hop coverage and resilience with comparatively low storage requirements.

Algorithm 1 is a description of the scheme of [4], which works by selecting a DD(m, r), then allocating the keys so that the pattern of nodes in the grid that share any given key coincides with the pattern of points in the DD(m, r). We assume the nodes of the network lie in a rectangle of size $l_1 \times l_2$ (if this is not the case, l_1 and l_2 can be taken to be the dimensions of a rectangle in the square grid that contains all the nodes of the network). Each node is identified by the pair of integers (i, j), representing the column and row of the rectangle in which it is located (see Fig. 1). The distinct-difference configuration is represented by a set $DDC := \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^m\}$, where $\mathbf{P}^i = (P_0^i, P_1^i)$ represents the coordinates of the i^{th} dot in the configuration, and we assume $0 \leq P_0^i \leq r$ and $0 \leq P_1^i \leq r$ for $i = 1, 2, \ldots, m$. The keyrings of the nodes are represented as sets of key identifiers, integers that each represent a specific key that is drawn pseudorandomly from a larger keypool. Note that while the distribution of key identifiers is entirely deterministic, the correspondence between the identifiers and specific keys is necessarily probabilistic.

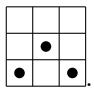
Algorithm 1: Grid-based KPS

Input: a distinct-difference configuration $DDC := \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^m\} \subset [0, r] \times [0, r]$, positive integers l_1, l_2 representing the dimensions of the target rectangle

Output: an $l_1 \times l_2$ array S whose entries are sets $S[i][j]$ of m key identifiers

```
keycounter:=0;
for i from −r to l_1 − 1 do
    for j from −r to l_2 − 1 do
        for P ∈ DDC do
            if 0 ≤ i + P_0 < l_1, 0 ≤ j + P_1 < l_2
            then
                S[i + P_0][j + P_1] :=
                S[i + P_0][j + P_1] ∪ {keycounter};
            end
        end
        keycounter := keycounter + 1;
    end
end
return S;
```

Example 1. We now illustrate the behaviour of Algorithm 1 by considering some small parameters. Suppose $l_1 = 8$, $l_2 = 6$, $r = 2$ and we wish to distribute keys using the DD$(3, 2)$ whose points are $\{(0, 0), (1, 1), (2, 0)\}$. This DDC can be depicted as follows:



The following table illustrates the key identifiers al-

located to each node in the grid by Algorithm 1.

14 23 7	22 31 15	30 39 23	38 47 31	46 55 39	54 63 47	62 71 55	80 79 63
13 22 6	21 30 14	29 38 22	37 46 30	45 54 38	53 62 46	61 70 54	69 78 62
12 21 5	20 29 13	28 37 21	36 45 29	44 53 37	52 61 45	60 69 53	68 77 61
11 20 4	19 28 12	27 36 20	35 44 28	43 52 36	51 60 44	59 68 52	67 76 60
10 19 3	18 27 11	26 35 19	34 43 27	42 51 35	50 59 43	58 67 51	66 75 59
9 18 2	17 26 10	25 34 18	33 42 26	41 50 34	49 58 42	57 66 50	65 74 58

We see that each square contains three integers, corresponding to the three keys stored by each node. Similarly, each of these integers occurs in precisely three squares (except for those occurring too close to the edge of the network). Two squares that contain the same integer correspond to two nodes that share a key; any two nodes in the grid share at most one key, and any two nodes that share a key occur at a distance of at most 2 (where the width of each grid square is taken to be 1).

The scheme in [4] was designed for a network based on a square grid, but can also be adapted for the hexagonal grid, through the use of the coordinates shown in Fig. 1. We denote by $\mathrm{DD}^*(m,r)$ a set of points in the hexagonal grid such that the difference between any two points is at most r, and the vectors connecting any two pairs of points differ either in length or direction. Algorithm 1 can then be used directly for key predistribution on the hexagonal grid by replacing the $\mathrm{DD}(m,r)$ by a $\mathrm{DD}^*(m,r)$. Rather than considering a network of nodes that lie in a rectangle, in this case we consider nodes lying in a parallelogram of sidelengths l_1 and l_2, with angles of $\pi/3$ and $2\pi/3$ between the sides (Fig. 1 shows such a parallelogram with $l_1 = 5$ and $l_2 = 4$.)

4 Finding an Appropriate Trade-Off Between Storage, Connectivity and Resilience

The behaviour of the KPS described by Algorithm 1 is determined by the choice of distinct-difference configuration used to construct the scheme. Therefore, in order to adapt this scheme for a particular application, it is necessary to appreciate how the properties of the scheme are influenced by the properties of the distinct-difference configuration.

storage Perhaps the most well-defined constraint on the selection of parameters for a KPS is the amount of memory available for storing keys. In a scheme based on a $\mathrm{DD}(m,r)$ or $\mathrm{DD}^*(m,r)$ each node is required to store m keys, thus the appropriate number of dots in the DDC chosen to instantiate the scheme is determined directly by available storage.

one-hop coverage The connectivity of the scheme is directly related to m, since each node shares keys with $m(m-1)$ other nodes (we refer to this as the *one-hop coverage* of the schemes) [4]. Thus there is a direct tradeoff between the storage requirements, and the one-hop coverage of the scheme. The value of the communication range r places constraints on the one-hop coverage that can be achieved: it is shown in [5] that if a $\mathrm{DD}(m,r)$ exists, then $m \leq 0.88623r + O(r^{2/3})$ and if a $\mathrm{DD}^*(m,r)$ exists then $m \leq 0.95231r + O(r^{2/3})$, which in turn leads to upper bounds on the one-hop coverage.

two-hop coverage The two hop coverage of a KPS based on a $\mathrm{DD}(m,r)$ is the number of nodes that can communicate securely with a given node by a two-hop path. Whereas the one-hop coverage of a $\mathrm{DD}(m,r)$ is entirely determined by m, the two-hop coverage depends on the particular configuration that is chosen. In [3] it was shown that it can vary between $2m(m-1)$ and $\frac{1}{4}m(m-1)(m^2-m+6)$. This indicates that the particular choice of DDC has a substantial effect on the connectivity of the resulting KPS.

resilience The value of m has a certain influence on the resilience of the KPS: for a network with n nodes, the number of pairs of nodes that share a key is approximately $n\binom{m}{2}$, whereas the number of links between uncompromised nodes that are compromised when an adversary captures a single node is $m\binom{m-1}{2}$. Thus the proportion of secure links that are affected is approximately $\frac{m-2}{n}$, which increases with m. The resilience of the scheme can be increased at the cost of greater storage and a small amount of computation by the standard technique of replacing each individual key with an instance of Blom's KPS [6,7] (see [12], for example.)

Thus we see that both the connectivity and the resilience of the KPS can be traded against increased storage, hence we effectively have a three-way trade-off between these properties. This suggests that one appropriate method for choosing a $\mathrm{DD}(m,r)$ or $\mathrm{DD}^*(m,r)$ as an input to Algorithm 1 is therefore to pick a DDC with the smallest value of m that still gives the desired level of connectivity, as this minimises the storage and increases the resilience. It can then be combined with Blom's KPS based on polynomials whose degree is as high as is permitted by the storage constraints, in order to boost the resilience. In order to do this, we require efficient techniques for generating DDCs with desired levels of connectivity; we address this issue in Section 5.

5 Construction of DDCs with Good Connectivity

We saw in Section 4 that the connectivity of a $\mathrm{DD}(m,r)$ or $\mathrm{DD}^*(m,r)$ is a good basis for deciding whether to use it in the KPS of [4]. In this section, we consider how to construct DDCs that have good connectivity properties.

In order to control the one-hop coverage of our KPS, is it desirable to be able to construct DDCs for as wide a range of m as possible given the value of r. In Section 5.1 we give algorithms that efficiently generate $\mathrm{DD}(m,r)$ with $m = 0.80795r - o(r)$ and $\mathrm{DD}^*(m,r)$ with $m = 0.86819r - o(r)$, based on constructions from [5]. The resulting DDCs thus have one-hop coverage that is (asymptotically) close to optimal relative to the communication range, and therefore represent a good choice for applications in which the one-hop coverage of the KPS is of paramount importance. In other words, these DDCs can be regarded as having 'close' to the maximum number of dots possible for the given value of r; if configurations with fewer dots are desired, they can be obtained by simply deleting dots from these configurations.

To achieve schemes with high connectivity, we would like to be able to generate DDCs with high two-hop coverage. In [3], Blackburn *et al.* show that for any m there exists a DDC with m dots that attains the maximum possible two-hop coverage of $\frac{1}{4}m(m-1)(m^2-m+6)$. The proof given is constructive, but in practice the $\mathrm{DD}(m,r)$ thus obtained require r to be so large as to render them unsuitable for use with a KPS. However, the overall value of the two-hop coverage is perhaps not even the most useful measure of connectivity, since it measures only the number of secure two-hop paths and not their physical distribution. In the interests of facilitating efficient communication, we would like to guarantee that a nodes can communicate securely with its closest neighbours by either a one-hop or two-hop path. A construction is given in [3] for a $\mathrm{DD}(p+2, \sqrt{2p^2+2p+1})$, where $p \geq 5$ is prime, which leads to a KPS in which each node can communicate securely via a one-hop or two-hop path with all the nodes in a surrounding $(2p-1)\times(2p-3)$ rectangle. We refer to this as *complete two-hop connectivity* within such a rectangle. This is a useful property, as it ensures nodes can communicate securely with their nearest neighbours. We describe this construction in Section 5.2, and extend it to give a construction for a $\mathrm{DD}^*(p+2, \sqrt{3p^2-3p+1})$ with complete two-hop coverage on a parallelogram.

5.1 KPSs with Good One-Hop Coverage

In this section we give an explicit description of how to instantiate constructions from [5] of distinct-difference configurations with large numbers of dots for both the square and hexagonal grid. These constructions each make use of a B_2-*sequence* in $\mathbb{Z}_n$: a set $D = \{d_1, d_2, \ldots, d_i\} \subset \mathbb{Z}_n$ with the property that the differences between any two pairs of numbers in the set are distinct (mod n). In [8], Bose describes a construction of a B_2-sequence in $\mathbf{Z}_{q^2-1}$ containing q elements. Algorithm 2 describes an explicit method of generating the elements of a Bose B_2-sequence. This algorithm requires the use of a quadratic primitive polynomial over the finite field $\mathrm{GF}(q)$; information on generating such polynomials can be found in [16]. We use the notation $M^i_{1,1}$ to denote the top left entry of the i^{th} power of the matrix M.

Example 2. Suppose we wish to construct a Bose B_2 sequence in $\mathbf{Z}_{24}$. The polynomial x^2+4x+2 is a primitive polynomial over $\mathrm{GF}(5)$; we use it to construct the matrix $M = \begin{pmatrix} -4 & 1 \\ -2 & 0 \end{pmatrix}$. Taking successive powers of M, we find that the top left entries of M^0, M^1, M^{14}, M^{16} and M^{21} are 1, hence the desired B_2-sequence consists of the set $\{0, 1, 14, 16, 21\}$. It is easy to check that no two of the 20 possible differences between distinct pairs of elements of this set coincide (mod 24).

We now describe the conversion of a Bose B_2-

Algorithm 2: Construction of a Bose B_2-sequence

Input: elements $a, b \in \mathrm{GF}(q)$ such that $x^2 - ax - b \in \mathrm{GF}(q)[x]$ is a primitive polynomial
Output: a B_2-sequence $D = \{d_1, d_2, \ldots, d_q\} \subset \mathbb{Z}_{q^2-1}$

$M := \begin{pmatrix} a & 1 \\ b & 0 \end{pmatrix}$;
$D := \{0\}$;
for i *from* 1 *to* $q^2 - 2$ **do**
  if $M^i_{1,1} = 1$ **then**
    $D := D \cup \{i\}$;
  end
end
return D;

sequence into a DD(m, r) or DD*(m, r).

5.1.1 DD(m, r) with Good One-Hop Coverage

Algorithm 3 is based on the techniques of [5] and can be used to construct a DD(m, r) with $m = 0.80795r - o(r)$ from a Bose B_2-sequence.

Let $R = \lfloor \frac{r}{2} \rfloor$. We will construct a distinct-difference configuration whose dots are contained in a circle of radius R, which implies that the distance between any two dots is at most r. Let $n = \lfloor 0.914769r \rfloor$ (the constant was chosen in [5] to obtain an optimal construction), and let q be the smallest prime power with $q^2 - 1 \geq n^2$. Algorithm 3 converts a B_2-sequence in $\mathbb{Z}_{q^2-1}$ into a distinct difference configuration whose dots are contained in an $n \times n$ square[2], then takes the intersection of the square with a circle of radius R centred at the centre of the square, in order to produce a DD(m, r) for some $m \leq q$.

If $D = \{d_1, d_2, \ldots, d_q\} \subset \mathbb{Z}_{q^2-1}$ is a B_2-sequence, then so is the set $D + i := \{d_1 + i, d_2 + i, \ldots, d_q + i\}$ for any $i \in \mathbb{Z}_{q^2-1}$; we refer to this as a *shift* of D. We can apply Algorithm 3 to each of the $q^2 - 1$ possible shifts of the Bose B_2-sequence in $\mathbb{Z}_{q^2-1}$ and select the resulting configuration with the greatest number of dots. The results of [5] show that configurations with approximately $0.80795r - o(r)$ dots can be obtained by this method.

Example 3. Suppose we wish to construct a DD$(m, 8)$. Then $n = 7$, so we take $q = 8$ in Algorithm 3. The Bose B_2 sequence in $\mathbb{Z}_{63}$ generated by Algorithm 2 is $\{0, 4, 6, 7, 29, 39, 50, 55\}$. Taking this B_2-sequence as input to Algorithm 3 yields a DD$(3, 8)$. However, if we shift the original sequence by 37 to obtain the sequence $\{3, 13, 24, 29, 37, 41, 43, 44\}$ we obtain the DD$(8, 8)$ whose points are $\{(0, 3), (5, 4), (3, 5), (1, 5), (4, 5), (5, 1), (5, 3), (3, 0)\}$, illustrated in Fig. 2. This is the maximum number of points that can be obtained from any shift of this Bose B_2-sequence.

[2] *i.e.* a set of n^2 grid points arrange in a square

Algorithm 3: Construction of a DD(m, r) from a B_2-sequence

Input: a positive integer r, a B_2-sequence $D = \{d_1, d_2, \ldots, d_q\} \subset \mathbb{Z}_{q^2-1}$ where q is the smallest prime power such that $q^2 - 1 \geq \lfloor 0.914769r \rfloor^2$
Output: a set $DDC := \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^m\}$ of points in $\mathbf{Z}^2$ forming a DD(m, r) for some $m \leq q$

$R := \lfloor \frac{r}{2} \rfloor$;
$n := \lfloor 0.914769r \rfloor$;
$DDC := \{\}$;
for i *from* 0 *to* $n - 1$ **do**
  for j *from* 0 *to* $n - 1$ **do**
    if $iq + j \pmod{q^2 - 1} \in D$ **then**
      if $(\frac{n-1}{2} - i)^2 + (\frac{n-1}{2} - j)^2 \leq R^2$ **then**
        $DDC := DDC \cup \{(i, j)\}$;
      end
    end
  end
end
return DDC;

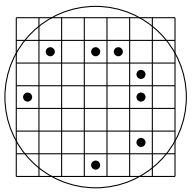


Figure 2: A DD$(8, 8)$ contained in a circle of radius 4

5.1.2 DD*(m, r) with Good One-Hop Coverage

The points of the hexagonal grid are packed more densely than those of the square grid, making it possible to obtain DD*(m, r) with $0.86819r - o(r)$ dots

by a similar construction to that used in the case of the square grid [5].

Let $R = \lfloor \frac{r}{2} \rfloor$, $n = \lfloor 0.914769r \rfloor$, and let q be the smallest prime power with $q^2 - 1 \geq \frac{2}{\sqrt{3}} n^2$. We take a Bose B_2-sequence in $\mathbb{Z}_{q^2-1}$ and apply Algorithm 4, which is essentially a variant of Algorithm 3 adapted to suit the different pattern of grid points in the hexagonal grid. It makes use of the fact that a node of the hexagonal grid labeled (i, j) as in Fig. 1 has Cartesian coordinates $(i - \frac{j}{2}, \frac{\sqrt{3}j}{2})$.

Algorithm 4: Construction of a DD$^*(m, r)$ from a B_2-sequence

Input: a positive integer r, a B_2-sequence $D = \{d_1, d_2, \ldots, d_q\} \subset \mathbb{Z}_{q^2-1}$ where q is the smallest prime power such that $q^2 - 1 \geq \frac{2}{\sqrt{3}} \lfloor 0.914769r \rfloor^2$

Output: a set $DDC := \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^m\}$ of points in $\mathbf{Z}^2$ forming a DD$^*(m, r)$

```
R := ⌊r/2⌋;
n := ⌊0.914769r⌋;
b := ⌊n/√3⌋;
DDC := {};
for j from 0 to 2b − 1 do
    for i from ⌈j/2⌉ to ⌊n − 1 + j/2⌋ do
        if i(2b − 1) + j(b − 1) (mod q^2 − 1) ∈ D
        then
            if ((n−1)/2 − (i − j/2))^2 +
            (√3(b−1)/2 − √3j/2)^2 ≤ R^2 then
                DDC := DDC ∪ {(i, j)};
            end
        end
    end
end
return DDC;
```

As before, the number of points in the configuration arising from this construction can be potentially increased by applying Algorithm 4 to successive shifts of the original Bose B_2-sequence, and selecting the resulting configuration that contains the greatest number of points.

Example 4. Here we consider the construction of a DD$^*(m, 8)$. Now $n = 7$, so we take $q = 8$, as $8^2 - 1 = 63 > \frac{2}{\sqrt{3}} 7^2 \approx 56.58$. The greatest number of points we can obtain from this method is 8, resulting from applying Algorithm 4 to the Bose B_2-sequence shifted by 34. This yields the DD$^*(m, 8)$ whose points (in hexagonal coordinates) are $\{(8, 5), (4, 6), (7, 5), (8, 6), (7, 3), (0, 0), (3, 0), (5, 3)\}$, illustrated in Fig. 3.

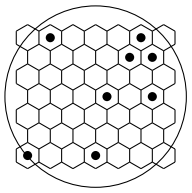


Figure 3: A DD$^*(8, 8)$ contained in a circle of radius 4

5.2 KPSs with Complete 2-Hop Coverage

Algorithm 5 is a construction from [3] based on the Welch construction for a Costas array. It produces DDCs with complete two-hop connectivity on a $2p - 1 \times 2p - 3$ rectangle. If the points of the DD$(\mathrm{p} + 2, \sqrt{2\mathrm{p}^2 + 2\mathrm{p} + 1})$ resulting from this algorithm are interpreted in hexagonal coordinates, they yield a DD$^*(\mathrm{p}+2, \sqrt{3\mathrm{p}^2 - 3\mathrm{p} + 1})$ with complete two-hop coverage in a parallelogram of sides $2p - 1$ and

Algorithm 5: Construction of a DD$(p + 2, \sqrt{2p^2 + 2p + 1})$ with complete two-hop coverage on a $2p - 1 \times 2p - 3$ rectangle

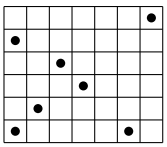
Input: a prime $p \geq 5$, an element $\alpha \in \mathrm{GF}(p)$ that is a primitive element (mod p)

Output: a set $DDC := \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^{p+2}\}$ of points in $\mathbf{Z}^2$ forming a DD$(\mathrm{p} + 2, \sqrt{2\mathrm{p}^2 + 2\mathrm{p} + 1})$ contained in a $(p + 1) \times (p + 2)$ rectangle

```
DDC := {(0, 0), (p, 0), (1, 1), (0, p − 1), (p + 1, p)};
jshift:=0;
while α^(jshift+1) − α^jshift ≢ 1 (mod p) do
    jshift := jshift + 1;
end
ishift := α^jshift;
for i from 2 to p − 1 do
    for j from 2 to p − 2 do
        if α^(j+jshift) ≡ i + ishift (mod p) then
            DDC := DDC ∪ {(i, j)};
        end
    end
end
return DDC;
```

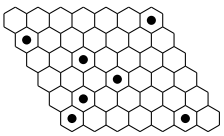
$2p - 3$ (see Fig. 4b).

Example 5. When $p = 5$, Algorithm 5 yields the following DD$(7, 8)$.



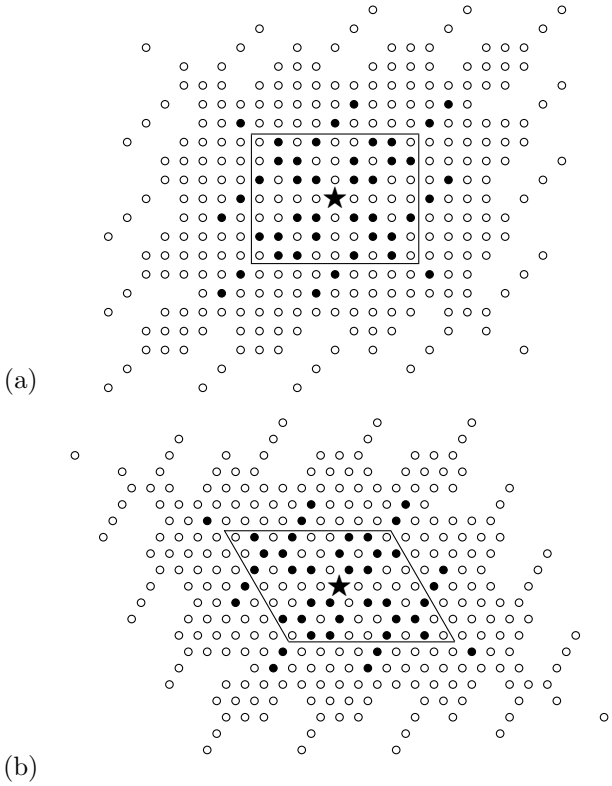
For each node, the pattern of nodes with which it can communicate via a one-hop or two-hop path is that shown in Fig. 4a. The results of [3] guarantee that the nodes within a 9×7 rectangle centred at the node are included in the pattern; as Fig. 4a indicates, the coverage achieved in practice is much greater than this.

When the output of Algorithm 5 is interpreted in terms of hexagonal coordinates, the following DD$(\mathrm{p}+2, \sqrt{3\mathrm{p}^2 - 3\mathrm{p} + 1})$ results; its two-hop coverage is illustrated in Fig. 4b.



6 Conclusion

The full location knowledge of the grid-based scenario means that key predistribution can be particularly effective, as it is possible to specify the precise distribution of keys that is desired. It does not appear to have been considered in the WSN literature prior to [4], however. In [4], key predistribution based on DDCs was compared with a representative selection of schemes from the literature, both location-based and otherwise, and unsurprisingly was shown to achieve greater resilience for a given level of connectivity and storage. In this paper we have analysed how the results of [3, 5] can be applied in practice in a sensor network context. The result is efficient techniques for constructing instantiations of the scheme from [4] that lead to KPSs with good connectivity that are suitable for a range of network parameters.



(a)

(b)

Figure 4: Diagram showing nodes with which the node marked ★ can communicate via a one-hop path (•) or two-hop path (∘) when keys are distributed with the DD$(7, 8)$ (a) or DD$^*(7, 8)$ (b) of Example 5. The rectangle/parallelogram indicate the nodes with which secure one-hop or two-hop communication is guaranteed by the results of [3].

References

[1] Integrated smart sensing systems. http://dpi.projectforum.com/isss/11, 2007.

[2] Institut für Chemie und Dynamik der Geosphäre (ICG), Forschungszentrum Jülich: SoilNet – a Zigbee based soil moisture sensor network. http://www.fz-juelich.de/icg/icg-4/index.php?index=739, 2008.

[3] S. R. Blackburn, T. Etzion, K. M. Martin, and M. B. Paterson. Distinct-difference configurations: Multihop paths and key predistribution in sensor networks. http://arxiv.org/abs/0811.3896, 2008.

[4] S. R. Blackburn, T. Etzion, K. M. Martin, and M. B. Paterson. Efficient key predistribution for grid-based wireless sensor networks. In R. Safavi-Naini, editor, *ICITS*, volume 5155 of *Lecture Notes in Computer Science*, pages 54–69. Springer, 2008.

[5] S. R. Blackburn, T. Etzion, K. M. Martin, and M. B. Paterson. Two-dimensional patterns with distinct differences – constructions, bounds, and maximal anticodes. http://arxiv.org/abs/0811.3832, 2008.

[6] R. Blom. An optimal class of symmetric key generation systems. In *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*, pages 335–338, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[7] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In E. F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 471–486. Springer, 1992.

[8] R. C. Bose. An affine analogue of Singer's theorem. *J. Indian Math. Soc. (N.S.)*, 6:1–15, 1942.

[9] S. A. Çamtepe and B. Yener. Key distribution mechanisms for wireless sensor networks: a survey. Technical Report TR-05-07, Rensselaer Polytechnic Institute, March 2005.

[10] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, New York, NY, USA, 2002. ACM Press.

[11] T. Ito, H. Ohta, N. Matsuda, and T. Yoneda. A key pre-distribution scheme for secure sensor networks using probability density function of node deployment. In V. Atluri, P. Ning, and W. Du, editors, *SASN*, pages 69–75. ACM, 2005.

[12] J. Lee and D. R. Stinson. Deterministic key predistribution schemes for distributed sensor networks. In H. Handschuh and M. A. Hasan, editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2004.

[13] D. Liu and P. Ning. Location-based pairwise key establishments for static sensor networks. In S. Setia and V. Swarup, editors, *SASN*, pages 72–82. ACM, 2003.

[14] K. M. Martin and M. B. Paterson. An application-oriented framework for wireless sensor network key establishment. *Electron. Notes Theor. Comput. Sci.*, 192(2):31–41, 2008.

[15] J. McCulloch, P. McCarthy, S. M. Guru, W. Peng, D. Hugo, and A. Terhorst. Wireless sensor network deployment for water use efficiency in irrigation. In *REALWSN '08: Proceedings of the workshop on Real-world wireless sensor networks*, pages 46–50, New York, NY, USA, 2008. ACM.

[16] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.

[17] K. Römer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications Magazine*, 11(6):54–61, 2004.

[18] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway. A survey of key management schemes in wireless sensor networks. *Comput. Commun.*, 30(11-12):2314–2341, 2007.