On the Role of Expander Graphs in Key Predistribution Schemes for Wireless Sensor Networks

Michelle Kendall and Keith M. Martin

Information Security Group, Royal Holloway,
University of London, Egham, Surrey, TW20 0EX, UK

Abstract. Providing security for a wireless sensor network composed of small sensor nodes with limited battery power and memory can be a non-trivial task. A variety of key predistribution schemes have been proposed which allocate symmetric keys to the sensor nodes before deployment. In this paper we examine the role of expander graphs in key predistribution schemes for wireless sensor networks. Roughly speaking, a graph has good expansion if every 'small' subset of vertices has a 'large' neighbourhood, and intuitively, expansion is a desirable property for graphs of networks. It has been claimed that good expansion in the product graph is necessary for 'optimal' networks. We demonstrate flaws in this claim, argue instead that good expansion is desirable in the intersection graph, and discuss how this can be achieved. We then consider key predistribution schemes based on expander graph constructions and compare them to other schemes in the literature. Finally, we propose the use of expansion and other graph-theoretical techniques as metrics for assessing key predistribution schemes and their resulting wireless sensor networks.

Keywords: Wireless Sensor Networks, Key Predistribution, Expander Graphs.

1 Introduction

A *wireless sensor network* (WSN) is a collection of small, battery powered devices called sensor nodes. The nodes communicate with each other wirelessly and the resulting network is usually used for monitoring an environment by gathering local data such as temperature, light or motion. As the nodes are lightweight and battery powered, it is important to consider battery conservation in order to allow the network to remain effective for the appropriate period of time, and to ensure that the storage required of the nodes is not beyond their memory capacity.

WSNs are suitable for deployment in many different environments, including potentially hostile areas such as military or earthquake zones, where it would be dangerous or impractical to carry out the monitoring of data gathering by hand. In hostile environments it may be necessary to encrypt messages for security and / or authentication. Various cryptographic key management schemes

F. Armknecht and S. Lucks (Eds.): WEWoRC 2011, LNCS 7242, pp. 62–82, 2012.

have been proposed for such scenarios. In some cases there is an online key server or base station to distribute keys to the nodes where needed; if not, *key predistribution schemes* (KPSs) are required, which assign keys to nodes before deployment. Due to the resource-constrained nature of the nodes, it may be infeasible to use asymmetric cryptographic techinques in some WSN scenarios, and so we consider symmetric key predistribution schemes.

Since networks may be modelled as graphs, tools from graph theory have been used both to analyse and to design networks. In particular, we explore the role of expander graphs in KPSs for WSNs. The expansion of a graph is a measure of how well connected it is, and how difficult it is to separate subsets of vertices; we will see the precise definition in Sect. 2.3. The term 'expander graphs' is used informally to refer to graphs with good expansion.

In 2006, expander graph theory was introduced to the study of KPSs for WSNs from two perspectives. On the one hand, Camtepe et al [5] showed that a mathematical construction for an expander graph could be used to design a KPS, resulting in a network which is well connected under certain constraints. On the other hand, Ghosh [13] claimed that good expansion is a necessary condition for 'optimal' WSNs. We examine these claims and determine the role of expander graphs in KPSs for WSNs.

Firstly, we show that Ghosh's claim is flawed but identify where expansion properties are desirable for WSNs, namely in the intersection graph rather than the product graph. We then analyse the effectiveness of constructions for KPSs based on expander graphs, showing that they provide perfect resilience against an adversary but lower connectivity and expansion than many existing KPSs, for the same network size and key storage. We argue that expansion is an important metric for assessing KPSs to be used alongside the other common metrics of key storage, connectivity and resilience for a given network size. However, we note the difficulty of finding the expansion coefficient of a graph and so propose estimating the expansion and using other graph-theoretical techniques to indicate weaknesses.

We begin by introducing the relevant terminology and concepts in Sect. 2. In Sect. 3 we outline Ghosh's claims and show by means of a counter-example that his conclusion is misdirected towards expansion in product graphs rather than intersection graphs. In Sect. 4 we discuss how to maximise the probability of a high expansion coefficent in the intersection graph, and in Sect. 5 we analyse the extent to which KPSs based on expander graph constructions achieve this, in comparison to other schemes from the literature. Finally, in Sect. 6 we suggest practical metrics for analysing and improving KPSs and the resulting WSNs, and conclude in Sect. 7.

2 Background

2.1 Key Predistribution Schemes for Wireless Sensor Networks

A *key predistribution scheme* is a well-defined method for determining the combination of keys which should be stored on each node before deployment. Once

the nodes have been deployed in the environment, they broadcast identifiers which uniquely correspond to the keys they store, and determine the other nodes (within communication range) with which they share at least one common key, in order to form a WSN.

There are many ways of designing a KPS, and different KPSs suit different WSN applications. We consider KPSs which assign symmetric keys, since small sensor nodes are resource-constrained with low storage, communication and computational abilities, and are often unable to support asymmetric cryptography. In order to make best use of the nodes' limited resources, it is usually desirable to minimise the *key storage* requirement whilst maximising the *connectivity* and *resilience* of a network of n nodes. We now define these concepts more precisely.

- *Key storage* is the maximum number of keys which an individual node is required to store for a particular KPS.
- *Connectivity* of a network can be measured or estimated both *globally* and *locally*. We will refer again to global connectivity in Sect. 6 but in general we will use the measure of local connectivity Pr_1. This is defined to be the probability that two randomly-chosen nodes are 'connected' because they have at least q keys in common. Most KPSs require nodes to share just one key before they can establish a secure connection, ie. $q = 1$, and so Pr_1 is simply the probability that a random pair of nodes have at least one key in common. Some schemes such as the q-composite scheme of Chan et al. [6] introduce a threshold such that nodes may only communicate if they have at least $q > 1$ common keys. Where two nodes share more than q keys, some protocols dictate that they should use a combination of those keys, such as a hash, to encrypt their communications.
- *Resilience* is a measure of the network's ability to withstand damage from an adversary. We use the adversary model of a continuous, listening adversary, which can listen to any communication across the network and continually over time 'compromise' nodes, learning the keys which they store. We measure the resilience with the parameter fail_s: we suppose that an adversary has compromised s nodes, and then compute the probability that a link between two uncompromised nodes in the network is compromised, that is, the adversary knows the key(s) being used to secure it. Equivalently, fail_s measures the fraction of compromised links between uncompromised nodes throughout the network, after an adversary has compromised s nodes. Notice that high resilience corresponds to a low value of fail_s. We say that a network has *perfect resilience* against such an adversary if $\mathsf{fail}_s = 0$ for all $1 \leq s \leq n - 2$.

To illustrate the trade-offs required between these three parameters, we consider some trivial examples of KPSs.

1. *Every node is assigned a single key k before deployment.*
   This would require minimal key storage and ensure that any pair of nodes could communicate securely, so $\mathsf{Pr}_1 = 1$ for all pairs of nodes. However, there would be minimal resilience against an adversary, as the compromise

of a single node would reveal the key k, rendering all other links insecure. Formally, $\mathsf{fail}_s = 1$ for all $1 \leq s \leq n-2$.

2. *A unique key is assigned to every pair of nodes.*
That is, for all $1 \leq i, j \leq n$, nodes n_i and n_j are both preloaded with a key k_{ij}, where $k_{ij} \neq k_{lm}$ for all pairs $(l, m) \neq (i, j)$. This is called the *complete pairwise* KPS. Such a KPS would have perfect resilience and maximum connectivity, as $\mathsf{Pr}_1 = 1$ for all pairs of nodes. However, each node would have to store $n-1$ keys, which is infeasible when n is large.
3. *Every node is assigned a single unique key.*
Whilst providing minimal key storage and perfect resilience, this KPS has no connectivity, as $\mathsf{Pr}_1 = 0$ for all pairs of nodes.

We see, then, that it is trivial to optimise any two of these three parameters. However, for many WSN applications these schemes are inappropriate, and so we consider KPSs which find trade-offs between all three of these metrics. A variety of such KPSs have been proposed, both deterministic and random, a survey of which is given in [4][7][19][23].

We now describe the Eschenauer Gligor KPS [12] as an example of a KPS which provides values for the three metrics which are appropriate for many WSN scenarios. It will also be used as a comparison for KPSs based on expander graph constructions in Sect. 5.

Example 1. The Eschenauer Gligor KPS [12] works in the following way. A key pool $\mathcal{K}$ is generated. To each node n_i is assigned a random subset of k keys from $\mathcal{K}$. The probability of two nodes sharing at least one key is

$$\mathsf{Pr}_1 = 1 - \frac{\binom{|\mathcal{K}|-k}{k}}{\binom{|\mathcal{K}|}{k}} \quad . \tag{1}$$

An equivalent formula is given in [12]. We verify (1) by considering the probability of two arbitrary nodes n_i and n_j sharing no common keys. If node n_i stores a set S_i of k keys, node n_j stores S_j and $S_i \cap S_j = \emptyset$, then every key of S_j must have been picked from the key pool $\mathcal{K} \setminus S_i$, that is from a set of $|\mathcal{K}| - k$ keys. Therefore the probability of two nodes having no keys in common is equal to the number of ways of choosing k keys from a key pool of $|\mathcal{K}| - k$, divided by the number of ways of choosing k keys from the full key pool.

As explained in [6], the resilience after the compromise of s nodes is

$$\mathsf{fail}_s = \left(1 - \left(1 - \frac{k}{|\mathcal{K}|}\right)^s\right)^q \quad . \tag{2}$$

where q is the number of keys shared between two randomly-chosen uncompromised nodes. This leads to the intuitive result that if the adversary has compromised one node, learning k keys, and two randomly-chosen uncompromised nodes share $q = 1$ key k_1, then the probability of the adversary knowing k_1 is $\mathsf{fail}_1 = \frac{k}{|\mathcal{K}|}$. If $q > 1$ keys are shared between the nodes then the value of fail_1 will be smaller, as $\mathsf{fail}_1 = \left(\frac{k}{|\mathcal{K}|}\right)^q$.

Table 1 demonstrates the value of Pr_1 *and upper bound for* fail_1 *for some different sizes of key pool and key storage. (We assume for this example that all nodes are within communication range of one another.) It can be seen that by adjusting the values of* $|\mathcal{K}|$ *and* k*, with* k *small, we can achieve arbitrarily large* Pr_1 *whilst keeping* fail_1 *relatively small. This makes the Eschenauer Gligor KPS appropriate for many WSN scenarios.*

Table 1. Example values for an Eschenauer Gligor KPS

$\lvert\mathcal{K}\rvert$	k	Pr_1	fail_1
500	25	0.731529	0.050
500	50	0.996154	0.100
1000	25	0.473112	0.025
1000	50	0.928023	0.050

In accordance with most papers in the literature, we will use key storage, connectivity and resilience along with network size as metrics for comparing KPSs. Network size is relevant since for small networks the complete pairwise KPS is practical, and because some KPSs are not adaptable for all sizes of network. For example we will see in Sect. 5 that the KPS proposed by Camtepe et al [5] is only possible for networks of size $n = t + 1$ where t is a prime congruent to 1 mod 4.

Later, in Sect. 6, we will propose that in addition to network size, key storage, connectivity and resilience, it is important to consider expansion as a metric when comparing KPSs.

2.2 Graph Theory

We now introduce some graph-theoretic definitions, beginning with general terminology in this section before giving the specific definitions related to expander graphs in Sect. 2.3.

A *graph* $G = (V, E)$ is a set of *vertices* $V = \{v_1, \ldots, v_n\}$ and a set of edges E. We use the notation $(v_i, v_j) \in E$ to express that there is an edge between the vertices v_i and v_j, and we say that the edge (v_i, v_j) is *incident* to its endpoints v_i and v_j. Wherever an edge (v_i, v_j) exists, v_i and v_j can be said to be *adjacent*.

All graphs considered in this paper will be *simple graphs*, that is, they are *unweighted*, *undirected* and do not contain *self-loops* or *multiple edges*. These terms respectively mean that vertices are not assigned different weights, edges are not directed from one vertex to the other, there are no edges from the a node to itself, and any edge between two vertices is unique.

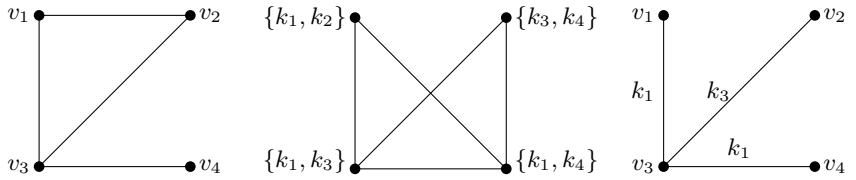
Given subsets of vertices $X, Y \subset V$, the set of edges which connect X and Y is denoted

$$E(X, Y) = \{(x, y) : x \in X, y \in Y \text{ and } (x, y) \in E\} \ ,$$

and the *complement* $\overline{X}$ of X is the vertices which are not in X, that is, $\overline{X} = V \backslash X$.

An ordered set of consecutive edges $\{(v_{i1}, v_{i2}), (v_{i2}, v_{i3}), \ldots, (v_{i(p-1)}, v_{ip})\}$ in which all the vertices $v_{i1}, v_{i2}, \ldots, v_{ip}$ are distinct is called a *path* of length $p-1$. A *cycle* is a 'closed' path which begins and ends at the same vertex, ie. a cycle is a path $\{(v_{i1}, v_{i2}), (v_{i2}, v_{i3}), \ldots, (v_{i(p-1)}, v_{ip})\}$ where $v_{i1}, v_{i2}, \ldots, v_{i(p-1)}$ are distinct but $v_{i1} = v_{ip}$. We say that a graph is *connected* if there is a path between every pair of vertices, and *complete* if there is an edge between every pair of vertices. The *degree* $d(v)$ of a vertex v is the number of edges incident to that vertex. If all nodes have the same degree r, we call the graph *r-regular*.

We draw a graph of a WSN by representing the nodes as vertices and the 'connections' as edges. To be precise in our analysis, we distinguish between the two possible types of 'connection' and consider the separate constituent graphs of a network: the *communication graph* $G_1 = (V, E_1)$ where $(v_i, v_j) \in E_1$ if v_i and v_j are within communication range, and the *key graph* $G_2 = (V, E_2)$ where $(v_i, v_j) \in E_2$ if v_i and v_j share at least q common keys. An example of a communication graph and a key graph are given in Fig. 1.



(a) Comm. graph (b) Key graph (c) Intersection graph

Fig. 1. Example of corresponding communication, key and intersection graphs

If the communication graph is complete, it is often omitted from the analysis as there is no need to check whether nodes can communicate. However, as we will explain in Sect. 4, the communication graph is commonly modelled using a random graph, and it then becomes important to analyse how the communication and key graphs relate to each other.

We say that two nodes v_i and v_j can *communicate securely* if $(v_i, v_j) \in E_1 \cap E_2$, that is if they are adjacent in the *intersection* graph $G_1 \cap G_2 = (V, E_1 \cap E_2)$. This is illustrated in Fig. 1(c). We note that the standard definition of an intersection graph is $G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$, but throughout this paper $V_1 = V_2$ and so we simply refer to the set of vertices as V.

If two nodes are not adjacent in the intersection graph then there are usually ways for them to communicate by routing messages through intermediary nodes and/or establishing a new key. Since any protocol for either of these methods requires extra communication overheads, it is desirable to minimise the *diameter* of the intersection graph, that is to minimise the longest path length between nodes. Similarly, it may also be desirable to minimise the average path length of the intersection graph.

Finally, we introduce another way of combining two graphs, which will be needed for Sect. 3 where we consider Ghosh's claims.

Definition 1. *The (Cartesian)* product graph *of two graphs* $G = (V_G, E_G)$ *and* $H = (V_H, E_H)$ *is defined as* $G.H = (V_G \times V_H, E_{G.H})$*, where the set of edges* $E_{G.H}$ *is defined in the following way:* $(uv, u'v') \in E_{G.H}$ *if*

$$\begin{gathered}(u = u' \text{ or } (u, u') \in E_G) \\ \text{and} \\ (v = v' \text{ or } (v, v') \in E_H) \quad .\end{gathered}$$

We will now define expander graphs and explain why their properties are desirable for WSNs.

2.3 Expander Graphs

For a thorough survey of expander graphs and their applications, see [14]. The *expansion* of a graph is a measure of the quality of its connectivity.

Definition 2. *A finite graph* $G = (V, E)$ *is an* ϵ-expander graph*, where the edge-expansion coefficient* ϵ *is defined by*

$$\epsilon = \min_{S \subset V : |S| \leq \frac{|V|}{2}} \left(\frac{|E(S, \overline{S})|}{|S|} \right) \quad ,$$

where $|E(S, \overline{S})|$ *denotes the number of edges from the set* S *to its complement.*

The phrase 'expander graph' is used informally to refer to graphs with good expansion, that is, graphs with a high value of ϵ, as we explain below. We note that definitions vary across the literature, in particular some defintions use the strict inequality $|S| < \frac{|V|}{2}$. Another name for the edge-expansion coefficient is the *isoperimetric number*, and in a graph where every vertex has the same weight this is equivalent to the *Cheeger constant*; see [9] for further details.

We now explore what the definition of the edge-expansion coefficient ϵ means, and why a high value of ϵ is desirable, through the following observations:

- If $\epsilon = 0$ then we see from the definition that there exists a subset $S \subset V$ without any edges connecting it to the rest of the graph. This implies that the graph is not connected.
- A graph is connected if and only if $\epsilon > 0$ (see proof of Proposition 1), hence all connected graphs are ϵ-expander graphs for some positive value of ϵ.
- If ϵ is 'small', for example $\epsilon = \frac{1}{100}$, then there exists a set of vertices S which is only connected to the rest of the graph by one edge per 100 nodes in S. This is undesirable for a WSN for the following reasons:
  - The set S is vulnerable to being 'cut off' from the rest of the network by a small number of attacks or faults. If S contains $c \times 100$ nodes then there are only c edges between S and $\overline{S}$. A small number of compromises or failures amongst the particular $\leq 2c$ nodes incident to these edges will render all communication between S and $\overline{S}$ insecure.

  - Since S is connected to the rest of the network by comparatively few edges, a higher communication burden is placed on the small set of $\leq 2c$ nodes, since a higher proportion of data needs to be routed through them. This will drain the batteries of the nodes nearest to the edges between S and $\overline{S}$ faster than those of an average node, so that after some period of time they will run out of energy, disconnecting S from the rest of the network even though many nodes in S may still have battery power remaining.
  - Reliance on a small number of edges to connect large sets of nodes may create bottlenecks in the transmission of data through the network, making data collection and/or aggregation less efficient.
- If ϵ is larger, particularly if $\epsilon > 1$, then there is no 'easy' way to disconnect large sets of nodes, and communication burdens, battery usage and data flow are more evenly spread.

We see from these observations that intersection graphs with higher values of ϵ are more desirable for WSNs. A graph with a 'large' value of ϵ is often said to have 'good expansion'. The the size of ϵ is subject to the following bounds.

Proposition 1. *For any connected graph $G = (V, E)$ with $|V| \geq 2$,*

$$0 < \epsilon \leq \min_{v \in V} d(v) \ .$$

Proof. We begin by considering the lower bound. Suppose for a contradiction that $\epsilon = 0$. Then there exists a set $S \subset V$ such $|E(S, \overline{S})| = 0$. This contradicts the fact that G is connected. Since ϵ cannot be negative, we have that $\epsilon > 0$.

For the upper bound, consider the set $S = \{v\}$ where $v \in V$. It is clear that $|E(S, \overline{S})| = d(v)$, where $d(v)$ is the degree of v as defined in Sect. 2.2, and so $\frac{|E(S,\overline{S})|}{|S|} = \frac{d(v)}{1} = d(v)$. Since the definition of the edge-expansion coefficient ϵ uses the *minimum* value over all $S \subset V$ with $|S| \leq \frac{|V|}{2}$, we have that $\epsilon \leq \min_{v \in V} d(v)$. □

In addition to the observations made above, graphs with good expansion also have low diameter, logarithmic in the size of the network [14] and contain multiple short, disjoint paths between nodes [16], which is beneficial for schemes like the multipath reinforcement of Chan et al. [6]. These properties mean that key graphs with good expansion are particularly desirable for WSNs.

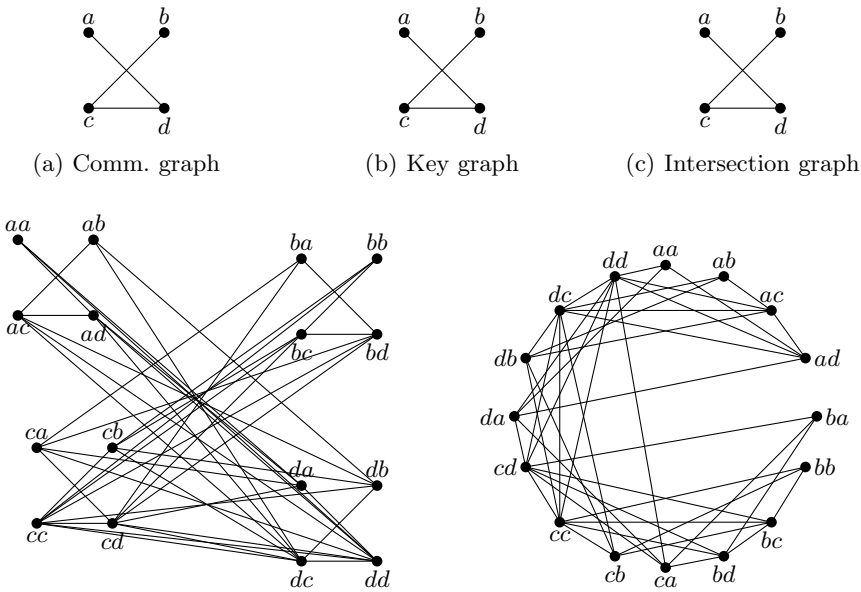
The papers by Camtepe et al. [5] and Shafiei et al. [21] propose KPSs based on expander graph constructions. These methods of designing a KPS ensure that the key graph has good expansion, and we further examine these proposals in Sect. 5. First, we consider the claims made by Ghosh in [13] about the necessity of good expansion for 'optimal' networks.

3 Expansion in Product Graphs

In [13] Ghosh considers KPSs with large network size, low key storage per node, high connectivity and high resilience. He considers jointly 'optimising' these

parameters, although exactly what this means is unclear, since different applications will prioritise them differently. Nevertheless, he argues that if a KPS is in some sense 'optimal', the product graph of the key graph and communication graph must have 'good expansion properties'. We show by a counterexample that expansion in the product graph is not a helpful measure because the product graph is almost inevitably an expander graph. Additionally, we show that the product graph is unable to capture the required detail to analyse a WSN, and that it is the intersection graph where such analysis is relevant.

In Fig.s 2 and 3 we consider examples of product graphs and examine how they relate to their constituent communication and key graphs. Figure 2 shows a communication and a key graph, and their corresponding intersection and product graphs. The product graph is represented in Fig. 2(d) in a way which demonstrates its construction, and redrawn in Fig. 2(e) for clarity.



(a) Comm. graph (b) Key graph (c) Intersection graph

(d) Cartesian product graph (e) Cartesian product graph re-drawn

Fig. 2. A product graph corresponding to an identical communication and key graph pair

Figure 2(d) illustrates that the product graph construction results in four copies of the key graph, connected to each other in a pattern which resembles the communication graph. To provide an alternative perspective, we re-draw the same graph with the vertices arranged in a circle in Fig. 2(e). To understand the construction, recall Def. 1 which defines the vertices and edges of the product graph. We find that there is an edge in the product graph $(ac, ab) \in E_{G.H}$ because

$a = a$ and $(c, b) \in E_H$. Similarly, $(ca, ba) \in E_{G.H}$ because $(c, b) \in E_G$ and $a = a$. However, we find that $(aa, ab) \notin E_{G.H}$ because whilst $a = a$, $(a, b) \notin E_H$.

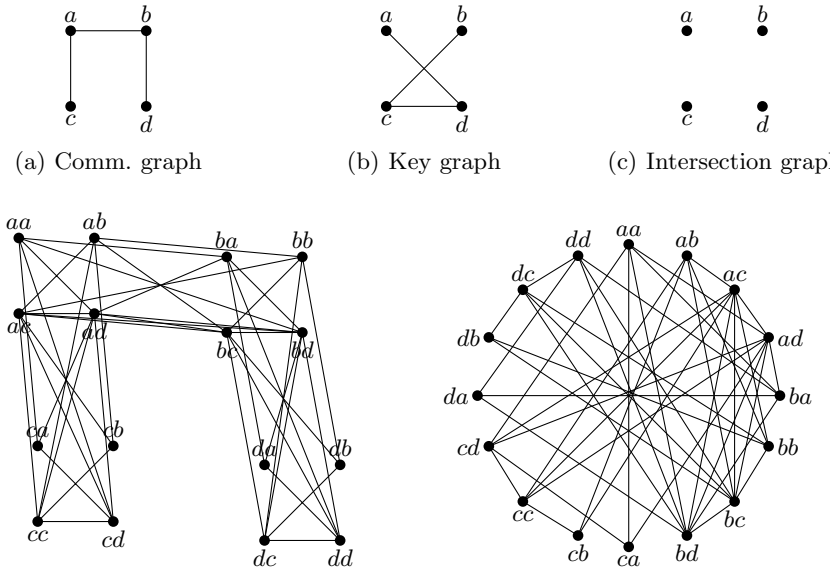
In Fig. 2 the communication and key graphs are identical, giving the best possible case for intersection. We now calculate the expansion coefficient of the product graph. Consider sets S of $1, 2, \ldots, 8$ vertices (recall from the definition that we should consider subsets S with $|S| \leq \frac{|V|}{2}$, and here $|V| = 16$). We observe that any single vertex is connected to the rest of the graph by at least three edges, any set of two vertices is connected to the rest of the graph by at least six edges, etc., so that

$$\epsilon = \min \left\{ \frac{3}{1}, \frac{6}{2}, \frac{9}{3}, \frac{9}{4}, \frac{11}{5}, \frac{16}{6}, \frac{12}{7}, \frac{10}{8} \right\} .$$

That is, $\epsilon = \frac{10}{8} = \frac{5}{4}$, so the product graph of Fig. 2 has expansion coefficient $\epsilon = \frac{5}{4}$.

Now consider Fig. 3, where we have the same key graph but the communication graph is altered. It has the same number of edges as in Fig. 2 but in such a way that the intersection graph, shown in Fig. 3(c), has no edges. Clearly for WSN purposes this would mean that no secure communication was possible.

However, the product graph does have edges, and indeed appears well connected. By observation, we find that it too has expansion coefficient $\epsilon = \frac{5}{4}$.



(a) Comm. graph (b) Key graph (c) Intersection graph

(d) Cartesian product graph (e) Cartesian product graph re-drawn

Fig. 3. A product graph corresponding to a communication and key graph pair with empty intersection

Indeed, after some inspection, we find that the product graphs of Fig.s 2 and 3 are isomorphic, using a simple bijection to relabel vertices as follows:

Fig. 2(e)		Fig. 3(e)
$(a*)$	$\rightarrow$	$(c*)$
$(b*)$	$\rightarrow$	$(d*)$
$(c*)$	$\rightarrow$	$(b*)$
$(d*)$	$\rightarrow$	$(a*)$

This means that all graph-theoretic properties of connectivity, expansion, degree, diameter etc. are identical between the two product graphs. From this we see that a product graph with good expansion can occur when the key and communication graphs intersect 'fully', ie. when $E_G \cap E_H = E_G = E_H$, and when there are no edges in the intersection, ie. $E_G \cap E_H = \emptyset$. This shows that the expansion of the product graph certainly does not correspond to any degree of 'optimality' regarding the intersection graph and therefore the WSN. In particular, it strongly suggests that expansion in the product graph is not a good tool for analysing the connectivity of WSNs without reference to the intersection graph. Ghosh's claim that an 'optimal' combination of key and communication graph will result in a product graph with good expansion tells us very little, since good expansion in the product graph is almost inevitable, as we will now explain.

Proposition 2. *A (Cartesian) product graph $G.H = (V_G \times V_H, E_{G.H})$ is connected if and only if both $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are connected.*

Proof. If the product graph is connected then there is a path between all pairs of vertices $u_1v_1, u_pv_q \in V \times V$, say

$$(u_1v_1, u_2, v_2), (u_2v_2, u_3v_3), \ldots, (u_{p-1}v_{q-1}, u_pv_q) \ .$$

Using the definition of the product graph, this implies that either $u_1 = u_2$ or $(u_1, u_2) \in E_G$, and indeed for all $1 \leq i \leq p-1$, either $u_i = u_{i+1}$ or $(u_i, u_{i+1}) \in E_G$. Thus either $u_1 = u_p$ or there is a path from u_1 to u_p in G. Since this is true for all pairs of vertices $u_1, u_p \in V$, we have that G is connected. By the same argument, H is also connected.

Suppose that G and H are both connected graphs. Then for each distinct pair of vertices $u_1, u_p \in V_G$, there is a path between them, say

$$(u_1, u_2), (u_2, u_3), \ldots, (u_{p-1}, u_p) \ .$$

Similarly, for each distinct pair of vertices $v_1, v_q \in V_H$, there is a path, say

$$(v_1, v_2), (v_2, v_3), \ldots, (v_{q-1}, v_q) \ .$$

By the definition of $E_{G.H}$, we have that $(u_1v_1, u_2v_2) \in E_{G.H}$. Thus we can construct a path

$$(u_1v_1, u_2v_2), (u_2v_2, u_3v_3), \ldots, (u_{p-1}v_{q-1}, u_pv_q)$$

in $G.H$ between any pair of vertices, and therefore $G.H$ is connected. □

Corollary 1. *If G and H are connected, the product graph $G.H$ has expansion coefficient $\epsilon_{G.H} > 0$.*

Proof. Recall from Prop. 1 that a connected graph is an expander graph for some value of ϵ. Therefore, if G and H are connected, the product graph will be an expander graph for some value of $\epsilon_{G.H} > 0$. □

We conjecture that with high probability, $\epsilon_{G.H} > \epsilon_G, \epsilon_H$ and $\epsilon_{G.H} >> 0$. We justify this by considering the comparatively large degrees of nodes in the product graph, and the product graph's similarity to an expander graph construction.

For any node $v \in V$ with degrees $d_G(v), d_H(v)$ in the communication and key graphs respectively, we can compute its degree in the product graph as

$$d_{G.H}(v) = d_G(v)d_H(v) + d_G(v) + d_H(v) \ . \tag{3}$$

Using Prop. 1, we have that

$$\epsilon_{G.H} \leq \min_{v \in V} \left(d_G(v)d_H(v) + d_G(v) + d_H(v)\right) \ ,$$

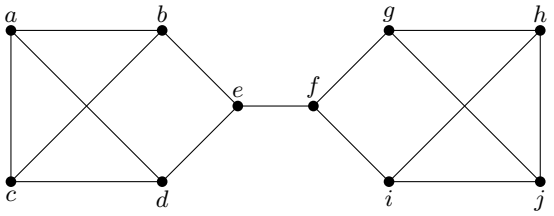
a much higher bound than for the constituent graphs. Since, on average, vertices of the product graph have higher degree than vertices in the constituent graphs, and since the construction of the product graph makes 'isolated' sets of vertices extremely unlikely, we see that $\epsilon_{G.H}$ is likely to be large, and in particular greater than either of ϵ_G and ϵ_H. By comparison, the expansion coefficient of the intersection graph $\epsilon_{G \cap H}$ is forced to be no more than those of the constituent graphs, ϵ_G and ϵ_H, as explained in the next section.

Additionally, the construction of the product graph is not dissimilar to that of the *zig-zag product* graph presented in [20] as an expander graph construction, and used by Shafiei et al in [21] to produce key graphs with good expansion. We see then that expansion in the product graph is inevitable if the constituent graphs are connected, is likely to be 'good', and does not imply anything about the quality of the connectivity or expansion in the intersection graph, where it is needed. Ghosh does not justify his choice of using the product graph as a means of studying two graphs simultaneously, and we conclude that there are no benefits to doing so. In order to capture the relevant interaction between the key and communication graphs, the intersection graph is the relevant tool, and it is in the intersection graph where good expansion is desired.

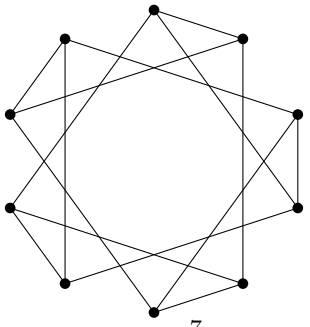
4 Expansion in Intersection Graphs

We claim that when comparing two WSNs of the same size with identical key storage, connectivity and resilience parameters, the WSN represented by the intersection graph with higher expansion will be the more robust, with a more evenly distributed flow of data. We justify this using the following example.

Example 2. Consider Fig. 4 and suppose that these are two intersection graphs, representing WSNs. Each graph is 3-regular on 10 nodes. We suppose that an



(a) $\epsilon = \frac{1}{5}$



(b) $\epsilon = \frac{7}{5}$

Fig. 4. Examples of 3-regular graphs on 10 nodes with different expansion parameters.

Eschenauer Gligor KPS (as described in Sect. 2.1) has been used to construct the key graph, where each node stores three keys chosen randomly from a pool of 25 keys. To simplify the analysis, we say that where nodes have more than one key in common, they select just one of them for use in securing their communications.

Using the formulae given in Sect. 2.1 we calculate that $\mathsf{Pr}_1 = 1 - \frac{\binom{22}{3}}{\binom{25}{3}} \approx 0.33$, $\mathsf{fail}_1 = \frac{3}{25}$ *and* $\mathsf{fail}_s = 1 - \left(1 - \frac{3}{25}\right)^s$ *for both graphs. Calculating the expansion, we observe that in Fig. 4(a)* $\epsilon = \frac{1}{5}$*. This minimum value is achieved by (for example) picking the set of 5 vertices* $S = \{a, b, c, d, e\}$*, which is only connected to the rest of the graph by the single edge* (e, f)*. However, in Fig. 4(b) we find that* $\epsilon = \frac{7}{5}$*; any set of 5 vertices is connected by at least 7 edges to the rest of the graph.*

For WSN applications, the network represented by Fig. 4(a) is less desirable, because

- *it is more vulnerable to a listening adversary, who could decrypt a high proportion of communications through the network by the compromise of a single node* e *or* f
- *nodes* e *and* f *are more vulnerable to battery failure*
- *battery failure of just one of the two nodes* e *and* f *would disconnect the network*

– *communication bottlenecks are likely to occur around nodes e and f, making communication through the network less efficient.*

Conversely, in Fig. 4(b) the communication burdens are distributed evenly across the nodes so that battery power will be used more evenly and there are no weak spots for an adversary to target in order to quickly damage the rest of the network. The graph can only be split into disjoint sets by the removal of 4 or more nodes, that is, almost half of the network. It is clear then that Fig. 4(b) represents the less vulnerable WSN.

From this example we see that some strengths and weaknesses of the 'layout' of the WSN are hidden if we only consider the size, key storage, connectivity and resilience, and in Sect. 6 we discuss the practicality of using expansion as another metric for assessing networks. Before that, we consider how best to probabilistically maximise the expansion in an intersection graph, and in Sect. 5 we will consider schemes which aim to produce key graphs with good expansion.

We now consider how to achieve high expansion in an intersection graph $G \cap H$.

Proposition 3. *An intersection graph* $G \cap H = (V \cap V, E_G \cap E_H)$ *has expansion parameter*

$$\epsilon_{G\cap H} \leq \min\{\epsilon_G, \epsilon_H\} \ .$$

Proof. We begin by considering the degree of a node in the intersection graph, which is

$$d_{G\cap H}(v) \leq \min\left(d_G(v), d_H(v)\right)$$

because each edge (v, w) incident to a vertex v in G will be removed in the intersection unless there is also an edge (v, w) in H. Using Prop. 1, we have that $\epsilon_{G\cap H} \leq \min_{v\in V}\{d_{G\cap H}(v)\}$.

Without loss of generality, suppose that $\epsilon_G \leq \epsilon_H$. Consider a set S of vertices in G which achieves the minimum $\frac{|E(S,\overline{S})|}{|S|} = \epsilon_G$. If every edge of $E(S, \overline{S})$ remains in the intersection then $\epsilon_{G\cap H} \leq \epsilon_G$, otherwise $\epsilon_{G\cap H} < \epsilon_G$, since no edges are added elsewhere in the intersection. Therefore we have that $\epsilon_{G\cap H} \leq \min\{\epsilon_G, \epsilon_H\}$. □

We see that it is necessary that G and H have high expansion coefficients for $G \cap H$ to be a good expander. If the communication graph is complete then the expansion of the key graph will be preserved in the intersection. If information about the locations of the nodes is known a priori or if there is some control over the communication graph, then keys can be assigned to nodes in a more efficient manner; see [18] for a survey of KPSs for such scenarios.

However, we usually assume that there is little or no control over the communication graph and model it as a random graph, typically using either the Erdös Rényi model [11] or the random geometric model, as in [5]. If the communication graph is random, all that can be done to aid good expansion in the intersection graph is to design the KPS so that the key graph has as high expansion as possible for a particular network size and for given levels of key storage, connectivity and resilience.

5 Analysing the Expansion Properties of Existing KPSs

Many KPSs produce key graphs with high expansion coefficients for chosen levels of key storage and resilience, as demonstrated by the following examples.

- Random graphs are good expanders with high probability [14], and so Eschenauer and Gligor's random KPS [12] is likely to produce key graphs which are good expanders, as are other random KPSs such as those given in [6].
- Deterministic schemes based on combinatorial designs, as unified in [19], typically guarantee properties such as constant node degree and μ common intersection. That is, if two nodes are not adjacent then they have μ common neighbours, meaning that the graph has diameter 2, and is therefore a good expander. In particular, two deterministic KPSs based on constructions for 'strongly regular' graphs are given in [17].
- Camtepe et al. [5] and Shafiei et al. [21] propose KPSs based on expander graph constructions and demonstrate that these schemes compare well to other well-regarded KPS approaches.

We now consider the KPSs based on expander graph constructions in more detail, and compare them to the other schemes listed above. Camtepe et al. [5] use the Ramanujan construction which produces an 'asymptotically optimal' expander graph (see [14]) for network size $n = t+1$ and key storage $k = s+1$, where t and s are primes congruent to 1 mod 4. Shafiei et al. [21] use the zig-zag construction, which has the benefit of being more flexible to produce key graphs for any sizes of n and k. Both papers use the following method:

1. construct an expander graph G for the appropriate network size and degree
   - in the case of [5], remove any self-loops or multiple edges and replace with randomly-selected edges such that all nodes have the same degree
2. assign a unique pairwise key to every edge of G
3. preload each node with the set of keys which correspond to its set of edges

This ensures that the key graph has high expansion for the chosen network size and node degree r which equals the key storage k.

However, we claim that it is possible to achieve higher expansion in a KPS for the same network size and key storage. This is because in the KPSs based on expander graph constructions, the node degree r is the same as the key storage k, because unique pairwise keys are used. In other KPSs we usually expect that $k < d(v)$ for all vertices v, as illustrated by the following example.

Example 3. In the Eschenauer Gligor random KPS [12], the key storage k is almost certainly less than the degree $d(v)$ of each node $v \in V$ in the key graph. For example, if nodes store 50 *keys randomly selected from a pool of 1000 keys, then the expected degree of any node is*

$$(n-1) \times \left(1 - \frac{\binom{950}{50}}{\binom{1000}{50}}\right) \quad .$$

If the network has 1000 nodes, this means that the expected degree is ≈ 71.905. *This implies that for the same values of* k *and* n, Pr_1 *is greater in the Eschenauer Gligor scheme than in KPSs produced by expander graph constructions. Since random graphs are known to be good expanders with high probability, this means that, contrary to intuition, a key graph based on an expander graph construction is likely to be a worse expander than a key graph generated by the Eschenauer Gligor scheme.*

Similarly, most schemes based on combinatorial designs also reuse keys so that $k < d(v)$, and therefore produce key graphs with higher average degree and better expansion than those based on expander graph constructions. A benefit of the KPSs based on expander graph constructions is that each key is only used for one edge, meaning that the graphs have perfect resilience: $\mathsf{fail}_s = 0$ for all $1 \leq s \leq n-2$. Therefore, in comparison to many other comparable schemes with perfect resilience, these constructions do produce key graphs with good expansion. However, for fixed values of k and n, if it is desirable to achieve high expansion and higher connectivity at the cost of slightly lowered resilience, then a KPS based on an expander graph construction is not the best choice.

6 Using Expansion as a Metric

We have seen that for two networks of the same size with fixed values of key storage, connectivity and resilience, the WSN represented by the intersection graph with the highest expansion coefficient ϵ is the more robust, with the more evenly distributed flow of data. Therefore we suggest that expansion is an important metric to be considered alongside those listed above, when designing KPSs and assessing their suitability for use in WSNs. However, we now state some drawbacks to the use of expansion as a metric, and explain the extent to which they can be overcome.

Difficulty of determining the expansion coefficient. Determining the expansion coefficient of a given graph is known to be co-NP-complete [3], and so testing KPSs for their expansion coefficient is not an easy task. Additionally, even if the expansion coefficient of the key graph is known, the expansion of the intersection graph will not be known a priori if the communication graph is modelled as a random graph.

Nevertheless, a method for *estimating* the expansion coefficient using the eigenvalues of the incidence matrix of the graph is given in [9], which could be used in the comparison of KPSs. Indeed, if it is possible to determine the locations of the nodes after deployment, for example using an online base station or GPS, it may be feasible to construct the intersection graph and therefore estimate its expansion coefficient, once the WSN has been deployed. This is likely to be relevant if post-deployment key management protocols are available such as key refreshing [2] or key redistribution [10], for which it could be useful to know as much as possible about the vulnerability of the WSN. Some key management protocols are able to provide targeted improvements to specific weak areas of the network, and we explain below how best to identify such weaknesses.

Limitations of the expansion coefficient. We note that the expansion coefficient alone does not claim to fully describe the structure of the graph, giving only a 'worst case' assessment. That is, the value of ϵ only reflects the weakest point of the graph and tells us nothing about the structure of the graph elsewhere.

For example, consider an intersection graph on n nodes which is effectively partitioned into two sets: a set of $n - 1$ nodes with high expansion, and a final node which is disconnected from the rest of the graph, as demonstrated in Fig. 5(b). We would find that $\epsilon = 0$, and we would suspect that the graph is less than desirable for WSN applications.

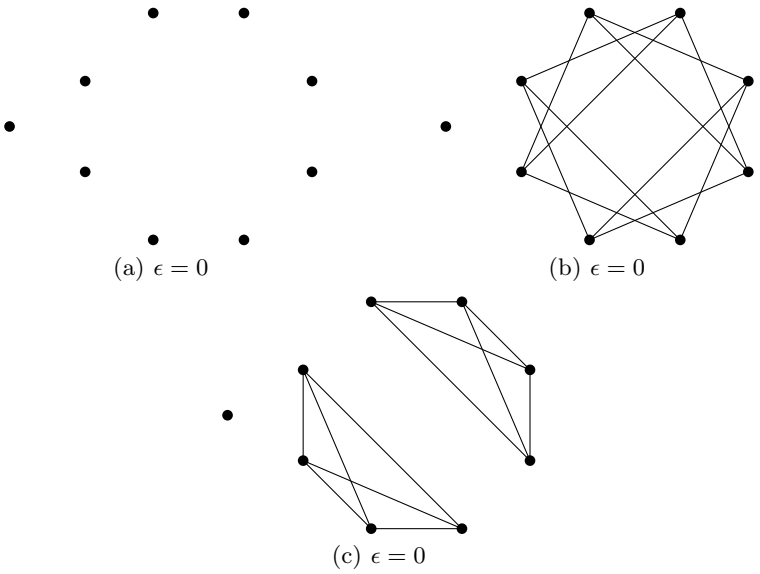


Fig. 5. Distinguishing between cases where $\epsilon = 0$

However, particularly in a network of thousands of nodes, the disconnection of one is unlikely to be severely detrimental to the network; indeed, loss of some nodes due to poor positioning or battery failure may be expected. Knowing only that $\epsilon = 0$ does not distinguish between the following cases:

1. the graph is completely disconnected (Fig. 5(a))
2. a single node is disconnected from the rest of the graph, which otherwise has good expansion (Fig. 5(b))
3. the disconnected graph is a union of smaller graphs, some with good expansion (Fig. 5(c))

If an intersection graph falls into Case 1 then it is likely that the key graph has low connectivity, ie. a low value of Pr_1. However, for the same values of network size, key storage, connectivity and resilience, knowing only that $\epsilon = 0$ in the

intersection graph cannot distinguish between the Cases 2 and 3, though Case 2 is likely to be much better for WSN applications.

Therefore, we suggest some graph-theoretic tools which also serve as indicators of whether the structure of a graph is suitable for WSNs. These may be used alone or in conjunction with (an estimate of) the expansion coefficient in order to analyse a proposed KPS, and where possible to analyse the resulting intersection graph.

Components. We note that to distinguish between the cases in Fig. 5 it is relevant to know the number of *components*. A *component* of a graph is a connected subgraph containing the *maximal* number of edges [8], that is, a subset S of one or more vertices of the graph, where the vertices of S are connected but $E(S, \overline{S}) = \emptyset$. Hence Fig. 5(a) has nine components, Fig. 5(b) has two, and Fig. 5(c) has three. For WSN applications where data must be routed throughout the network, it is desirable to minimise the number of components.

Unlike finding the expansion coefficient of a graph, calculating the number of components can be done in linear time using depth-first search, as described in [15]. The *global connectivity* of a graph is the number of nodes in its largest component divided by the total number of nodes. We wish the global connectivity to be as close to one as possible.

Cut-edges. A *cut-edge* (also known as a *bridge*) is an edge whose deletion increases the number of components. Equivalently, an edge is a cut-edge if it is not contained in any cycle of the graph. This is illustrated in Fig. 4, where the edge (e, f) is a cut-edge.

As we have seen, cut-edges in the intersection graph for a WSN are undesirable because they can cause bottlenecks, increase communication burdens on the nodes at their endpoints, and create weak points of the network where a small fault or compromise by an adversary creates a lot of damage. Therefore, one of the reasons why intersection graphs with high expansion are desirable for WSNs is because they are less likely to have cut-edges:

- If $\epsilon > \frac{1}{\lfloor \frac{|V|}{2} \rfloor}$ then we know that there is no cut-edge which, if removed, would separate the graph into two components, each of size $\frac{|V|}{2}$.
- If $\epsilon = \frac{1}{2}$ then it is possible that there are cut-edges which, if removed, would disconnect at most two nodes from the network
- If $\epsilon > 1$ then for all $S \subset V$ with $|S| \leq \frac{|V|}{2}$,

$$|E(S, \overline{S})| > |S| \geq 1 \ ,$$

and so there can be no cut-edges in the graph.

Determining whether a graph contains cut-edges can also be done by a linear time algorithm [22].

Cutpoints. There is also a related notion of *cutpoints* in graphs, for which we will need the following definitions from [8]. A *subgraph* of a graph $G(V, E)$ is a graph $G_S(V_S, E_S)$ in which $V_S \subset V$ and $E_S \subset E$. If V_S or E_S is a proper subset (that is, $V_S \neq V$ or $E_S \neq E$), then the subgraph is a *proper subgraph* of G. If V_S or E_S is empty, the subgraph is called the *null graph*.

In a connected graph G, if there exists a proper non-null subgraph G_S such that G_S and its complement have only one node n_i in common, then the node n_i is called a *cutpoint* of G. In an unconnected graph, a node is called a cutpoint if it is a cutpoint of one of its components. If G has no self-loops, then a cutpoint is a node whose removal increases the number of components by at least one. We see then that in Fig. 4(a), nodes e and f are cutpoints, and that graphs with good expansion will have few cutpoints.

If a graph contains no cutpoints it is said to be *nonseparable* or *biconnected*, which again is clearly desirable for an intersection graph representing a WSN. The website [1] gives examples of Java algorithms which find the nonseparable components of given graphs and can even add edges to make graphs nonseparable.

These tools are just a few of the simple, effective ways to analyse an intersection graph of a deployed WSN, and to make intelligent improvements to the structure of the graph wherever the post-deployment key management protocols allow.

7 Conclusion

We have shown that if we fix levels of key storage, network size, connectivity and resilience, then the larger the value of the expansion coefficient ϵ in the intersection graph, the better suited it will be for WSNs. This is because graphs with good expansion are well connected with low diameter and do not have the vulnerabilities of cut-edges and cutpoints. We have shown that the expansion coefficient of the product graph is not a relevant metric, but that it is the intersection graph where good expansion is desired.

In a setting where there is control over the communication graph, the expansion of the intersection graph should be an important consideration in the design of the key graph. If there is no control over the communication graph, then after choosing levels of network size, key storage, connectivity and resilience, the best choice of KPS is the one with the highest expansion, since it will maximise the probability of achieving good expansion in the intersection graph.

We have shown that KPSs based on expander graph constructions are able to produce key graphs with high expansion for a given network size and key storage, and use unique pairwise keys to give perfect resilience. However, many existing KPSs are able to achieve better expansion for the same key storage and network size, at the cost of lower resilience.

Finally, we have suggested that expansion is an important metric for comparing KPSs proposed for WSNs, and a useful parameter for analysing intersection graphs after deployment in order to improve weak parts of the network. Determining the expansion of a graph is co-NP-complete and gives only a worst-case

assessment of the graph. Therefore we have proposed the use of linear time algorithms to estimate the expansion, and introduced related graph-theoretic properties which could be used to analyse the key and intersection graphs of WSNs.

References

1. Analyzing Graphs, `http://docs.yworks.com/yfiles/doc/developers-guide/analysis.html`
2. Blackburn, S.R., Martin, K.M., Paterson, M.B., Stinson, D.R.: Key Refreshing in Wireless Sensor Networks. In: Safavi-Naini, R. (ed.) ICITS 2008. LNCS, vol. 5155, pp. 156–170. Springer, Heidelberg (2008)
3. Blum, M., Karp, R.M., Vornberger, O., Papadimitriu, C.H., Yannakakis, M.: The Complexity of Testing whether a Graph is a Superconcentrator. Information Processing Letters 13(4-5), 164–167 (1981)
4. Camtepe, S.A., Yener, B.: Key Distribution Mechanisms for Wireless Sensor Networks: a Survey. Rensselaer Polytechnic Institute, Computer Science Department, Tech. Rep. TR-05-07 (2005)
5. Camtepe, S.A., Yener, B., Yung, M.: Expander Graph Based Key Distribution Mechanisms in Wireless Sensor Networks. In: ICC 2006, IEEE International Conference on Communications, pp. 2262–2267 (2006)
6. Chan, H., Perrig, A., Song, D.: Random Key Predistribution Schemes for Sensor Networks. In: SP 2003: Proceedings of the 2003 IEEE Symposium on Security and Privacy, pp. 197–213. IEEE Computer Society, Washington, DC (2003)
7. Chen, C.-Y., Chao, H.-C.: A Survey of Key Distribution in Wireless Sensor Networks. In: Security and Communication Networks (2011)
8. Chen, W.-K.: Applied Graph Theory. University of Virginia, North-Holland (1971)
9. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society, California State University, Fresno (1994)
10. Cichoń, J., Gołębiewski, Z., Kutyłowski, M.: From Key Predistribution to Key Redistribution. In: Scheideler, C. (ed.) ALGOSENSORS 2010. LNCS, vol. 6451, pp. 92–104. Springer, Heidelberg (2010)
11. Erdös, P., Rényi, A.: On the Evolution of Random Graphs. Publication of the Mathematical Institute of the Hungarian Academy of Sciences, 17–61 (1960)
12. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: CCS 2002: Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 41–47. ACM, New York (2002)
13. Ghosh, S.K.: On Optimality of Key Pre-distribution Schemes for Distributed Sensor Networks. In: Buttyán, L., Gligor, V.D., Westhoff, D. (eds.) ESAS 2006. LNCS, vol. 4357, pp. 121–135. Springer, Heidelberg (2006)
14. Hoory, S., Linial, N., Wigderson, A.: Expander graphs and their applications. Bulletin of the American Mathematical Society 43(04), 439–562 (2006)
15. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. Commun. ACM 16(6), 372–378 (1973)
16. Kleinberg, J., Rubinfeld, R.: Short Paths in Expander Graphs. In: Annual Symposium on Foundations of Computer Science, vol. 37, pp. 86–95 (1996)
17. 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. Springer, Heidelberg (2004)

18. Martin, K.M., Paterson, M.B.: An Application-Oriented Framework for Wireless Sensor Network Key Establishment. Electronic Notes in Theoretical Computer Science 192(2), 31–41 (2008)
19. Paterson, M.B., Stinson, D.R.: A unified approach to combinatorial key predistribution schemes for sensor networks. Cryptology ePrint Archive, Report 076 (2011)
20. Reingold, O., Vadhan, S., Wigderson, A.: Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp. 3–28. IEEE Computer Society, Washington, DC (2000)
21. Shafiei, H., Mehdizadeh, A., Khonsari, A., Ould-Khaoua, M.: A Combinatorial Approach for Key-Distribution in Wireless Sensor Networks. In: Global Telecommunications Conference, IEEE GLOBECOM 2008, pp. 1–5. IEEE (2008)
22. Tarjan, R.: A Note on Finding the Bridges of a Graph. Information Processing Letters, 160–161 (1974)
23. Xiao, Y., Rayi, V.K., Sun, B., Du, X., Hu, F., Galloway, M.: A survey of key management schemes in wireless sensor networks. Computer Communications 30(11-12), 2314–2341 (2007)