VRIJE
UNIVERSITEIT
BRUSSEL

FACULTEIT WETENSCHAPPEN
EN BIO-INGENIEURSWETENSCHAPPEN
Vakgroep Wiskunde

Expander Graphs and Key Predistribution Schemes

JEROEN OOGE

2de Master wiskunde

Promotor: Prof. Dr. Philippe Cara

Eindwerk ingediend voor het behalen van de graad van Master in de wiskunde met afstudeerrichting ‘Fundamentele wiskunde’.

ACADEMIEJAAR 2016-2017

Contents

Abstract and contributions **iii**

Samenvatting en bijdragen **iv**

1 Introduction **1**

2 Basic concepts **3**

2.1 Wireless sensor networks 3

2.2 Key predistribution schemes 4

2.3 Graphs 6

2.4 Multigraphs and hypergraphs 7

2.5 Cayley graphs 8

2.6 Combinatorial designs 9

3 Expander graphs **11**

3.1 The expansion coefficient 11

3.2 The adjacency operator 13

3.3 The Laplacian operator 17

3.4 The Rayleigh–Ritz theorem 19

3.5 The Alon–Boppana theorem 23

4 Zig-zag products **28**

4.1 Definition of the zig-zag product 28

4.2 Eigenvalues of zig-zag products 32

4.3 An explicit expander family 36

5 Key predistribution schemes **40**

5.1 Blom key predistribution scheme 40

5.2 Key distribution patterns 46

5.3 Random key predistribution schemes 50

5.4 Combinatorial designs 54

6 Conclusions **60**

A Implementations in Matlab **62**

A.1 Zig-zag products 63

A.2 Blom key predistribution scheme 67

A.3 Eschenauer–Gligor key predistribution scheme 70

Bibliography **73**

Index **76**

Abstract and contributions

A key predistribution scheme is a method for allocating symmetric cryptographic keys to devices in a network. These devices are often part of a wireless sensor network: they are scattered over a large area to perform basic tasks such as monitoring and data gathering. Since the devices' battery power and computational capacities are limited, schemes should be carefully designed to make a trade-off between key storage, connectivity and the network's resilience.

We discuss several influential key predistribution schemes, shed light on their strengths and weaknesses, and implement them in Matlab. Before we do so, however, a deeper mathematical understanding is required of what characteristics 'good' networks have. The introduction of the expansion coefficient as a graph invariant leads to surprisingly deep connections with lots of different branches in mathematics.

Due to the relative newness of the mathematical research into expander graphs and key predistribution, much of the literature is rather incoherent and few standard references are available. The main challenge was therefore to unite the many existing books and papers on these topics; to adopt a suitable notation and nomenclature for the whole document; to find a seemingly optimal order for presenting all different concepts; and to improve certain proofs.

Chapters 3 and 4 are basically an improvement and elaboration of [22], which is very good concerning content, but which tends to confuse the reader with many lemmas and sometimes unnecessarily complicated notation. Especially the introduction of zig-zag products required quite some work: it took effort to clarify the product's definition, to overcome the horrible notation in [35] and to find illuminating examples. By the way, all examples in this document are own work or adaptations of the literature. Also, the proof of Lemma 4.2.6 was rewritten in terms of adjacency operators, instead of vector decomposition, which dates from the original construction with tensor products. Chapter 5 is a melting pot of many references, attempting to briefly discuss the historically most significant key predistribution schemes. In Appendix A, the zig-zag product and several key predistribution schemes are implemented in Matlab, which demanded some non-trivial tricks.

Samenvatting en bijdragen

Een sleutel-predistributieschema *(vrije vertaling)* is een methode om symmetrische cryptografische sleutels toe te wijzen aan toestellen in een netwerk. Die toestellen maken vaak deel uit van een draadloos sensorennetwerk *(vrije vertaling)*: ze worden verspreid over een groot gebied om basistaken uit te voeren zoals het controleren van de omgeving en het verzamelen van gegevens. Aangezien de batterijduur en de computationele capaciteiten van de toestellen beperkt zijn, moeten schema's nauwkeurig ontworpen worden om een compromis te vinden tussen de opslag van sleutels, connectiviteit en de veiligheid van het netwerk.

We bespreken verscheidene invloedrijke sleutel-predistributieschema's, belichten hun sterktes en zwaktes, en implementeren ze in Matlab. Daarvoor hebben we echter eerst een beter wiskundig begrip nodig van wat 'goede' netwerken karakteriseert. De introductie van de expansiecoëfficiënt als grafinvariant leidt tot verrassend diepe verbanden met veel verschillende takken van de wiskunde.

Vanwege de relatieve nieuwheid van het wiskundige onderzoek naar expansiegraffen *(vrije vertaling)* en sleutelpredistributie is de literatuur redelijk onsamenhangend en zijn er weinig standaardreferenties beschikbaar. De grootste uitdaging was daarom de vele boeken en papers over die onderwerpen tot een geheel te maken; een gepaste notatie en een gepast jargon te gebruiken in het hele document; een schijnbaar optimale volgorde te vinden om alle verschillende concepten te presenteren; en bepaalde bewijzen te verbeteren.

Hoofdstukken 3 en 4 zijn in se een verbetering en uitbreiding van [22], dat inhoudelijk prima is, maar dat geneigd is de lezer te verwarren met veel lemma's en soms onnodig ingewikkelde notatie. Voornamelijk voor de invoering van zig-zagproducten was redelijk wat werk nodig: het vergde inspanning om de definitie van het product te verduidelijken, de vreselijke notatie in [35] te verteren en verhelderende voorbeelden te vinden. Alle voorbeelden in dit document zijn trouwens eigen werk of adaptaties van de literatuur. Ook het bewijs van Lemma 4.2.6 werd zelf herschreven in termen van adjacentie-operatoren, in plaats van vectordecompositie, die stamt van de oorspronkelijke constructie aan de hand van tensorproducten. Hoofdstuk 5 is een smeltkroes van vele referenties, als poging om kort de historisch meest significante sleutel-predistributieschema's te behandelen. In Appendix A worden het zig-zagproduct en enkele sleutel-predistributieschema's geïmplementeerd in Matlab, wat een aantal niet-triviale truucjes vergde.

Chapter 1

Introduction

A vital component of any cryptosystem is key establishment, which governs the distribution of cryptographic keys in a network. This can be particularly challenging in symmetric cryptosystems, where all parties — called **nodes** — should establish appropriate keys to securely communicate with one another. For symmetric key establishment, a trusted authority generates these keys and then assigns them to the nodes. We call this process **key distribution**.

In some applications, the trusted authority is online and available all the time to provide keys 'on the fly', that is, when required. However, this is often not the case: it is no longer possible to interact with nodes after they have been deployed in the environment. The only realistic alternative in those situations is for the trusted authority to distribute keys before deployment during a secure initialisation process, the so-called **key predistribution**. Afterwards, the trusted authority plays no further role in key establishment, so two nodes who require a common key must derive one from their predistributed keys. A well thought-out scheme to allocate keys is thus crucial; we call such a strategy a **key predistribution scheme** (KPS). Over the past few decades, dozens of KPSs have been suggested in the literature and all of them are forced to make trade-offs between key storage, connectivity and the network's resilience to attacks from adversaries.

The design of KPSs of course relies on our understanding of what features a 'good' network must have. Lots of connections certainly speed up communication, but they also imply higher storage and computational costs, which is disadvantageous in practice. In order to find sparse yet reliable networks, the expansion coefficient was introduced. Roughly speaking, this invariant measures how well parts of a network are connected to the rest of the network. Surprisingly, the concept of expansion leads to rich mathematical theories and connects many different branches of mathematics.

In Chapter 2, we frame our discussion around wireless sensor networks, which is nowadays the context for almost all of the related research. We discuss the three aforementioned parameters of KPSs in more detail and recall some basic mathematical concepts. In Chapter 3, we elaborately investigate the relationship between the expansion coefficient and the second-largest eigenvalue, and prove some deep mathematical results. Chapter 4 contains the construction of an explicit expander graph, based on the nicely named zig-zag product. Next, we present in Chapter 5 several historical KPS milestones, which form the baseline for many other KPSs, and find lower bounds for the expansion coefficient in the resulting networks. To conclude, we implement the zig-zag product and some KPSs in Matlab in Appendix A.

Chapter 2

Basic concepts

For a better understanding of key predistribution schemes, we frame our discussion around wireless sensor networks and then discuss the three conflicting parameters in the design of schemes. Next, we briefly recall some mathematical concepts that will play an important role throughout the rest of this document. We mainly used [7, 19, 20, 22] as references.

2.1 Wireless sensor networks

Due to the development of small wireless technologies, some communication networks are currently undergoing a major architectural shift: instead of centralised, wired networks that consist of a few powerful devices, networks nowadays tend to consist of many resource-constrained devices, which can be distributed over large areas and communicate wirelessly. Wireless sensor networks are a very good example of this change, and are just one class of emerging technologies for which a combination of key predistribution and symmetric cryptography is well-suited for communication.

A **wireless sensor network** (WSN) is a collection of small, battery powered sensor nodes, which collectively monitor and gather data about phenomena of interest. There are plenty of application examples: health monitoring, seismic data gathering, forest fire detection, gathering of ecological data, military intelligence gathering, etc. The number of nodes may vary from dozens to several thousands, depending on the application. WSNs are discussed in detail in [10, 29, 31, 33, 36, 37, 41]; the following paragraphs summarize the aspects that are relevant for our purposes.

WSNs are especially useful in hostile environments that are not accessible to humans, such as volcanic craters, disaster areas, enemy soil during wartime, etc. Indeed, it is for example possible to release the nodes from the air. In this scenario, the nodes' location cannot be predicted before deployment. Other situations allow partial or even full control over the network's topology, which leads to specialized efficient KPSs, such as KPSs for grid-based WSNs in [3]. We will always assume an uncontrolled network topology and static nodes for the design of KPSs.

After deployment, nodes use a shared key discovery protocol to identify the nodes with which they share a key and can thus establish a secure communication link. There are many known such protocols; the simplest way is for nodes to broadcast in plain text a list of identifiers of the keys they store. Our previous supposition immediately motivates the study of KPSs for incomplete network topologies, that is, networks wherein at least one node does not share keys with all other nodes. It is namely possible that two nodes are located too far apart and can thus not communicate directly, in spite of sharing a key. For non-direct communication between a pair of nodes, WSNs rely on **hopping**: nodes may successively pass data from the sender to nodes within range, until the receiver has been reached.

Because of their compact nature, nodes have limited memory, which constrains the number of storable keys; nodes have batteries that may quickly be drained by many computations and communications; and nodes may easily be compromised. The second complication explains why we restrict ourselves to the study of symmetric cryptography: public key cryptography typically requires more computational power. This approach seems future-proof, because it is plausible that more constrained sensor technology will be developed as soon as public key cryptography becomes practical for WSNs. Battery saving is also desired for the sake of a long-lasting network. Indeed, in many applications nodes cannot be recharged and simply expire when they are out of power. To conclude, we assume homogeneous nodes, i.e. all nodes have the same capabilities and restrictions.

2.2 Key predistribution schemes

The design of KPSs typically takes into account three conflicting parameters: the key storage for each node, which should be minimised; the connectivity between different nodes, which should be maximised; and the resilience of the network, which should be maximised as well. Since these parameters play a crucial role throughout the rest of our discussion, we will immediately explain them in more detail.

Key storage. This is the number of keys that each node is required to store, usually denoted by the constant k. Due to their limited storage and computational capacities, nodes are often unable to support public key cryptography. Therefore, any security must be provided by symmetric key cryptography, which requires less memory and is less expensive computationally.

Connectivity. This is the sharing of keys between nodes in the network: in order to establish a secure connection, nodes are required to have at least $q \geq 1$ keys in common. In some schemes, nodes must share multiple keys before they are allowed to communicate, i.e. $q > 1$. We will present such a KPS in Section 5.3. However, most KPSs allow communication between nodes when $q = 1$. We denote Pr_1 for the probability that a pair of randomly selected nodes is connected. Obviously, we want Pr_1 to be as close to 1 as possible.

Resilience. This is a measure of how often keys are re-used throughout the network, or equivalently, the network's ability to withstand attacks from an adversary. We assume a continuously listening adversary, who can intercept any communication across the network and who can thus **compromise nodes**, that is, learning the keys they store. The resilience is measured with the parameter fail_s where $1 \leq s \leq n - 2$ and n is the number of nodes in the network: if an adversary has compromised s nodes, then fail_s is equal to the probability that the link between a pair of uncompromised nodes is compromised. Of course, high resilience corresponds to a low value of fail_s. If $\mathrm{fail}_s = 0$ for all $1 \leq s \leq n - 2$, then the network has **perfect resilience**.

In order to illustrate the trade-offs between key storage, connectivity and resilience, we give some trivial examples of KPSs.

Example 1. Assigning the same key K to every node results in minimal key storage and ensures a secure connection between any pair of nodes, i.e. $\mathrm{Pr}_1 = 1$ for all pairs of nodes. However, there is also minimal resilience against an adversary, since the compromise of a single node would reveal the key K, rendering all links insecure. Formally, $\mathrm{fail}_s = 1$ for all $1 \leq s \leq n - 2$.

Example 2. Predistributing a unique key K_{ij} to each pair of nodes $\{N_i, N_j\}$, that is $K_{ij} \neq K_{lm}$ if $\{i, j\} \neq \{l, m\}$ for $1 \leq i, j, l, m \leq n$, results in perfect resilience and maximal connectivity. This scheme is known as the **complete pairwise KPS**. Unfortunately, such a KPS requires all nodes to store $n-1$ keys and needs $n(n-1)/2$ different keys in total, which is infeasible for large n.

Example 3. If we assign to each node its own unique key, then we obtain an absurd network with minimal key storage and maximal resilience, but no connectivity at all, i.e. $\mathrm{Pr}_1 = 0$ for all pairs of nodes.

The above examples show that it is trivial to optimise any two of the three parameters. However, these schemes are inappropriate for almost all real-life applications, so we are interesed in KPSs that find a trade-off between the three metrics. Many proposals for such KPSs can be found in the literature and we will present in Chapter 5 four of them, which greatly impacted the research field.

2.3 Graphs

Intuitively speaking, a graph consists of points, and lines that join one point to another. For example, the points can represent devices in a network and the lines can correspond to physical links between pairs of these devices. When drawing a graph, the relative positions of points and the shapes of lines does not matter; the only important information is whether or not two points are connected by a line. Therefore, the same graph can give rise to many dissimilar drawings.

Most of the time, we will only consider graphs that are unweighted, undirected and do not contain loops or multiple edges. These terms respectively indicate that the points and lines are not assigned any weights, the lines are not directed from one point to another, there are no lines from a point to itself, and there is at most one line between two points. In the literature, this type of graph is sometimes referred to as a 'simple graph', because there are some useful generalizations (see Section 2.4). The formal definition is given as follows:

Definition 2.3.1. *A* **graph** *is a pair (V, E), where V is a finite non-empty set and $E \subseteq \{ \{v, w\} \mid v, w \in V \text{ and } v \neq w \}$ is a set of unordered pairs.*

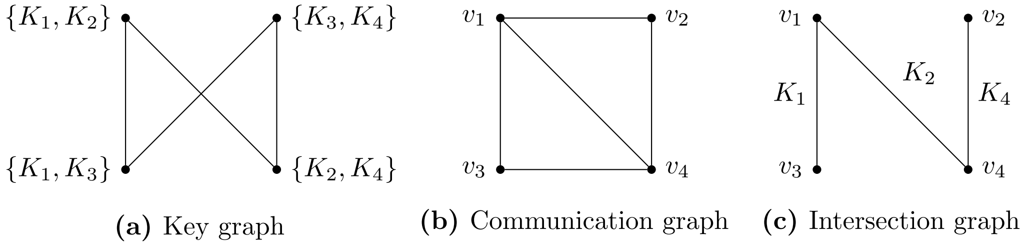
The elements of V and E are respectively called **vertices** (the 'points') and **edges** (the 'lines'). For a graph G, we write $V(G)$ and $E(G)$ to respectively refer to the vertex set and the edge set of G. If there exists an edge between two vertices v and w, i.e. $\{v, w\} \in E$, then we say that these vertices are **adjacent** and that the edge $\{v, w\}$ is **incident** with its **endpoints** v and w. Formalising the aforementioned remark about the graphical representation of graphs leads to the concept of an isomorphism. We say that two graphs G and H are **isomorphic** if there exists a bijection $\varphi\colon V(G) \to V(H)$ such that $\{v, w\} \in E(G)$ if and only if $\{\varphi(v), \varphi(w)\} \in E(H)$. An isomorphism can also be interpreted as a relabeling of the vertices in $V(G)$ and it is common practice to treat isomorphic graphs as if they were equal.

We will now introduce some more general terminology and notation for graphs. Given subsets $X, Y \subseteq V$, the set of edges between X and Y is denoted by

$$E(X, Y) = \{ \{x, y\} \in E \mid x \in X,\, y \in Y \}.$$

Notice that $E(X, Y) = E(Y, X)$, because all edges in a graph are undirected. Also, the **complement** X^{c} of X consists of the vertices that are not in X, i.e. $X^{\mathrm{c}} = V \setminus X$. An ordered set of consecutive distinct edges $(\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\})$ is called a **path** of **length** $k - 1$. A **cycle** is a path with distinct vertices $v_1, \ldots, v_{k-1}$ that begins and ends at the same vertex, i.e. $v_1 = v_k$. 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 any two distinct vertices. Another special type of graph is a **bipartite** graph for which there exists a bipartition (V_1, V_2) of V, i.e. there are disjoint subsets $V_1, V_2 \subset V$ such that $V = V_1 \cup V_2$, and every edge in E is incident with a vertex in V_1 and a vertex in V_2. The **distance** between vertices v and w, denoted by $d(v, w)$, is the length of the shortest path from v to w. The **diameter** of a graph G is given by $\operatorname{diam}(G) = \max_{v,w \in V} d(v, w)$. The **degree** of a vertex $v \in V$ is equal to the number of edges incident with v, and we write it as $\deg(v)$. If all vertices of a graph have the same degree r, we call the graph r**-regular**.

WSNs and KPSs can be interpreted as graphs if we conceive the nodes $N_1, \ldots, N_n$ as vertices $v_1, \ldots, v_n$ respectively and the connections between the nodes as edges. To be precise, we need to distinguish between connection before and after deployment of the nodes, and consider the corresponding graphs. Let us denote $V = \{v_1, \ldots, v_n\}$. Firstly, we have the **key graph** (V, E_1) where $\{v_i, v_j\} \in E_1$ if N_i and N_j share at least q common keys. Secondly, we define the **communication graph** (V, E_2) where $\{v_i, v_j\} \in E_2$ if the nodes N_i and N_j are physically within communication range. Two nodes N_i and N_j can communicate securely in a WSN if $\{v_i, v_j\} \in E_1 \cap E_2$, that is, if they are adjacent in the **intersection graph** $(V, E_1 \cap E_2)$. An example of these graphs for $q = 1$ is given in Fig. 2.1.



(a) Key graph (b) Communication graph (c) Intersection graph

Figure 2.1: Corresponding key, communication and intersection graphs.

As assumed in Section 2.1, we do not have any control over the positioning of the nodes. In other words, the communication graph is a random graph. The only way to affect the intersection graph is thus by carefully designing the key graph.

2.4 Multigraphs and hypergraphs

As mentioned earlier, we can generalise the notion of a graph to one that allows loops and multiple edges, and for which the terminology introduced in Section 2.3 is preserved. We start with a generalization of ordinary sets.

Definition 2.4.1. *A* **multiset** *is a collection of objects wherein each object may appear several times. The number of appearances is called the object's* **multiplicity***.*

All familiar notation for sets is carried over to multisets. We can for example write $a \in \{a, b, b, c, c, c\}$. In this particular multiset, the elements a, b and c respectively have multiplicities 1, 2 and 3. An ordered multiset of consecutive edges $(\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{(k-1)}, v_k\})$ is called a **walk** of length $k - 1$.

Definition 2.4.2. *A* **multigraph** *is a pair (V, E), where V is a finite non-empty set and $E \subseteq \{\, \{v, w\} \mid v, w \in V \,\}$ is a multiset of multisets of size 2.*

Comparing this definition to Definition 2.3.1, notice that we dropped the condition $v \neq w$ in the multiset that contains E. An edge of the form $\{v, v\}$ for $v \in V$ is called a **loop**. There is also a second generalisation of graphs: we may no longer restrict the number of vertices that can be connected by an edge to 2.

Definition 2.4.3. *A* **hypergraph** *is a pair (V, E) where V is a finite non-empty set and E consists of subsets $A \subseteq V$ with $2 \leq |A|$.*

The elements of E are called **hyperedges**. If every hyperedge contains r vertices, we say that the hypergraph is r**-uniform**. A graph can thus be thought of as a 2-uniform hypergraph.

2.5 Cayley graphs

Given a group G and a special kind of multisubset S of G, we can construct highly symmetrical multigraphs from which one can derive properties of the group. We start with introducing the condition that S should meet.

Definition 2.5.1. *A multisubset S of a group G is called* **symmetric** *if for any $g \in S$ with multiplicity m, also $g^{-1} \in S$ with multiplicity m. We write $S \subset_s G$.*

Definition 2.5.2. *Let G be a finite group and $S \subset_s G$. We define $\mathrm{Cay}(G, S)$, the* **Cayley graph** *of G with respect to S, as follows: the vertices in $V\big(\mathrm{Cay}(G, S)\big)$ are the elements of G and the multiplicity of an edge $\{g, h\}$ in $E\big(\mathrm{Cay}(G, S)\big)$ is equal to the multiplicity of $h^{-1}g$ in S.*

Stated differently, two vertices $g, h \in G$ are adjacent if and only if $h^{-1}g \in S$, i.e.

there exists a $t \in S$ such that $g = ht$. We need S to be symmetric, because the adjacency of two elements $g, h \in G$ implies both $g = ht$ and $h = gu$ for certain $t, u \in S$, which results in $t^{-1} = u \in S$. If we relax this condition, then we end up with directed Cayley graphs. Note that an ordinary set S results in a Cayley graph without multiple edges.

Proposition 2.5.3. *Let G be a finite group and $S \subset_s G$. Then,* $\mathrm{Cay}(G, S)$ *is $|S|$-regular.*

Proof. We write $S = \{t_1, \ldots, t_m\}$ and pick an element $g \in G$. The vertices adjacent to g are $gt_1, \ldots, gt_m$, counted with multiplicity. Therefore, $\deg(g) = m = |S|$. □

2.6 Combinatorial designs

For any set X, we denote $\mathscr{P}(X)$ for the power set of X, which is the set of all subsets of X.

Definition 2.6.1. *A* **set system** $(X, \mathscr{B})$ *consists of a set X and $\mathscr{B} \subseteq \mathscr{P}(X)$. The elements of X are called* **points** *and the elements of $\mathscr{B}$ are called* **blocks**.

Since we defined blocks as ordinary sets, each point occurs at most once in each block. The degree of a point $x \in X$ is the number of blocks that contain x. We say that a set system is r**-regular** or regular of degree r if every point has degree r. The size of the largest block is the **degree** of the set system and if all blocks have the same size k, then the set system is said to be k**-uniform** or uniform of rank k.

A **combinatorial design** (design for short) is a general term used to describe a set system with particular conditions on regularity, uniformity and block intersection. We usually add a prefix to the word 'design' to specify these properties.

Definition 2.6.2. *A set system $(X, \mathscr{B})$ where $|X| = n$ is a* t**-**(n, k, λ) **design** *if it is uniform of rank k and every set of t points is contained in exactly λ blocks.*

It is straightforward to represent a set system or design $(X, \mathscr{B})$ as a graph G: let $V(G) = \mathscr{B}$ and add an edge between two blocks B_i and B_j if $B_i \cap B_j \neq \emptyset$. This idea is encapsulated in the following concept.

Definition 2.6.3. *Let* $(X, \mathscr{B})$ *be a set system where* $X = \{x_1, \dots, x_m\}$ *and* $\mathscr{B} = \{B_1, \dots, B_n\}$*. The* **incidence matrix** *of* $(X, \mathscr{B})$ *is an* $m \times n$*-matrix* A *whose entries are given by*

$$a_{ij} = \begin{cases} 1 & \textit{if } x_i \in B_j, \\ 0 & \textit{otherwise.} \end{cases}$$

The rows and columns of an incidence matrix thus respectively represent points and blocks.

Chapter 3

Expander graphs

In this chapter, we discuss two very important graph invariants: the expansion coefficient and the second-largest eigenvalue. Both numbers are intimately related by a fundamental inequality that we will prove with standard linear algebra techniques. All results are based on [11, 22]. For further information on linear algebra, we refer to [8].

3.1 The expansion coefficient

Definition 3.1.1. *The* **expansion coefficient** *of a graph $G = (V, E)$ is defined as*

$$\varepsilon(G) = \min\left\{ \frac{|E(S, S^c)|}{|S|} \,\middle|\, S \subset V,\, 0 < |S| \leq \frac{|V|}{2} \right\}.$$

For brevity, we will write ε instead of $\varepsilon(G)$ if the graph G is clear from the context. Lots of synonyms for the expansion coefficient are being used in the literature: the isoperimetric constant or number, the edge expansion ratio, the Cheeger constant, the conductance... A large value of ε is desirable for many network applications, which can be seen by the following observations:

(1) If $\varepsilon = 0$, then there exists a subset $S \subset V$ such that $E(S, S^c) = \emptyset$. This is equivalent to saying that the graph is disconnected.

(2) A small ε, particularly $\varepsilon < 1$, indicates that at least one set of vertices is connected to the rest of the graph by relatively few edges. In a network, this can lead to vulnerabilities such as communication bottlenecks; uneven burdens on nodes, creating uneven battery drainage; a risk of being disconnected more easily by an adversary; and longer average path lengths between unconnected nodes.

(3) If ε is larger, particularly if $\varepsilon \geq 1$, then there is no 'easy' way to disconnect large sets of nodes, and there is a more even spread of communication burdens, battery usage and data flow. Roughly speaking, the larger ε is, the faster and more reliable the network is.

A graph with a 'large' value of ε is often said to have 'good expansion' and is informally referred to as an **expander graph**. Let's compute the expansion coefficient for two concrete types of graphs.

Example 4. In general, we denote the complete graph with $n \geq 1$ vertices by K_n. Considering a fixed $K_n = (V, E)$ where $n \geq 2$, we get for any subset $S \subset V$ that

$$\frac{|E(S, S^{\mathrm{c}})|}{|S|} = \frac{|S|(n - |S|)}{|S|} = n - |S|,$$

which implies $\varepsilon(K_n) = n/2$ if n is even and $\varepsilon(K_n) = (n+1)/2$ if n is odd. Observe that $\varepsilon(K_n)$ becomes bigger as K_n grows in size. This matches well with our intuition that K_n is a good communication network: all vertices are pairwise adjacent.

Example 5. We define a **cycle graph** C_n as a graph that consists of $n \geq 1$ vertices which form a cycle. For any fixed $n \geq 3$, the graph $C_n = (V, E)$ is 2-regular and

$$\min\left\{ \frac{|E(S, S^{\mathrm{c}})|}{|S|} \;\middle|\; S \subset V,\ |S| = s \right\} = \frac{2}{s},$$

where the minimum occurs when S is connected. This implies $\varepsilon(C_n) = 4/n$ if n is even and $\varepsilon(C_n) = 4/(n-1)$ if n is odd. These expansion coefficients converge to 0 as $n \to \infty$, which suggests that cycle graphs become worse communication networks as they become larger. This behaviour is intuitively clear: the network performance greatly reduces when a single vertex is removed from the cycle graph, and the graph even becomes disconnected when a second non-adjacent vertex is eliminated.

Although complete graphs seem to be good communication networks, they contain far too many edges for practical means. We therefore restrict our attention to regular graphs of fixed degree. This leads to the following definition:

Definition 3.1.2. *Suppose $(G_n)_n$ is a sequence of r-regular graphs such that $|V(G_n)| \to \infty$ as $n \to \infty$. We say that $(G_n)_n$ is an* **expander family** *if there exists a real number $\alpha > 0$ such that $\varepsilon(G_n) \geq \alpha$ for all n.*

In other words, for any expander family the sequence of expansion coefficients must be **bounded away from zero**. The definition immediately implies that every graph in an expander family is connected. Note that the terms 'expander graph' and 'expander family' are in fact inaccurate and a bit misleading, because the actual objects of study are sequences of graphs. However, in order to be conform with the literature, we will use the traditional terminology.

Example 6. The calculations in Example 4 show that $\varepsilon(K_n) \geq 1$ for any $n \geq 2$. Although every K_n is $(n-1)$-regular, this degree is not fixed, so $(K_n)_{n\geq 2}$ is not an expander family. Example 5 implies that $(C_n)_{n\geq 3}$ is no expander family either.

Based on Example 6, we can ask ourselves whether expander families exist at all. In 1973, Mark Pinsker (see [34]) used a probabilistic argument to demonstrate that they do. That same year, Gregori Margulis came up with the first explicit construction in [26]. We will present another elementary explicit construction in Chapter 4.

Computing ε requires the investigation of a lot of subsets and these grow exponentially in number as the amount of vertices increases. It is thus often infeasible to explicitly determine ε for very large graphs. Therefore, we will seek lower and upper bounds for the expansion coefficient. Two trivial bounds for ε are the following:

Proposition 3.1.3. *For any graph (V, E) with $|V| \geq 2$, we have*

$$0 \leq \varepsilon \leq \min_{v\in V} \deg(v).$$

Proof. We have already mentioned that a graph is disconnected if and only if $\varepsilon = 0$. Also, ε cannot be strictly negative by definition, so $0 \leq \varepsilon$. For the upper bound, pick an arbitrary $v \in V$ and consider $S = \{v\}$. Since $|E(S, S^{\mathrm{c}})| = \deg(v)$ and $|S| = 1 \leq |V|/2$, the result follows from the definition of ε. □

3.2 The adjacency operator

Before introducing the adjacency operator, we provide a general framework in which we will work.

Definition 3.2.1. *For a finite set S, we denote $L^2(S) = \{f\colon S \to \mathbb{C}\}$.*

The set $L^2(S)$ can be turned into a complex vector space if addition and scalar multiplication are defined as expected for $f, g \in L^2(S)$, $\alpha \in \mathbb{C}$ and $x \in S$:

$$(f+g)(x) = f(x) + g(x) \quad \text{and} \quad (\alpha f)(x) = \alpha f(x).$$

We can also place an inner product and a norm on this space:

$$\langle f, g\rangle_2 = \sum_{x\in S} f(x)\overline{g(x)} \quad \text{and} \quad \|f\|_2 = \sqrt{\langle f, f\rangle_2} = \sqrt{\sum_{x\in S} |f(x)|^2}.$$

Note that $\|\cdot\|_2$ is the traditional L^2-norm for finite-dimensional vector spaces; hence the notation $L^2(S)$. We will often drop the subscript and simply write $\langle\cdot,\cdot\rangle$

and $\|\cdot\|$ when it is clear from the context that we are working in $L^2(S)$ for a certain S. We will usually deal with the space $L^2(V)$ where V is the vertex set of a graph. After ordering $V = \{v_1, \dots, v_n\}$, we can think of $f \in L^2(V)$ as a vector $\big(f(v_1), \dots, f(v_n)\big)^T \in \mathbb{C}^n$.

Definition 3.2.2. *The* **adjacency matrix** *of a graph with ordered vertex set $V = \{v_1, \dots, v_n\}$ is an $n \times n$ matrix A, where every entry a_{ij} on the i-th row and j-th column is equal to the number of edges that are incident to v_i and v_j.*

We also write a_{vw} for the number of edges that are incident to the vertices v and w, such that we may refer to an entry of A without having to order the vertices. We easily see that the adjacency matrix A of a graph is symmetric with real entries 0 and 1 only, and zeroes on the main diagonal, so it has n real eigenvalues $\lambda_1 \geq \dots \geq \lambda_n$.

Definition 3.2.3. *We define the* **eigenvalues of a graph** G *as the eigenvalues of its corresponding adjacency matrix and we will denote them as*

$$\lambda_1(G) \geq \dots \geq \lambda_n(G) \quad \textit{where } n = |V(G)|.$$

If we do not need to emphasize the considered graph, then we simply write $\lambda_1, \dots, \lambda_n$ for its eigenvalues in increasing order. We now show that the above definition is well-defined by proving that the eigenvalues of a graph do not depend on the ordering of the graph's vertices.

Proposition 3.2.4. *If A_1 and A_2 are two adjacency matrices of a graph (V, E) using different orderings of the vertices in V, then they have the same eigenvalues.*

Proof. Suppose that A_1 and A_2 use the orderings $\{v_1, \dots, v_n\}$ and $\{v_{\sigma(1)}, \dots, v_{\sigma(n)}\}$ respectively, where σ is a permutation of $\{1, \dots, n\}$. We then have

$$A_1 = PA_2P^T = PA_2P^{-1}$$

for the permutation matrix P associated to σ. For all numbers $\lambda \in \mathbb{R}$, we obtain

$$\det(A_2 - \lambda I) = \det(P)\det(A_2 - \lambda I)\det(P^{-1}) = \det(A_1 - \lambda I),$$

so we can conclude that A_1 and A_2 share the same eigenvalues λ. □

In the following proposition, we construct the smallest possible symmetric interval that contains all the eigenvalues of a regular graph.

Proposition 3.2.5. *Let G be an r-regular graph with n vertices. Then,*

(1) $\lambda_i \in [-r, r]$ *for all* $1 \leq i \leq n$*,*

(2) $\lambda_1 = r$ *with a corresponding eigenvector whose entries are all equal.*

Proof. *(1)* Since G is r-regular, the sum of the entries in each row and column of its adjacency matrix A is r. If $(x_1, \ldots, x_n)^T$ is the corresponding eigenvector for a certain eigenvalue λ_k where $1 \leq k \leq n$, then

$$\sum_{j=1}^{n} a_{ij} x_j = \lambda_k x_i \quad \text{for all } 1 \leq i \leq n. \tag{3.1}$$

Choosing the specific index i such that $x_i \neq 0$ is maximal, gives

$$|\lambda_k| \leq \sum_{j=1}^{n} |a_{ij}| \frac{|x_j|}{|x_i|} \leq \sum_{j=1}^{n} |a_{ij}| = r.$$

(2) This follows immediately from Eq. (3.1). □

Suppose that we have a graph with an ordered vertex set $V = \{v_1, \ldots, v_n\}$ and let A be the adjacency matrix given this ordering. For any $f \in L^2(V)$, we can immediately compute the matrix product

$$Af = \left(\sum_{i=1}^{n} a_{1i} f(v_i), \ldots, \sum_{i=1}^{n} a_{ni} f(v_i) \right)^T.$$

Looking carefully at this equation, we can actually think of A as a linear operator from $L^2(V)$ to $L^2(V)$, given by the formula in the following definition.

Definition 3.2.6. *The* **adjacency operator** *of a graph (V, E) with adjacency matrix A is the linear operator $A \colon L^2(V) \to L^2(V)$ that maps an f to the function defined by*

$$A(f)(v) = \sum_{w \in V} a_{vw} f(w).$$

As the reader can see, we will use the letter A for both the adjacency operator and the adjacency matrix. In order to become familiar with this new operator, we prove two more spectral characterisations for regular graphs.

Proposition 3.2.7. *Let $G = (V, E)$ be an r-regular graph. Then,*

(1) G is connected if and only if $\lambda_1 > \lambda_2$,

(2) G is bipartite if and only if $-r$ is an eigenvalue of G.

Proof. As usual, we denote the adjacency matrix of G by A.

(1) Since A is diagonalizable, the multiplicity of the eigenvalue r is equal to the dimension of the eigenspace $E_r = \{ f \in L^2(V) \mid A(f) = rf \}$. It is sufficient to prove that G is connected if and only if $\dim(E_r) = 1$.

Suppose G is connected and f is an eigenvector associated to r. We will show that f is constant on the whole of V. Let $v \in V$ be a vertex such that $|f(v)| = \max_{w \in V} |f(w)|$. Since $-f$ is also an eigenvector associated to r, we may assume that $f(v) > 0$. If $f(w) < f(v)$ for some vertex w adjacent to v, then

$$f(v) = \frac{A(f)(v)}{r} = \sum_{w \in V} \frac{a_{vw}}{r} f(w) < \sum_{w \in V} \frac{a_{vw}}{r} f(v) = f(v),$$

which is a contradiction. We can repeat this reasoning for every vertex w that is adjacent to v. Induction on the distance of w to v leads to the desired result, because G is connected.

Conversely, suppose by contraposition that G is disconnected. Let $v \in V$ and write V_1 for the set of all vertices for which there exists a path to v. Then, the functions

$$f_1(w) = \begin{cases} 1 & \text{if } w \in V_1, \\ 0 & \text{if } w \in {V_1}^{\mathrm{c}} \end{cases} \quad \text{and} \quad f_2(w) = \begin{cases} 0 & \text{if } w \in V_1, \\ 1 & \text{if } w \in {V_1}^{\mathrm{c}} \end{cases}$$

are linearly independent eigenvectors of A associated to r. Hence, $\dim(E_r) > 1$.

(2) Let G be a bipartite graph with bipartition (V_1, V_2) and let λ be an eigenvalue of A with multiplicity m. We will show that $-\lambda$ is also an eigenvalue of A with multiplicity m, which in particular implies that $-r$ is an eigenvalue. Since $\dim(E_\lambda) = m$, there exist linearly independent eigenvectors $f_1, \ldots, f_m$ of A associated to λ. Define

$$g_i(v) = \begin{cases} f_i(v) & \text{if } v \in V_1, \\ -f_i(v) & \text{if } v \in V_2 \end{cases} \quad \text{for all } 1 \leq i \leq m.$$

For any $v \in V_1$, the only adjacent vertices are situated in V_2, hence

$$A(g_i)(v) = \sum_{w \in V_2} a_{vw} g_i(w) = -\sum_{w \in V} a_{vw} f_i(w) = -A(f_i)(v) = -\lambda g_i(v)$$

and similarly for $v \in V_2$. This shows that $-\lambda$ is an eigenvalue of A. Since the functions g_i are linearly independent, $-\lambda$ has multiplicity $l \geq m$. Reversing the roles of λ and $-\lambda$ gives $l = m$.

Conversely, suppose $-r$ is an eigenvalue of A with eigenvector f. First, assume that G is connected. As in the previous proof, we can pick a $v \in V$ such that

$|f(v)| = \max_{w \in V} |f(w)|$ and $f(v) > 0$, obtaining $f(v) = -f(w)$ for all w adjacent to v. Continuing this reasoning results in

$$f(w) = \begin{cases} f(v) & \text{if } d(v,w) \text{ is even,} \\ -f(v) & \text{if } d(v,w) \text{ is odd,} \end{cases}$$

so we get a bipartition (V_1, V_2) of G, where

$$V_1 = \{\, w \in V \mid f(w) = f(v) \,\} \quad \text{and} \quad V_2 = \{\, w \in V \mid f(w) = -f(v) \,\}.$$

If G is disconnected with m connected components, then every i-th connected component has a bipartition $(V_1^{(i)}, V_2^{(i)})$. It is easy to observe that $\left(\bigcup_{i=1}^m V_1^{(i)}, \bigcup_{i=1}^m V_2^{(i)}\right)$ is a bipartition of G. □

Note that we can also rewrite the proof of Proposition 3.2.5 in terms of the adjacency operator: the trick for the second statement is then to check that $A(f) = rf$ where $f(v) = 1$ for all $v \in V$. To conclude the section, we introduce a new concept that will play an important role throughout the rest of this chapter.

Definition 3.2.8. *Consider an r-regular graph G with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$. Then the* **spectral gap** *of G is defined as $\lambda_1 - \lambda_2 = r - \lambda_2$.*

There exists a close relation between the spectral gap and the expansion coefficient: the larger the former, the better the latter.

Theorem 3.2.9. *The following boundaries hold for an r-regular graph:*

$$\frac{r - \lambda_2}{2} \leq \varepsilon \leq \sqrt{2r(r - \lambda_2)}.$$

The Sections 3.3 to 3.4 are devoted to proving this fundamental result, which also gives a spectral characterization of expander families.

Corollary 3.2.10. *A sequence $(G_n)_n$ of r-regular graphs such that $|V(G_n)| \to \infty$ as $n \to \infty$ is an expander family if and only if there exists a real number $\alpha > 0$ such that $r - \lambda_2(G_n) \geq \alpha$ for all n.*

3.3 The Laplacian operator

In this section, we discuss another linear operator associated to a graph, called the Laplacian. We first introduce some notation.

Let (V, E) be a graph. We give E an arbitrary **orientation**, which means that for each edge $e \in E$, we label one endpoint e^- and the other endpoint e^+, and orient e from e^- to e^+.

Definition 3.3.1. *Given an orientation on the edges of a graph (V, E), we define*

$$\begin{aligned} d_V &\colon L^2(V) \to L^2(E) \colon f \mapsto d_V(f) \quad \textit{where} \quad d_V(f)(e) = f(e^+) - f(e^-), \\ d_E &\colon L^2(E) \to L^2(V) \colon f \mapsto d_E(f) \quad \textit{where} \quad d_E(f)(v) = \sum_{e \in E,\, v = e^+} f(e) - \sum_{e \in E,\, v = e^-} f(e). \end{aligned}$$

The **Laplacian operator** $\Delta \colon L^2(V) \to L^2(V)$ *is defined as* $\Delta = d_E \circ d_V$*.*

The maps d_V and d_E depend on the orientation of the graph. We will show in the next lemma that the Laplacian operator does not for regular graphs.

Lemma 3.3.2. *If (V, E) is an r-regular graph with adjacency matrix A, then $\Delta = rI - A$.*

Proof. For any fixed $f \in L^2(V)$ and $v \in V$, we have

$$\begin{aligned} \Delta(f)(v) &= \sum_{e \in E,\, v = e^+} d_V(f)(e) - \sum_{e \in E,\, v = e^-} d_V(f)(e) \\ &= \sum_{e \in E,\, v = e^+} f(v) - \sum_{\substack{e \in E \\ v = e^+,\, w = e^-}} f(w) - \sum_{\substack{e \in E \\ v = e^-,\, w = e^+}} f(w) + \sum_{e \in E,\, v = e^-} f(v) \\ &= rf(v) - \sum_{w \in V} a_{vw} f(w) \\ &= rf(v) - A(f)(v). \end{aligned}$$

The above calculation shows that $\Delta(f)(v) = (rI - A)f(v)$, which ends the proof. □

We end this short section with an investigation of some more properties of the Laplacian operator Δ. Meanwhile, we will see that the maps d_V and d_E are adjoint.

Proposition 3.3.3. *Let (V, E) be an r-regular graph with $|V| = n$. Given an orientation of the edges in E, the following results hold:*

(1) The eigenvalues of Δ are given by $0 = r - \lambda_1 \leq r - \lambda_2 \leq \cdots \leq r - \lambda_n$. In particular, the eigenvalues of Δ lie in $[0, 2r]$.

(2) Let $f \in L^2(V)$ and $g \in L^2(E)$. Then $\langle d_V(f), g\rangle = \langle f, d_E(g)\rangle$ and

$$\langle \Delta(f), f\rangle = \sum_{e \in E} |f(e^+) - f(e^-)|^2.$$

Proof. The first statement is easy to prove: if $f \in L^2(V)$ is an eigenvector of the adjacency matrix A with corresponding eigenvalue λ, then we immediately get

$$\Delta(f) = rf - Af = rf - \lambda f = (r - \lambda)f,$$

using Lemma 3.3.2 in the first equality. Since $|\lambda| \leq r$, we have $|r - \lambda| \leq 2r$. The second result is also proved by some straightforward calculations. First note that

$$\begin{aligned}\langle d_V(f), g\rangle &= \sum_{e\in E} d_V(f)(e)\overline{g(e)} = \sum_{e\in E} \big(f(e^+) - f(e^-)\big)\overline{g(e)} \\ &= \sum_{v\in V} f(v) \sum_{e\in E,\, v=e^+} \overline{g(e)} - \sum_{v\in V} f(v) \sum_{e\in E,\, v=e^-} \overline{g(e)} = \sum_{v\in V} f(v)\overline{d_E(g)(v)} \\ &= \langle f, d_E(g)\rangle.\end{aligned}$$

Thus, $\langle \Delta(f), f\rangle = \big\langle d_E\big(d_V(f)\big), f\big\rangle = \overline{\big\langle f, d_E\big(d_V(f)\big)\big\rangle} = \langle d_V(f), d_V(f)\rangle$ where

$$\langle d_V(f), d_V(f)\rangle = \sum_{e\in E} \big(f(e^+) - f(e^-)\big)\overline{\big(f(e^+) - f(e^-)\big)} = \sum_{e\in E} |f(e^+) - f(e^-)|^2,$$

so we are done. □

3.4 The Rayleigh–Ritz theorem

We will now prove the Rayleigh–Ritz theorem, which provides a useful method for determining the second-largest eigenvalue λ_2 of a regular graph. We first introduce some new notation.

Definition 3.4.1. *Let S be a finite set. We denote c_α for the constant function that is equal to $\alpha \in \mathbb{R}$ on the whole of S, and we define*

$$\begin{aligned}L^2(S, \mathbb{R}) &= \{\, f \colon S \to \mathbb{R} \,\}, \\ L^2_0(S, \mathbb{R}) &= \{\, f \in L^2(S, \mathbb{R}) \mid \langle f, c_1\rangle_2 = 0 \,\}.\end{aligned}$$

Note that the inner product $\langle\,\cdot,\cdot\,\rangle_2$ in the definition of $L^2_0(S, \mathbb{R})$ is well-defined, since $L^2(S, \mathbb{R}) \subseteq L^2(S)$. The inner product of two functions in $L^2(S, \mathbb{R})$ also simplifies, since the conjugate of a real number is the number itself. If $f, g \in L^2(S, \mathbb{R})$, then

$$\langle f, g\rangle_2 = \sum_{x\in S} f(x)g(x) \qquad \text{and} \qquad \|f\|_2 = \sqrt{\sum_{x\in S} f(x)^2}.$$

For brevity, the domain S of the constant functions c_α is not explicitly contained in the notation, but it will always be clear from the context.

Theorem 3.4.2 (Rayleigh–Ritz)**.** *For any r-regular graph (V, E) we have*

$$\lambda_2 = \max_{f \in L^2_0(V,\mathbb{R})} \frac{\langle A(f), f\rangle}{\|f\|^2} = \max_{f \in L^2_0(V,\mathbb{R}),\, \|f\|=1} \langle A(f), f\rangle.$$

Proof. Let $|V| = n$ and denote the adjacency matrix of the graph by A. A classic result in linear algebra guarantees the existence of an orthonormal basis $B = \{f_1, \dots, f_n\}$ for $L^2(V, \mathbb{R})$, such that every f_i is a real-valued eigenvector of the operator A associated with the eigenvalue λ_i. Pick an $f \in L^2_0(V, \mathbb{R})$ with $\|f\| = 1$. Then, $f = \sum_{i=1}^n \alpha_i f_i$ for certain coefficients $\alpha_i \in \mathbb{R}$. Also note that

$$0 = \langle f, f_1\rangle = \sum_{i=1}^n \alpha_i \langle f_i, f_1\rangle = \alpha_1$$

by the constancy of f_1 due to Proposition 3.2.5, and the orthonormality of B. Next,

$$\begin{aligned}\langle A(f), f\rangle &= \Big\langle \sum_{i=2}^n \alpha_i A(f_i), \sum_{j=2}^n \alpha_j f_j \Big\rangle = \Big\langle \sum_{i=2}^n \alpha_i \lambda_i f_i, \sum_{j=2}^n \alpha_j f_j \Big\rangle \\ &= \sum_{i=2}^n \sum_{j=2}^n \alpha_i \alpha_j \lambda_i \langle f_i, f_j\rangle = \sum_{i=2}^n \alpha_i^2 \lambda_i \\ &\leq \lambda_2 \sum_{i=2}^n \alpha_i^2 = \lambda_2 \|f\|^2 = \lambda_2,\end{aligned}$$

where we used that $\|f\|^2 = \langle f, f\rangle = \sum_{i=2}^n \sum_{j=2}^n \alpha_i \alpha_j \langle f_i, f_j\rangle = \sum_{i=2}^n \alpha_i^2$. Hence,

$$\lambda_2 \geq \max_{f \in L^2_0(V,\mathbb{R}),\, \|f\|^2=1} \langle A(f), f\rangle.$$

The equality follows by noting that $\langle A(f_2), f_2\rangle = \langle \lambda_2 f_2, f_2\rangle = \lambda_2$ and $f_2 \in L^2_0(V, \mathbb{R})$. Indeed, recall from the observations after Definition 3.2.2 that f_1 is constant, so $\langle f_2, c_1\rangle = \langle f_2, \alpha f_1\rangle = 0$ for a certain $\alpha \in \mathbb{R}$. □

Corollary 3.4.3. *For any r-regular graph (V, E) we have*

$$r - \lambda_2 = \min_{f \in L^2_0(V,\mathbb{R})} \frac{\langle \Delta(f), f\rangle}{\|f\|^2} = \min_{f \in L^2_0(V,\mathbb{R}),\, \|f\|=1} \langle \Delta(f), f\rangle.$$

Proof. This follows directly from Theorem 3.4.2 and Lemma 3.3.2. □

We are now ready to prove the first inequality $(r - \lambda_2)/2 \leq \varepsilon$ in Theorem 3.2.9.

Proof of Theorem 3.2.9 (part 1). By the definition of ε, we can pick a set $S \subset V$ such that $|S| \leq |V|/2$ and $\varepsilon = |E(S, S^{\mathrm{c}})|/|S|$. Write $a = |S^{\mathrm{c}}|$ and $b = |S|$. Define the following two functions in $L^2(V, \mathbb{R})$:

$$g(v) = \begin{cases} a & \text{if } v \in S, \\ -b & \text{if } v \notin S \end{cases} \quad \text{and} \quad f = \frac{g}{\|g\|}.$$

We have that $\|f\| = 1$ and because

$$\sum_{v \in V} g(v) = \sum_{v \in S} a - \sum_{v \in S^{\mathrm{c}}} b = ba - ab = 0,$$

we see that $f, g \in L_0^2(V, \mathbb{R})$. If we orient the edges arbitrarily, we get by the second statement in Proposition 3.3.3 that

$$\langle \Delta(g), g \rangle = \sum_{e \in E} |g(e^+) - g(e^-)|^2 = \sum_{e \in E(S, S^{\mathrm{c}})} (a+b)^2 = |E(S, S^{\mathrm{c}})|(a+b)^2.$$

Also,

$$\|g\|^2 = \sum_{v \in V} g(v)^2 = \sum_{v \in S} a^2 + \sum_{v \in S^{\mathrm{c}}} b^2 = ba^2 + ab^2 = ab(a+b).$$

Since $b \leq a$ by definition, we obtain that

$$\langle \Delta(f), f \rangle = \frac{\langle \Delta(g), g \rangle}{\|g\|^2} = \frac{(a+b)\varepsilon b}{ab} = \big(1 + \frac{b}{a}\big)\varepsilon \leq 2\varepsilon$$

and we are done if we now apply Corollary 3.4.3 to the left hand side. □

We will break the proof of the second inequality $\varepsilon \leq \sqrt{2r(r - \lambda_2)}$ in Theorem 3.2.9 into two lemmas. We write $g \in L_0^2(V, \mathbb{R})$ for the eigenvector of A associated with λ_2 and define $V^+ = \{\, v \in V \mid g(v) \geq 0 \,\}$. We also define a function $f \in L^2(V, \mathbb{R})$ as

$$f(v) = \begin{cases} g(v) & \text{if } v \in V^+, \\ 0 & \text{if } v \notin V^+. \end{cases} \tag{3.2}$$

Note that $V^+ \neq \emptyset$ and $V^+ \neq V$ since $\sum_{v \in V} g(v) = 0$ and $g \neq c_0$. Therefore, f cannot be constant. In particular, $f \neq c_0$ which implies $\langle f, f \rangle \neq 0$. Lastly, because $-g$ is also an eigenvector of A associated with λ_2, we may assume that $|V^+| \leq |V|/2$.

Lemma 3.4.4. *The following holds for the function f defined in (3.2):*

$$\frac{\langle \Delta(f), f \rangle}{\langle f, f \rangle} \leq r - \lambda_2.$$

Proof. If $v \in V^+$, then by Lemma 3.3.2 we have that

$$\begin{aligned}\Delta(f)(v) = rf(v) - A(f)(v) &= rg(v) - \sum_{w\in V^+} a_{vw} g(w) \\ &\le rg(v) - \sum_{w \in V} a_{vw} g(w) = \Delta(g)(v).\end{aligned}$$

Thus,

$$\begin{aligned}\langle \Delta(f), f\rangle &= \sum_{v\in V^+} \Delta(f)(v) f(v) \le \sum_{v \in V^+} \Delta(g)(v) f(v) \\ &= \sum_{v\in V^+} \big(rg(v) - \lambda_2 g(v)\big) f(v) = (r-\lambda_2) \sum_{v\in V^+} f(v)^2 \\ &= (r-\lambda_2)\langle f, f\rangle,\end{aligned}$$

which implies the desired result. □

Lemma 3.4.5. *The following holds for the function f defined in (3.2):*

$$\frac{\varepsilon^2}{2r} \le \frac{\langle \Delta(f), f\rangle}{\langle f, f\rangle}.$$

Proof. Orient the edges of the graph such that $f(e^+) \ge f(e^-)$ for all edges $e \in E$, and define

$$D = \sum_{e\in E} f(e^+)^2 - f(e^-)^2.$$

We will show that

$$\varepsilon\langle f, f\rangle \le D \le \sqrt{2r\langle \Delta(f), f\rangle\langle f, f\rangle}, \tag{3.3}$$

which proves the lemma after squaring. First, we use Cauchy–Schwarz, Proposition 3.3.3 and the fact that $(a+b)^2 \le (a+b)^2 + (a-b)^2 = 2(a^2+b^2)$ for all $a, b \in \mathbb{R}$:

$$\begin{aligned} D &= \sum_{e\in E} \big(f(e^+) + f(e^-)\big)\big(f(e^+) - f(e^-)\big) \\ &\le \sqrt{\sum_{e\in E} \big(f(e^+) + f(e^-)\big)^2} \sqrt{\sum_{e \in E} \big(f(e^+) - f(e^-)\big)^2} \\ &\le \sqrt{2\sum_{e\in E} \big(f(e^+)^2 + f(e^-)^2\big)} \sqrt{\langle \Delta(f), f\rangle} \\ &= \sqrt{2r\sum_{v\in V} f(v)^2} \sqrt{\langle\Delta(f), f\rangle} = \sqrt{2r\langle \Delta(f), f\rangle \langle f, f\rangle}.\end{aligned}$$

For the other inequality in (3.3), let $0 = \alpha_1 < \alpha_2 < \cdots < \alpha_m$ be the values of f on V and write $V_i = \{ v \in V \mid f(v) \geq \alpha_i \}$. We obviously have

$$V_m \subset V_{m-1} \subset \cdots \subset V_1 = V.$$

We have already mentioned that f cannot be constant, so there is an edge $e \in E$ such that $f(e^+) \neq f(e^-)$. Thus, $f(e^+) = \alpha_i$ and $f(e^-) = \alpha_j$ for certain $1 \leq j < i \leq m$ because of the chosen orientation. That means

$$e \in \bigcap_{k=j+1}^{i} E(V_k, {V_k}^{\mathrm{c}})$$

and $e \notin \{ E(V_l, V_l^{\mathrm{c}}) \mid l = 1, \ldots, j, i+1, \ldots, m \}$. Furthermore,

$$f(e^+)^2 - f(e^-)^2 = \alpha_i^2 - \alpha_j^2 = \sum_{k=j+1}^{i} \alpha_k^2 - \alpha_{k-1}^2,$$

hence

$$D = \sum_{\substack{e \in E \\ f(e^+)=\alpha_i \\ f(e^-)=\alpha_j \\ 1 \leq j < i \leq m}} \sum_{k=j+1}^{i} (\alpha_k^2 - \alpha_{k-1}^2) = \sum_{k=1}^{m} |E(V_k, {V_k}^{\mathrm{c}})| (\alpha_k^2 - \alpha_{k-1}^2).$$

Because $V_i \subseteq V^+$ and $|V^+| \leq |V|/2$, we have $\varepsilon \leq |E(V_i, {V_i}^{\mathrm{c}})|/|V_i|$ for all $1 \leq i \leq m$. This leads to the desired result as follows:

$$\begin{aligned} D &\geq \sum_{k=1}^{m} \varepsilon |V_k| (\alpha_k^2 - \alpha_{k-1}^2) = \varepsilon \Bigg(|V_m| \alpha_m^2 + \sum_{k=1}^{m-1} \alpha_k^2 \big(|V_k| - |V_{l+1}| \big) \Bigg) \\ &= \sum_{i=1}^{m} \sum_{\substack{v \in V^+ \\ f(v)=\alpha_i}} \varepsilon f(v)^2 = \varepsilon \langle f, f \rangle, \end{aligned}$$

where we used in the second line that $v \in V_i \setminus V_{i+1}$ if and only if $f(v) = \alpha_i$. □

Proof of Theorem 3.2.9 (part 2). Combining Lemma 3.4.4 and Lemma 3.4.5 immediately gives the desired inequality. □

3.5 The Alon–Boppana theorem

Recall that the inequalities in Theorem 3.2.9 imply a large expansion coefficient if the spectral gap $r - \lambda_2$ is large. Therefore, if we want regular graphs with a large expansion coefficient, our goal is to find graphs with a small eigenvalue λ_2. In this section, we will show that there exists a constraint on how small λ_2 can be:

Proposition 3.5.1. *If $(G_n)_n$ is a sequence of connected r-regular graphs with $|V(G_n)| \to \infty$ as $n \to \infty$, then*

$$\liminf_{n\to\infty} \lambda_2(G_n) \geq 2\sqrt{r-1}.$$

This is equivalent to saying that for every $\varepsilon > 0$, there exists an index $n_0 > 0$ such that

$$\lambda_2(G_n) > 2\sqrt{r-1} - \varepsilon \quad \text{for all } n \geq n_0.$$

In other words: for connected r-regular graphs, λ_2 is at best a little bit smaller than $2\sqrt{r-1}$, so asymptotically the best spectral gap is $r - 2\sqrt{r-1}$.

We first prove another result due to A. Nilli (a pseudonym of Noga Alon, see [32]) that will imply Proposition 3.5.1. Remember that the traditional floor function is defined as follows: for any $\alpha \in \mathbb{R}$, the symbol $\lfloor \alpha \rfloor$ stands for the greatest integer less than or equal to α.

Proposition 3.5.2. *For any connected r-regular graph G with* $\operatorname{diam}(G) \geq 4$, *we have*

$$\lambda_2 > 2\sqrt{r-1} - \frac{2\sqrt{r-1}-1}{\left\lfloor \frac{1}{2}\operatorname{diam}(G) - 1 \right\rfloor}.$$

Proof. Define $b = \left\lfloor \frac{1}{2}\operatorname{diam}(G) - 1 \right\rfloor$ and $q = r - 1$, and write $G = (V, E)$. Note that $\operatorname{diam}(G) \geq 4$ implies $b > 0$ and $q > 0$. The latter is true because the only connected 1-regular graph is K_2, which consists of a single edge between two vertices, hence $\operatorname{diam}(K_2) = 1 < 4$. The proof requires quite some computations, so we will break it down into four steps. The general idea is to use the Rayleigh–Ritz theorem for a carefully constructed $f \in L_0^2(V, \mathbb{R})$, which leads to the inequality $\lambda_2 \geq r - \langle \Delta(f), f\rangle / \langle f, f\rangle$. Calculating $\langle f, f\rangle$ and finding an upper bound for $\langle \Delta(f), f\rangle$ will then provide the desired lower bound for λ_2.

Step 1. By definition of a graph's diameter, we can pick two vertices $x, y \in V$ such that $d(x, y) \geq 2b + 2$. Define the following sets for all $0 \leq i \leq b$:

$$A_i = \{\, v \in V \mid d(v, x) = i \,\} \quad \text{and} \quad B_i = \{\, v \in V \mid d(v, y) = i \,\}.$$

If $v \in A_i \cap B_j$ for certain $0 \leq i, j \leq b$, then $d(x, y) \leq d(x, v) + d(v, y) = i + j < 2b + 2$. This is impossible, so all A_i and B_j are disjoint and we can define the disjoint unions

$$A = \bigcup_{i=0}^{b} A_i \quad \text{and} \quad B = \bigcup_{i=0}^{b} B_i.$$

Suppose that there exist adjacent vertices $v \in A$ and $w \in B$. Then, we would get $d(x, y) \leq d(x, v) + d(v, w) + d(w, y) \leq 2b + 1 < 2b + 2$, which is again impossible. We now construct a function $f \in L^2(V, \mathbb{R})$ as follows:

$$f(v) = \begin{cases} \alpha & \text{if } v \in A_0, \\ \alpha/q^{(i-1)/2} & \text{if } v \in A_i \text{ for } 1 \leq i \leq b, \\ 1 & \text{if } v \in B_0, \\ 1/q^{(i-1)/2} & \text{if } v \in B_i \text{ for } 1 \leq i \leq b, \\ 0 & \text{otherwise,} \end{cases}$$

where we can choose $\alpha \in \mathbb{R}$ such that $f \in L_0^2(V, \mathbb{R})$. Indeed, because

$$\langle f, c_1 \rangle = \alpha\left(1 + \sum_{i=1}^{b} \frac{|A_i|}{q^{(i-1)/2}}\right) + \left(1 + \sum_{i=1}^{b} \frac{|B_i|}{q^{(i-1)/2}}\right) = \alpha\beta_1 + \beta_2$$

for certain $\beta_1, \beta_2 > 0$, we get that $\langle f, c_1 \rangle = 0$ if $\alpha = -\beta_2/\beta_1$.

Step 2. We can easily compute the inner product

$$\langle f, f \rangle = \left(\alpha^2 + \sum_{i=1}^{b} \frac{\alpha^2 |A_i|}{q^{i-1}}\right) + \left(1 + \sum_{i=1}^{b} \frac{|B_i|}{q^{i-1}}\right)$$

and we denote the first and second term by P_1 and P_2 respectively.

Step 3. We now search an upper bound for $\langle \Delta(f), f \rangle$. If we orient the edges in E arbitrarily, we get from Proposition 3.3.3 that $\langle \Delta(f), f \rangle = Q_1 + Q_2$ where

$$Q_1 = \sum_{\substack{e \in E \\ e^+ \in A \text{ or } e^- \in A}} \big(f(e^+) - f(e^-)\big)^2 \quad \text{and} \quad Q_2 = \sum_{\substack{e \in E \\ e^+ \in B \text{ or } e^- \in B}} \big(f(e^+) - f(e^-)\big)^2,$$

because there are no vertices in A that are adjacent to a vertex in B. We also have

$$Q_1 = \sum_{i=0}^{b-1} \sum_{\substack{v \in A_i \\ w \in A_{i+1}}} A_{vw}\big(f(v) - f(w)\big)^2 + \sum_{\substack{v \in A_b \\ w \notin A}} A_{vw} f(v)^2,$$

where $f(v) - f(w) = \alpha - \alpha = 0$ if $i = 0$. For all $v \in A_i$ where $1 \leq i \leq b-1$, at most q vertices in A_{i+1} are adjacent to v, because at least one vertex in A_{i-1} is adjacent to v. We therefore obtain

$$\begin{aligned} Q_1 &\leq \sum_{i=1}^{b-1} q|A_i| \left(\frac{\alpha}{q^{(i-1)/2}} - \frac{\alpha}{q^{i/2}}\right)^2 + q|A_b| \frac{\alpha^2}{q^{b-1}} \\ &= \sum_{i=1}^{b-1} q|A_i| \frac{\alpha^2(\sqrt{q}-1)^2}{q^i} + \big((\sqrt{q}-1)^2 + 2\sqrt{q} - 1\big) \frac{\alpha^2 |A_b|}{q^{b-1}} \\ &= (\sqrt{q}-1)^2 \sum_{i=1}^{b} \frac{\alpha^2 |A_i|}{q^{i-1}} + (2\sqrt{q}-1) \frac{\alpha^2 |A_b|}{q^{b-1}} \\ &= (\sqrt{q}-1)^2 (P_1 - \alpha^2) + \frac{2\sqrt{q}-1}{b} \frac{\alpha^2 b |A_b|}{q^{b-1}}. \end{aligned}$$

We try to find an upper bound for the second term in the previous line. We saw earlier that $|A_{i+1}| \leq q|A_i|$ for all $1 \leq i \leq b-1$, so

$$|A_1| \geq \frac{|A_2|}{q} \geq \frac{|A_3|}{q^2} \geq \cdots \geq \frac{|A_b|}{q^{b-1}}.$$

In particular,

$$\frac{\alpha^2 b|A_b|}{q^{b-1}} = \sum_{i=1}^{b} \frac{\alpha^2|A_b|}{q^{b-1}} \leq \sum_{i=1}^{b} \frac{\alpha^2|A_i|}{q^{i-1}} = P_1 - \alpha^2.$$

We thus get

$$Q_1 \leq \left((\sqrt{q}-1)^2 + \frac{2\sqrt{q}-1}{b}\right)(P_1 - \alpha^2) < \left(q + 1 - 2\sqrt{q} + \frac{2\sqrt{q}-1}{b}\right)P_1,$$

where we used that the first factor cannot be equal to 0 because $q \geq 1$. Of course, we can repeat the above calculations entirely for Q_2, replacing all appearances of A_i, α^2 and P_1 by B_i, 1 and P_2 respectively. This results in

$$Q_2 < \left(q + 1 - 2\sqrt{q} + \frac{2\sqrt{q}-1}{b}\right)P_2,$$

so we can finally conclude that

$$\langle \Delta(f), f\rangle < \left(r - 2\sqrt{q} + \frac{2\sqrt{q}-1}{b}\right)\langle f, f\rangle.$$

Step 4. Corollary 3.4.3 delivers

$$r - \lambda_2 \leq \frac{\langle \Delta(f), f\rangle}{\langle f, f\rangle} < r - 2\sqrt{r-1} + \frac{2\sqrt{r-1}-1}{b}$$

and solving this inequality for λ_2 gives the desired result. □

Before continuing, we would like to point out that the previous result is not true for a connected regular graph G with $\operatorname{diam}(G) = 1$, i.e. a complete graph. We have already mentioned that the above proof does not work for K_2 and this graph is also a counterexample. Indeed, one can easily compute that both sides of the strict inequality in Proposition 3.5.2 are equal to -1, which is impossible.

Proof of Proposition 3.5.1. Consider a fixed graph $G_n = (V, E)$ and pick a certain vertex $v \in V$. Because of the regularity of G_n, there are r paths of length 1 starting at v and $r(r-1) < r^2$ paths of length 2 starting at v. By induction, there are less than $r^{\operatorname{diam}(G_n)}$ paths of length $\operatorname{diam}(G_n)$ starting at v. These paths cover the entire graph and each such path contains $\operatorname{diam}(G_n) + 1$ vertices, thus

$$|V(G_n)| < \big(\operatorname{diam}(G_n) + 1\big)r^{\operatorname{diam}(G_n)}.$$

We assumed that $|V(G_n)| \to \infty$ as $n \to \infty$, so $\operatorname{diam}(G_n) \to \infty$ accordingly. Therefore, the last term in Proposition 3.5.2 approaches 0 as $n \to \infty$ and this implies the desired inequality. □

Suppose G is an r-regular graph with n vertices. Recall from Proposition 3.2.5 that r is always an eigenvalue and from Proposition 3.2.7 that $-r$ is an eigenvalue if G is bipartite. These integers are called the **trivial eigenvalues**. Following [25], we denote $\lambda(G)$ for the absolute value of the largest eigenvalue of G that is distinct from the trivial eigenvalues. Equivalently,

$$\lambda(G) = \begin{cases} \max\{|\lambda_2(G)|, |\lambda_n(G)|\} & \text{if } G \text{ is not bipartite,} \\ \max\{|\lambda_2(G)|, |\lambda_{n-1}(G)|\} & \text{if } G \text{ is bipartite.} \end{cases}$$

Note that $\lambda(G)$ is able to 'detect' disconnected graphs: in that case $\lambda(G) = r$ by Proposition 3.2.7. In general, we have $\lambda(G) \geq \lambda_2(G)$, so a sequence $(G_n)_n$ of connected r-regular graphs with $|V(G_n)| \to \infty$ as $n \to \infty$ is an expander family if $\big(r - \lambda(G_n)\big)_n$ is bounded away from zero. Since any lower bound for $\lambda_2(G)$ is also a lower bound for $\lambda(G)$, Proposition 3.5.1 directly implies

Theorem 3.5.3 (Alon–Boppana). *If $(G_n)_n$ is a sequence of connected r-regular graphs with $|V(G_n)| \to \infty$ as $n \to \infty$, then*

$$\liminf_{n\to\infty} \lambda(G_n) \geq 2\sqrt{r-1}.$$

Therefore, if we want $\lambda(G)$ to be as small as possible, $2\sqrt{r-1}$ serves as the lower limit of what can be done for graphs with a very large vertex set. In [25], Lubotzky, Phillips and Sarnak introduced a special kind of graph that is optimal in this sense.

Definition 3.5.4. *An r-regular graph G is* **Ramanujan** *if $\lambda(G) \leq 2\sqrt{r-1}$.*

In the last three decades, the study of Ramanujan graphs has gained prominence because they fuse diverse branches of pure mathematics, such as number theory, representation theory and algebraic geometry. For our purposes, sequences of r-regular Ramanujan graphs $(G_n)_n$ are interesting because they form expander families if $r \geq 3$. Indeed,

$$\varepsilon(G_n) \geq \frac{r - \lambda_2(G_n)}{2} \geq \frac{r - 2\sqrt{r-1}}{2} > 0 \quad \text{for all } n \geq 1.$$

Note that we cannot pick $r = 2$, because we already know from Example 6 that the cycle graphs $(C_n)_n$ are no expander family.

Chapter 4

Zig-zag products

We mentioned in the previous chapter that Pinsker was able to prove the existence of expander families in [34], using relatively easy probabilistic techniques. However, explicitly constructing such families seemed troublesome. Margulis was the first to give an explicit construction in [26], based on the theory of group representations. Over the next thirty years, many other explicit constructions were discovered, but they all relied on rather heavy algebraic techniques, such as algebraic geometry, group theory and number theory. Only in 2002, Reingold, Vadhan and Wigderson significantly simplified matters with an elementary combinatorial method in [35]. They came up with a whole new type of graph product, the so-called zig-zag product. The only minor disadvantage of their construction is that it forces us to consider multigraphs. In this chapter, we will present the Reingold–Vadhan–Wigderson expansion family, following [17, 22, 35, 40].

4.1 Definition of the zig-zag product

Informally, taking the zig-zag product of a large and a small multigraph results in a multigraph that roughly inherits its size from the large one, its degree from the small one, and its expansion properties from both. That means the composed multigraph has good expansion properties if both original multigraphs have good expansion properties. In Sections 4.1 and 4.2, G and H respectively will always be

$$r_G\text{-regular and } r_H\text{-regular multigraphs such that } |V(H)| = r_G, \tag{4.1}$$

and we will write $V = V(G) \times V(H)$. We start off with formally defining what it means to assign a unique label to each of the edges incident with a certain vertex $v \in V(G)$. For $E = E(G)$, we denote the multiset $E_v = \{\, e \in E \mid e \text{ is incident with } v \,\}$.

Definition 4.1.1. *Let G and H be multigraphs satisfying (4.1). For every vertex $v \in V(G)$, we call a bijection $L_v \colon V(H) \to E_v$ the* **labeling at** *v. The set $L = \{\, L_v \mid v \in V(G) \,\}$ is called the* **labeling** *from H to G.*

The condition $|V(H)| = r_G$ guarantees the existence of such labelings. In order to avoid notational clutter further on, it is convenient to regard $V(H)$ as the set $\{1, \dots, r_G\}$, such that labelings assign a numerical label to every edge. It might be syntactically unclear that $L_v(i)$ is an edge in $E(G)$ for every $v \in V(G)$ and $i \in V(H)$; it should therefore be read as "the edge incident with v with label i". Before describing the full technicalities of the zig-zag product, it seems best to first introduce another construction in order to gain some intuition (similar to the approaches in [17, 40]).

Definition 4.1.2. *Let G and H be multigraphs satisfying (4.1). The* **replacement product** *$G \circledR_L H$ with labeling L is the graph $\big(V, E\big)$ where*

- *$\{(v,i),(v,j)\} \in E$ for all $v \in V(G)$ if $\{i,j\} \in E(H)$,*
- *$\{(v,i),(w,j)\} \in E$ if $L_v(i) = L_w(j)$.*

It is preferable to think of $V(G \circledR_L H)$ as being created by replacing every vertex $v \in V(G)$ with a **cloud** of vertices $\{\, (v,i) \mid i \in V(H) \,\}$. Adding the prescribed edges to these clouds results in $|V(G)|$ exact copies of H, which are interconnected in a particular way that depends on the labeling. The following example illustrates that different labelings can lead to non-isomorphic replacement products, hence the necessary inclusion of the chosen labeling L in the notation.

Example 7. Consider the graphs G and H in Fig. 4.1. We see that G is 3-regular, H is 2-regular and $|V(H)| = 3$. Thus, the conditions (4.1) are fulfilled, and we can denote $V(G) = \{v, w\}$ and $V(H) = \{1, 2, 3\}$ as shown in the figure.

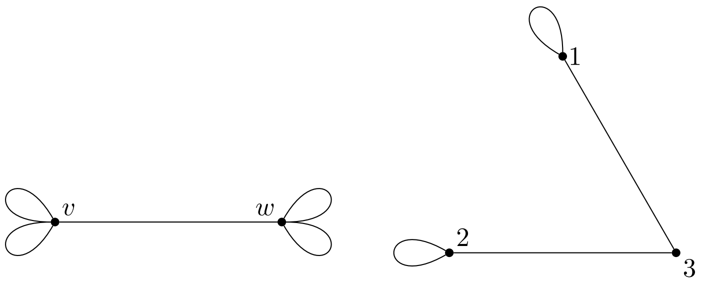


Figure 4.1: The multigraphs G (left) and H (right).

We fix two distinct labelings L and L' from H to G, which we depict in Fig. 4.2

by labeling the edges of G near each vertex. For example, $L_v(3)$, $L_w(3)$, $L'_v(3)$ and $L'_w(2)$ are all the edge that is incident with the vertices v and w.

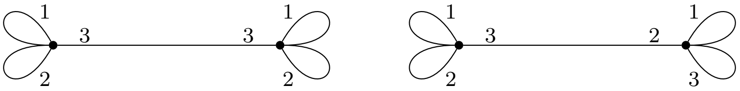


Figure 4.2: The labelings L (left) and L' (right).

The resulting replacement products $R = G \circledR_L H$ and $R' = G \circledR_{L'} H$ can be found in Fig. 4.3. The bold edges are the 'inter-cloud' edges, arising from the second bullet in Definition 4.1.2.

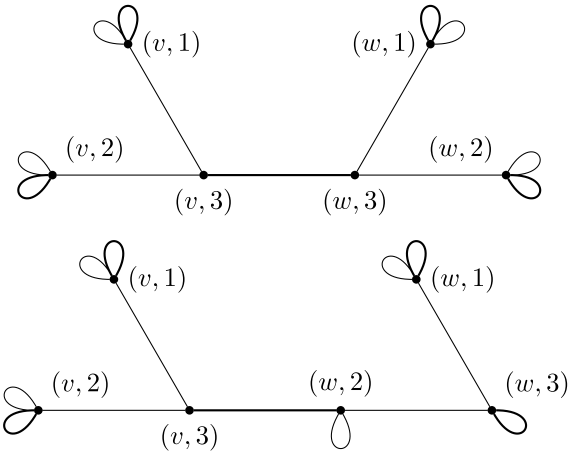


Figure 4.3: The graphs R (top) and R' (bottom).

The graphs R and R' respectively have four and three vertices with two loops, so they cannot be isomorphic. In this simple example, we could quickly determine the replacement products by hand; for more complicated cases, one can use the Matlab code provided in Appendix A.

The zig-zag product of multigraphs G and H also takes V as vertex set, the edges arise from walks in $G \circledR_L H$ of length three and of 'zig-zag shape'. The latter means that (v,i) and (w,j) are adjacent if we can move from (v,i) to an adjacent vertex (v,i') in the same cloud, then jump to a vertex (w,j') in another cloud and finally move from (w,j') to (w,j) in this new cloud. This three-step process clarifies the nomenclature. The formal definition is now much less intimidating.

Definition 4.1.3. *Let G and H be multigraphs satisfying (4.1). The* **zig-zag product** $G \circledz_L H$ *with labeling L is equal to $\big(V, E\big)$, where the multiplicity of an edge $\{(v,i),(w,j)\}$ is equal to the number of pairs $(i',j') \in V(H) \times V(H)$ such that $\{i,i'\},\{j,j'\} \in E(H)$ and $L_v(i') = L_w(j')$.*

By construction, if two replacement products under different labelings are isomorphic, then so are the zig-zag products under the same labelings. The converse, however, is not true.

Example 8. Consider the multigraphs G and H, and the labelings L and L' from Example 7. It is easily checked by hand or with the Matlab code in Appendix A that the zig-zag products $M = G \circledz_L H$ and $M' = G \circledz_{L'} H$ are both isomorphic to the graph displayed in Fig. 4.4.

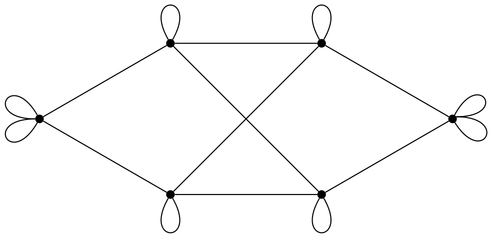


Figure 4.4: The zig-zag products M and M'.

Based on Example 8, we might conjecture that zig-zag products are independent of the chosen labeling, but the following counterexample demonstrates the contrary. However, we will always simplify the notation of a zig-zag product to $G \circledz H$ if the used labeling is either irrelevant or clear from the context.

Example 9. The multigraphs G and H displayed in Fig. 4.5 satisfy (4.1), because they are both 3-regular and $|V(H)| = 3$.

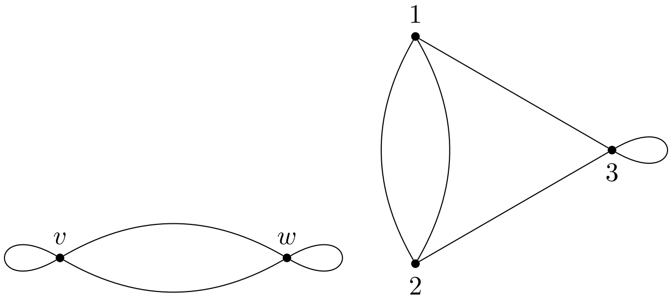


Figure 4.5: The multigraphs G (left) and H (right).

We can thus use the Matlab code in Appendix A to compute the adjacency matrices of the zig-zag products $G \circledcirc_L H$ and $G \circledcirc_{L'} H$, where the labelings L and L' from H to G are shown in Fig. 4.6.

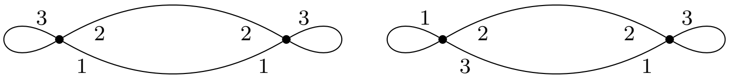


Figure 4.6: The labelings L (left) and L' (right).

It turns out that these adjacency matrices are respectively

$$\begin{pmatrix} 1 & 1 & 1 & 4 & 0 & 2 \\ 1 & 1 & 1 & 0 & 4 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 \\ 4 & 0 & 2 & 1 & 1 & 1 \\ 0 & 4 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & 0 & 0 & 4 & 2 & 3 \\ 0 & 4 & 2 & 0 & 2 & 1 \\ 0 & 2 & 1 & 2 & 2 & 2 \\ 4 & 0 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ 3 & 1 & 2 & 1 & 1 & 1 \end{pmatrix}.$$

We immediately observe that the first zig-zag product contains a loop at every vertex, whereas the second has a vertex without loop. Hence, the multigraphs $G \circledcirc_L H$ and $G \circledcirc_{L'} H$ cannot be isomorphic.

In Example 9, it is no coincidence that the entries in every row and column of the adjacency matrices sum to the same value 9.

Proposition 4.1.4. *For multigraphs G and H as in (4.1), $G \circledcirc H$ is r_H^2-regular.*

Proof. This follows immediately from the regularity of H, and the fact that the choices of i' and j' in Definition 4.1.3 are independent of each other. □

4.2 Eigenvalues of zig-zag products

The main theorem of this section provides an upper bound for $\lambda(G \circledcirc H)$. In Section 4.3, this result will be the crucial key to show that a carefully constructed sequence of graphs forms an expander family.

Theorem 4.2.1. *Let G and H be non-bipartite multigraphs satisfying (4.1). Then,*

$$\lambda(G \circledcirc H) \leq \frac{r_H^2 \lambda(G)}{r_G} + r_H \lambda(H) + \lambda(H)^2.$$

Throughout the whole section, we will give $V(G)$ a fixed ordering $v_1, \ldots, v_n$ and we will order $V = V(G) \times V(H)$ lexicographically, which means that the ordering looks like $(v_1,1),\dots,(v_1,r_G),(v_2,1),\dots,(v_2,r_G),\dots,(v_n,1),\dots,(v_n,r_G)$. We respectively denote Z and N for the adjacency matrices of the graphs with vertex set V and edges defined as in respectively the first and second bullet of Definition 4.1.2. Note that Z is a $|V(G)||V(H)| \times |V(G)||V(H)|$-matrix of the special form

$$\begin{pmatrix} B & 0 & \cdots & 0 \\ 0 & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B \end{pmatrix},$$

where B is the adjacency matrix of the multigraph H. That means Z is independent of the labeling, whereas N is not.

It so happens that it is possible to entirely repeat the proof of Theorem 3.4.2 for complex-valued functions, such that we obtain

$$\lambda(G \circledz H) = \max_{f \in L_0^2(V)} \frac{|\langle M(f), f\rangle|}{\|f\|^2}, \tag{4.2}$$

where M is the adjacency operator associated to $G \circledz H$. The only technical subtlety is that $G \circledz H$ should be non-bipartite in order to have $\lambda_n \leq \lambda$. We can thus prove Theorem 4.2.1 with a typical Rayleigh–Ritz argument: pick an arbitrary $f \in L_0^2(V)$ and find an appropriate upper bound for $|\langle M(f), f\rangle|$.

Definition 4.2.2. *We call $f \in L^2(V)$* **constant on clouds** *if $f(v,i) = f(v,j)$ for all $v \in V(G)$ and $i,j \in V(H)$. We define an operator $C\colon L^2(V) \to L^2\big(V(G)\big)$ by*

$$C(f)(v) = \sum_{i \in V(H)} f(v,i) \quad \textit{for all } v \in V(G).$$

The main idea is to decompose each $f \in L_0^2(V)$ into a part that is constant on clouds and a part that sums to 0 on clouds.

Definition 4.2.3. *For all $f \in L_0^2(V)$, we define $f^{\parallel} \in L^2(V)$ and $f^{\perp} \in L^2(V)$ by*

$$f^{\parallel}(v,i) = \frac{1}{r_G} \sum_{j \in V(H)} f(v,j) \quad \textit{and} \quad f^{\perp} = f - f^{\parallel}.$$

In other words, $f^{\parallel}(v,i)$ is the average value of f over the cloud $\{\,(v,j) \mid j \in V(H)\,\}$, recalling that $r_G = |V(H)|$. Hence, $f^{\parallel}$ is constant on clouds and we also see that

$$C(f^{\perp})(v) = \sum_{i \in V(H)} f(v,i) - \sum_{i \in V(H)} f^{\parallel}(v,i) = 0.$$

When we regard $f^{\parallel}$ and $f^{\perp}$ as vectors, they are thus respectively parallel and orthogonal to the constant vector with entries 1, which explains the notation. The decomposition allows us to estimate $|\langle M(f), f\rangle|$ by a sum of three terms which each consist of either a Z or an N.

Lemma 4.2.4. *For any $f \in L_0^2(V)$, the following holds:*

$$|\langle M(f), f\rangle| \leq r_H^2 |\langle N(f^{\|}), f^{\|}\rangle| + 2r_H \|f^{\|}\| \|Z(f^{\perp})\| + \|Z(f^{\perp})\|^2.$$

Proof. By construction of the zig-zag product, we have $M = ZNZ$. Let $f \in L_0^2(V)$. The symmetry of Z gives $\langle M(f), f\rangle = \langle N(Z)(f), Z(f)\rangle$. If $g \in L^2(V)$ is constant on clouds, then we have for any $(v, i) \in V$ that

$$Z(g)(v,i) = \sum_{(w,j)\in V} z_{(v,i)(w,j)} g(w,j) = \sum_{j \in V(H)} z_{(v,i)(v,j)} g(v,i) = r_H g(v,i).$$

In particular, $Z(f^{\|}) = r_H f^{\|}$. The Cauchy–Schwarz inequality leads to

$$\begin{aligned} |\langle M(f), f\rangle| &= |\langle N(Z)(f^{\|} + f^{\perp}), Z(f^{\|} + f^{\perp})\rangle| \\ &\leq r_H^2 |\langle N(f^{\|}), f^{\|}\rangle| + r_H \|N(f^{\|})\| \|Z(f^{\perp})\| \\ &\quad + r_H \|N(Z)(f^{\perp})\| \|f^{\|}\| + \|N(Z)(f^{\perp})\| \|Z(f^{\perp})\| \\ &= r_H^2 |\langle N(f^{\|}), f^{\|}\rangle| + 2r_H \|f^{\|}\| \|Z(f^{\perp})\| + \|Z(f^{\perp})\|^2, \end{aligned}$$

where we used in the last equality that $\|N(h)\| = \|h\|$ for all $h \in L^2(V)$. This follows from the fact that N is a permutation matrix. Indeed, the graph determined by N is regular of degree 1 by construction. □

We now estimate the factors $|\langle N(f^{\|}), f^{\|}\rangle|$ and $\|Z(f^{\perp})\|$ in terms of the eigenvalues $\lambda(G)$ and $\lambda(H)$.

Lemma 4.2.5. *If G is non-bipartite, then for any $f \in L_0^2(V)$:*

$$|\langle N(f^{\|}), f^{\|}\rangle| \leq \frac{\lambda(G)}{r_G} \|f^{\|}\|^2.$$

Proof. Suppose $g, h \in L^2(V)$ such that h is constant on clouds. Then,

$$\begin{aligned} \langle C(g), C(h)\rangle &= \sum_{v \in V(G)} \Big(\sum_{i \in V(H)} g(v,i)\Big)\Big(\sum_{j \in V(H)} \overline{h(v,j)}\Big) \\ &= r_G \sum_{(v,i)\in V} g(v,i)\overline{h(v,i)} = r_G \langle g, h\rangle. \end{aligned}$$

For any $v \in V(G)$, we denote $\alpha_v = h(v,i)$ where $i \in V(H)$ and we write A for the adjacency operator associated to the multigraph G. On the one hand we have

$$A\big(C(h)\big)(v) = \sum_{w \in V(G)} a_{vw}\Big(\sum_{i \in V(H)} h(w,i)\Big) = r_G \sum_{w \in V(G)} \alpha_w a_{vw}$$

and on the other hand we get

$$C\big(N(h)\big)(v) = \sum_{i\in V(H)}\sum_{(w,j)\in V} n_{(v,i)(w,j)}h(w,j) = \sum_{w\in V(G)} \alpha_w\Big(\sum_{i,j\in V(H)} n_{(v,i)(w,j)}\Big),$$

where the last summation between brackets is equal to the number of edges in $E(G)$ between v and w, i.e. a_{vw}. Hence, $A\big(C(h)\big) = r_G C\big(N(h)\big)$. For all $f \in L_0^2(V)$, we can particularly choose $h = f^{\|}$. Since $C(f^{\perp}) = 0$, we obtain $C(f) = C(f^{\|})$ and

$$\langle N(f^{\|}), f^{\|}\rangle = \frac{\langle C\big(N(f^{\|})\big), C(f^{\|})\rangle}{r_G} = \frac{\langle A\big(C(f)\big), C(f)\rangle}{r_G^2}.$$

This finally leads to the desired result by the general counterpart of Eq. (4.2):

$$|\langle N(f^{\|}), f^{\|}\rangle| \le \frac{\lambda(G)}{r_G^2}\langle C(f^{\|}), C(f^{\|})\rangle = \frac{\lambda(G)}{r_G}\|f^{\|}\|^2,$$

using that $C(f) \in L_0^2\big(V(G)\big)$. Indeed, $\sum_{v\in V(G)} C(f)(v) = \sum_{(v,i)\in V} f(v,i) = 0$. □

Lemma 4.2.6. *If H is non-bipartite, then for any $f \in L_0^2(V)$:*

$$\|Z(f^{\perp})\| \le \lambda(H)\|f^{\perp}\|.$$

Proof. Let $f \in L_0^2(V)$. We write $f_v^{\perp}(i) = f^{\perp}(v,i)$, such that $f_v^{\perp} \in L_0^2\big(V(H)\big)$ for all $v \in V(G)$. As in the proof of Theorem 3.4.2, we can write $f_v^{\perp} = \sum_{j=2}^{n} \alpha_j g_j$, where any $\alpha_j \in \mathbb{C}$ and every g_j is an eigenvector of B associated to $\lambda_j(H)$. Since H is non-bipartite, we get that $\|B(f_v^{\perp})\| \le \lambda(H)\|f_v^{\perp}\|$ for all $v \in V(G)$. Next,

$$Z(f^{\perp})(v,i) = \sum_{(w,j)\in V} z_{(v,i)(w,j)} f^{\perp}(w,j) = \sum_{j\in V(H)} b_{ij} f^{\perp}(v,j) = B(f_v^{\perp})(i)$$

for any $(v,i) \in V$ and we also compute

$$\|f^{\perp}\|^2 = \sum_{(v,i)\in V} f^{\perp}(v,i)\overline{f^{\perp}(v,i)} = \sum_{v\in V(G)} \|f_v^{\perp}\|^2.$$

Combining the above three facts leads to

$$\|Z(f^{\perp})\|^2 = \sum_{(v,i)\in V} B(f_v^{\perp})(i)\overline{B(f_v^{\perp})(i)} = \sum_{v\in V(G)} \|B(f_v^{\perp})\|^2 \le \lambda(H)^2\|f^{\perp}\|^2,$$

so we are done after taking the square root. □

Proof of Theorem 4.2.1. Due to Eq. (4.2), it is sufficient to prove

$$\frac{|\langle M(f), f\rangle|}{\|f\|^2} \le \frac{r_H^2\lambda(G)}{r_G} + r_H\lambda(H) + \lambda(H)^2 \quad \text{for all } f \in L_0^2(V).$$

Let $f \in L_0^2(V)$. Lemmas 4.2.4, 4.2.5 and 4.2.6 immediately give

$$\frac{|\langle M(f), f\rangle|}{\|f\|^2} \leq \frac{r_H^2 \lambda(G)}{r_G} p^2 + 2r_H \lambda(H) pq + \lambda(H)^2 q^2,$$

where $p = \|f^{\|}\|/\|f\|$ and $q = \|f^{\perp}\|/\|f\|$. Since $f^{\|}$ and $f^{\perp}$ are orthogonal, we obtain by Pythagoras' theorem that $p^2 + q^2 = 1$. Hence, $p, q \leq 1$ and $0 \leq (p-q)^2 = 1 - 2pq$ or $2pq \leq 1$, which implies the desired result. □

To conclude, we note that the original paper [35] states Theorem 4.2.1 in a slightly different form: given an eigenvalue λ of an r-regular graph, we put $\tilde{\lambda} = \lambda/r$ and Theorem 4.2.1 then results in the more elegant $\tilde{\lambda}(G \, ⓩ \, H) \leq \tilde{\lambda}(G) + \tilde{\lambda}(H) + \tilde{\lambda}(H)^2$.

4.3 An explicit expander family

In this section, we present the recursive construction of the Reingold–Vadhan–Wigderson expander family from [35]. Therefore, we first need a special graph G that will act as a building block. It will turn out that G should be r-regular and non-bipartite such that $|V(G)| = r^4$ and $\lambda(G) \leq r/5$. Let us first construct such a graph.

We consider $\mathbb{Z}_p^8$ for a fixed prime number $p > 35$, the direct product of eight copies of the additive group $\mathbb{Z}_p$, which consists of the integers modulo p. The choice of 8 and 35 seems obscure at the moment, but these integers will ultimately do the job. Next, we define an injective map $h\colon \mathbb{Z}_p^2 \to \mathbb{Z}_p^8$ by

$$h(x, y) = (x, xy, xy^2, xy^3, xy^4, xy^5, xy^6, xy^7)$$

and write $S = \mathrm{Im}(h)$. We can now define $G = \mathrm{Cay}(\mathbb{Z}_p^8, S)$, because S is symmetric. Indeed, the inverse of an element $h(x, y) \in S$ is obviously $h(-x, y) \in S$. By Proposition 2.5.3, we have that G is $|S|$-regular, hence $r = p^2$. This also settles the condition $|V(G)| = p^8 = r^4$.

We are left to verify that G is non-bipartite and satisfies $\lambda(G) \leq r/5$. Therefore, we prove two lemmas, in which we write $\xi = \exp(2\pi i/p)$ and denote an element $(z_1, \ldots, z_8) \in \mathbb{Z}_p^8$ as z. The identity element of $\mathbb{Z}_p^8$ is written as 0 and we also recall the usual scalar product $a \cdot b = a_1 b_1 + \cdots + a_8 b_8$ for all $a, b \in \mathbb{Z}_p^8$.

Lemma 4.3.1. *All eigenvalues of G are of the following form for some $a \in \mathbb{Z}_p^8$:*

$$\lambda_a = \sum_{x, y \in \mathbb{Z}_p} \xi^{a \cdot h(x,y)}.$$

Proof. Let A be the adjacency operator associated to G. We first show that the numbers λ_a are eigenvalues of A with according eigenvectors $f_a \in L^2\big(V(G)\big)$, defined by $f_a(b) = \xi^{a\cdot b}$ for $b \in \mathbb{Z}_p^8$:

$$\begin{aligned} A(f_a)(b) &= \sum_{h(x,y)\in S} f_a\big(b + h(x,y)\big) = \sum_{x,y\in\mathbb{Z}_p} \xi^{a\cdot\big(b+h(x,y)\big)} \\ &= \sum_{x,y\in\mathbb{Z}_p} f_a(b)\xi^{a\cdot h(x,y)} = \lambda_a f_a(b). \end{aligned}$$

Secondly, we observe that all f_a are distinct. Indeed, if $f_a = f_c$ for some $a, c \in \mathbb{Z}_p^8$, then by definition $\xi^{(a-c)\cdot b} = 1$ and thus $p \mid (a-c)\cdot b$ for all $b \in \mathbb{Z}_p^8$. If we now choose particular elements b with entries 0 except for a 1 on the jth place, then $p \mid (a_j - c_j)$. That means $a_j - c_j = 0$ for every $1 \le j \le 8$, i.e. $a = c$.

As a last step, we show that the set $\{\, f_a \mid a \in \mathbb{Z}_p^8 \,\}$ is linearly independent, such that it becomes a basis of $\mathbb{Z}_p^8$. It is sufficient to prove that all eigenvectors f_a and f_b with $a \ne b$ are orthogonal. We compute

$$\langle f_a, f_b\rangle = \sum_{x\in\mathbb{Z}_p} \xi^{(a-b)\cdot x} = \frac{1-\xi^{(a-b)\cdot p}}{1-\xi^{a-b}} = 0,$$

so we are done. □

Lemma 4.3.2. *Using the notation from Lemma 4.3.1, we have*

$$0 \le \lambda_a \le r/5 \quad \textit{for all } a \ne 0.$$

Proof. Fix an $a \ne 0$. By definition, we have that $0 \le \lambda_a$. We can rewrite λ_a as

$$\lambda_a = \sum_{x,y\in\mathbb{Z}_p} \xi^{a_1 x}\xi^{a_2 xy}\cdots\xi^{a_8 xy^7} = \sum_{x,y\in\mathbb{Z}_p} \xi^{(a_1+a_2y+\cdots+a_8y^7)x}.$$

If we write $g(t) = a_1 + a_2 t + \cdots + a_8 t^7$, then we have for any fixed $y \in \mathbb{Z}_p$ that

$$\sum_{x\in\mathbb{Z}_p} \xi^{g(y)x} = \begin{cases} p & \text{if } g(y) = 0, \\ 0 & \text{if } g(y) \ne 0. \end{cases}$$

The first case is trivial and the second one follows again from the elementary formula

$$\sum_{x\in\mathbb{Z}_p} \xi^{g(y)x} = \frac{1-\xi^{g(y)p}}{1-\xi^{g(y)}} = 0.$$

Thus, $\lambda_a = np$ where n is the number of roots of g in $\mathbb{Z}_p$. But since $a \ne 0$, the polynomial g is non-zero and of degree at most 7, so a well-known result from field theory ensures that $n \le 7$. Recalling that $p > 35$, we obtain that $\lambda_a \le 7p < p^2/5$. □

Lemma 4.3.1 and Lemma 4.3.2 directly imply that $\lambda(G) \leq r/5$ and that $-r$ is not an eigenvalue of G. The latter fact combined with Proposition 3.2.4 and Proposition 3.2.7 shows that G is not bipartite. Before we finally demonstrate how the building block G is used, we need the concept of powers of multigraphs.

Definition 4.3.3. *The mth* **power of a multigraph** *G is the multigraph G^m whose vertex set is $V(G)$ and where the multiplicity of an edge $\{v, w\}$ is equal to the number of walks of length m from $v \in V(G)$ to $w \in V(G)$.*

For any fixed ordering of $V(G)$, it is known that the adjacency matrix of G^m is equal to A^m, where A is the adjacency matrix of G (refer for example to [7]). Specifically, the **square of a multigraph**, i.e. $m = 2$, will play an important role in the construction of the Reingold–Vadhan–Wigderson expander family. We will therefore investigate it a bit more on the basis of an easy example.

Example 10. Given a suitable ordering of the vertices, the adjacency matrices of the cycle graph C_4 and its square are respectively

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}.$$

We observe that C_4^2 contains both loops and multiple edges, so it is a multigraph. It is easily seen that the former holds in general: the square of a (multi)graph with a non-empty edge set contains at least one loop.

Proposition 4.3.4. *If a graph G has eigenvalues $\lambda_1, \ldots, \lambda_n$, then the multigraph G^m has eigenvalues $\lambda_1^m, \ldots, \lambda_n^m$.*

Proof. Let A be the adjacency matrix of G. There exists a matrix B such that BAB^{-1} is diagonal with diagonal entries $\lambda_1, \ldots, \lambda_n$. Thus, BA^mB^{-1} is diagonal with diagonal entries $\lambda_1^m, \ldots, \lambda_n^m$. Using a similar argument as in the proof of Proposition 3.2.4, we see that A^m and BA^mB^{-1} share the same eigenvalues. □

At long last, we have all the tools at our disposal to define the Reingold–Vadhan–Wigderson expander family and show that it is indeed an expander family.

Theorem 4.3.5. *Let G be the graph that was defined at the beginning of this section. The following sequence of multigraphs $(G_n)_n$ is an expander family:*

$$G_1 = G^2 \quad \text{and} \quad G_{n+1} = G_n^2 \textcircled{z} G,$$

where the zig-zag product is formed using an arbitrary labeling from G to G_n^2.

Taking the zig-zag product is allowed, because G_n^2 is regular of degree $|V(G)| = p^8$ for every $n \geq 1$. Indeed, G_1 is regular of degree $r^2 = p^4$, so G_1^2 is p^8-regular, and G_n is regular of degree $r^2 = p^4$ by Proposition 4.1.4, hence G_n^2 is p^8-regular for all $n > 1$. Also, Example 10 demonstrates the need for multigraphs.

Proof of Theorem 4.3.5. First, we have $|V(G_n)| \to \infty$ as $n \to \infty$, because it follows by induction that $|V(G_n)| = p^{8n}$. Indeed, $|V(G_1)| = p^8$ by definition of a Cayley graph and if we assume that $|V(G_{n-1})| = p^{8(n-1)}$, then we obtain

$$|V(G_n)| = |V(G_{n-1}^2)||V(G)| = p^{8(n-1)}p^8 = p^{8n}.$$

We have already checked that every G_n is p^4-regular, so we are left to prove that the spectral gaps of all G_n are bounded away from 0 (recall Corollary 3.2.10). It is sufficient to show that $\lambda(G_n) \leq 2p^4/5$ for all $n \geq 1$, since this implies that the spectral gap of G_n is bigger than $p^4 - 2p^4/5 > 0$. We use induction once more. The base case follows from Proposition 4.3.4 and the crucial property $\lambda(G) \leq p^2/5$:

$$\lambda(G_1) = \lambda(G)^2 \leq \frac{p^4}{25} \leq \frac{2p^4}{5}.$$

For the induction step, we assume that $\lambda(G_n) \leq 2p^4/5$. Since G_n^2 and G are non-bipartite, Theorem 4.2.1 implies

$$\lambda(G_{n+1}) \leq \frac{r^2\lambda(G_n^2)}{p^8} + r\lambda(G) + \lambda(G)^2 \leq \frac{4r^2}{25} + \frac{r^2}{5} + \frac{r^2}{25} = \frac{2p^4}{5},$$

where we again used Proposition 4.3.4 in the first term of the second inequality. □

To conclude, we may ask ourselves whether the miraculous numbers 8 and 35 are the only possible choice for constructing an expander family in the way we did. At first sight, we might try to modify the map h and repeat all proofs for other integers. Apart from the estimates, though, it seems tricky to fulfil all the required conditions for the subsequent zig-zag products. Maybe there is a deeper connection between 8, 35 and the zig-zag product after all?

Chapter 5

Key predistribution schemes

We now present in historical order several milestones in the study of key predistribution. All the discussed KPSs had and still have a major impact on the research field. For Sections 5.1 and 5.2 we mainly consulted [27,28,39], the formulas for the connectivity and resilience in Section 5.3 are from [18], and Section 5.4 is based on [16,20].

5.1 Blom key predistribution scheme

In 1985, Blom suggested the earliest combinatorial KPS in [4]. As will become clear later on, Blom's proposal does not strictly conform to what we defined as a KPS. Indeed, instead of keys, nodes store secret information that allows them to compute keys themselves. This storage reduction comes at the cost of a computational overhead for key establishment: it requires a particular polynomial evaluation. We will, however, call Blom's scheme a KPS nevertheless, since it inspired many KPSs in the more recent literature (see [13] for example).

Blom's scheme is **S-unconditionally secure**, which means that an adversary cannot compute any partial information about the keys of uncompromised nodes in polynomial time until a certain threshold S is exceeded. The complete pairwise KPS is $(n-2)$-unconditionally secure, but recall from Example 2 that we should try to reduce the amount of keys that need to be stored. The Blom KPS does precisely that, while still allowing each pair of nodes N_i and N_j to compute a secret key K_{ij}.

In general, S-unconditionally secure KPSs pre-specify a **security parameter** S, which is independent of the network size n. By definition, the network is perfectly resilient to node compromise until $S+1$ nodes have been compromised; at this point the entire network's communications are compromised. Differently stated, $\mathrm{fail}_s = 0$

for all $1 \leq s \leq S$ and $\text{fail}_s = 1$ when $S+1 \leq s \leq n-2$. For each pair of nodes N_i and N_j, the security condition thus becomes: compromising a set of at most S nodes that is disjoint from $\{N_i, N_j\}$ must not reveal any information about K_{ij}.

The key space in the Blom KPS is a finite field $\mathbb{F}_p$, where $p \geq n$ is a publicly known prime. The trusted authority transmits $S+1$ elements of $\mathbb{F}_p$ to each node over a secure channel, as opposed to $n-1$ elements in the complete pairwise KPS. In order to obtain some intuition, we first present the special case where $S = 1$.

Definition 5.1.1. *The* **Blom KPS** *for $S = 1$ works as follows:*

(1) A unique publicly known element $r_N \in \mathbb{F}_p$ is assigned to every node N.

(2) The trusted authority chooses three random elements $a, b, c \in \mathbb{F}_p$ and forms the bivariate polynomial $f \in \mathbb{F}_p[x, y]$, which is defined as

$$f(x, y) = a + b(x + y) + cxy \bmod p.$$

(3) All nodes N are preloaded with the two coefficients of $f(x, r_N) \bmod p$.

(4) Two nodes N_i and N_j communicate with the common key

$$K_{ij} = f(r_{N_j}, r_{N_i}) \bmod p.$$

Note that the first step is possible, because $p \geq n$. It is crucial that the polynomial f in the second step is symmetric, i.e. $f(x, y) = f(y, x)$ for all $x, y \in \mathbb{F}_p$, because it ensures that the common key in the last step is well-defined. Indeed,

$$K_{ij} = f(r_{N_i}, r_{N_j}) \bmod p = K_{ji}.$$

The polynomial $f(x, r_N) = (a + br_N) + (b + cr_N)x \bmod p$ is computed by the trusted authority before securely transmitting its coefficients, so every node N only stores the values

$$a_N = a + br_N \bmod p \quad \text{and} \quad b_N = b + cr_N \bmod p,$$

which means that the elements a, b and c remain private. The following example illustrates this protocol.

Example 11. Suppose that we have a network of size $n = 10$, which contains the nodes N_i, N_j and N_k. The trusted authority chooses $p = 11$, $r_{N_i} = 3$, $r_{N_j} = 6$ and $r_{N_k} = 8$, and fixes the polynomial

$$f(x, y) = 4 + 1(x + y) + 6xy.$$

After sending the appropriate coefficients to N_i, N_j and N_k, the aforementioned nodes can compute their communication keys

$$\begin{aligned} K_{ij} &= 4 + 1 \cdot (3+6) + 6 \cdot 3 \cdot 6 \bmod 11 = 0, \\ K_{ik} &= 4 + 1 \cdot (3+8) + 6 \cdot 3 \cdot 8 \bmod 11 = 5, \\ K_{jk} &= 4 + 1 \cdot (6+8) + 6 \cdot 6 \cdot 8 \bmod 11 = 9. \end{aligned}$$

In this case, all parameters were chosen at random by the Matlab function in Appendix A.2. In fact, the code computes a common key for every pair of nodes in the network, which is depicted in Fig. A.1.

We now prove that an adversary who compromised one node cannot determine the common key of two other nodes.

Proposition 5.1.2. *The Blom KPS is* 1*-unconditionally secure.*

Proof. Suppose that an adversary compromised the node N_k, which means that he knows the values a_{N_k} and b_{N_k}. We will show that this information is consistent with any possible value $K \in \mathbb{F}_p$ of the key K_{ij} (where $k \neq i, j$), because this would imply that the adversary cannot rule out any values for K_{ij}. Consider

$$\begin{pmatrix} 1 & r_{N_i} + r_{N_j} & r_{N_i} r_{N_j} \\ 1 & r_{N_j} & 0 \\ 0 & 1 & r_{N_j} \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} K \\ a_{N_k} \\ b_{N_k} \end{pmatrix}.$$

The determinant of the first matrix is equal to

$$r_{N_k}^2 - (r_{N_i} + r_{N_j}) r_{N_k} + r_{N_i} r_{N_j} = (r_{N_k} - r_{N_i})(r_{N_k} - r_{N_j}).$$

Since $r_{N_i} \neq r_{N_k} \neq r_{N_j}$ by construction, this determinant is non-zero in $\mathbb{F}_p$. Therefore, the equation has a unique solution for a, b and c in $\mathbb{F}_p$, which shows the desired result. □

On the other hand, figuring out the information stored in two nodes suffices to compromise every node in the network.

Proposition 5.1.3. *The Blom KPS with $S = 1$ can be broken by an adversary who compromised two nodes.*

Proof. If an adversary compromised two nodes N_i and N_j, then he knows the values

$$\begin{aligned} a_{N_i} &= a + b r_{N_i} \bmod p, & b_{N_i} &= b + c r_{N_i} \bmod p, \\ a_{N_j} &= a + b r_{N_j} \bmod p, & b_{N_j} &= b + c r_{N_j} \bmod p. \end{aligned}$$

Solving these four equations in three unknowns for a, b and c delivers $f(x, y)$, which allows the computation of each common key in the network. □

Example 12. Consider the network from Example 11 once more. Suppose that an adversary compromised the nodes N_i and N_j. Then,

$$\begin{aligned} a + 3b = a_{N_i} &= 4 + 1 \cdot 3 \bmod 11 = 7, \\ a + 6b = a_{N_j} &= 4 + 1 \cdot 6 \bmod 11 = 10, \\ b + 3c = b_{N_i} &= 1 + 6 \cdot 3 \bmod 11 = 8, \\ b + 6c = b_{N_j} &= 1 + 6 \cdot 6 \bmod 11 = 2. \end{aligned}$$

are known. The first two equations give $b \equiv 1 \pmod{11}$, so $b = 1$ since $b \in \mathbb{F}_{11}$. It then immediately follows from the other equations, say the first and the third, that $a = 4$ and $c = 6$, which were indeed the chosen parameters.

We will now generalize the Blom KPS to larger security parameters S. Definition 5.1.1 remains almost unchanged; the only thing that changes is the polynomial $f(x, y)$. Of course, it should still be symmetric.

Definition 5.1.4. *The* **Blom KPS** *with security parameter S works as follows:*

(1) A unique publicly known element $r_N \in \mathbb{F}_p$ is assigned to every node N.

(2) The trusted authority chooses random elements $a_{ij} \in \mathbb{F}_p$ where $a_{ij} = a_{ji}$ for all $0 \leq i, j \leq S$. A bivariate polynomial $f \in \mathbb{F}_p[x, y]$ is defined as

$$f(x, y) = \sum_{i=0}^{S} \sum_{j=0}^{S} a_{ij} x^i y^j \bmod p.$$

(3) All nodes N are preloaded with the coefficient vector of $f(x, r_N) \bmod p$.

(4) Two nodes N_i and N_j communicate with the common key

$$K_{ij} = f(r_{N_j}, r_{N_i}) \bmod p.$$

We observe that a_{00}, $a_{01} = a_{10}$ and a_{11} respectively correspond to the parameters a, b and c for the case $S = 1$. Note that every node N is preloaded with a vector of length $S + 1$ whose ith entry is given by

$$\sum_{j=0}^{S} a_{ij} r_N^j \bmod p.$$

We are left to prove the claimed S-unconditional security of the general Blom KPS. We will first show that $S + 1$ compromised nodes can break the scheme. Instead of modifying the attack from the case $S = 1$, we will make use of the well-known interpolation formula for polynomials. The following is a modification of the theorem and proof in [2]. For brevity, we write x/y instead of xy^{-1} for all $x, y \in \mathbb{F}_p$.

Theorem 5.1.5 (Lagrange's interpolation theorem)**.** *Let $x_1, x_2, \dots, x_n$ be distinct elements in the finite field $\mathbb{F}_p$ and let $y_1, y_2, \dots, y_n$ be not necessarily distinct elements in $\mathbb{F}_p$. Then there is a unique $P \in \mathbb{F}_p[x]$ of degree at most $n-1$ such that $P(x_i) = y_i$ for all $1 \leq i \leq n$. This polynomial is given by*

$$P(x) = \sum_{i=1}^{n} y_i \prod_{\substack{1 \leq j \leq n \\ i \neq j}} \frac{x - x_j}{x_i - x_j}.$$

Proof. Fix an $1 \leq i \leq n$. Since all elements x_j are distinct, we have that $x_i - x_j \neq 0$ for every $1 \leq j \leq n$ where $i \neq j$. Therefore, the polynomial

$$Q_i(x) = \prod_{\substack{1 \leq j \leq n \\ i \neq j}} \frac{x - x_j}{x_i - x_j}$$

is well-defined and has degree $n-1$. We observe that $Q_i(x_j) = 0$ if $i \neq j$, because one of the factors in the nominator will be zero. Also, $Q_i(x_i) = 1$ so it follows that P satisfies the requirements. Suppose that there is another polynomial $Q \in \mathbb{F}_p[x]$ with the same properties. Then, $P - Q$ is of degree $n-1$ and has n distinct roots, which is only possible if $P - Q = 0$ by the fundamental theorem of algebra. □

Lagrange's interpolation theorem also has a bivariate form, which is proven in precisely the same way.

Theorem 5.1.6. *Let $x_1, x_2, \dots, x_n$ be distinct elements in the finite field $\mathbb{F}_p$ and let $y_1(x), y_2(x), \dots, y_n(x)$ be not necessarily distinct polynomials in $\mathbb{F}_p[x]$ of degree at most $n-1$. Then there is a unique $P \in \mathbb{F}_p[x, y]$ of degree at most $n-1$ such that $P(x, x_i) = y_i(x)$ for all $1 \leq i \leq n$, namely*

$$P(x, y) = \sum_{i=1}^{n} y_i(x) \prod_{\substack{1 \leq j \leq n \\ i \neq j}} \frac{y - x_j}{x_i - x_j}.$$

It now becomes very straightforward to show that the Blom KPS is broken if $S+1$ nodes are compromised.

Proposition 5.1.7. *The Blom KPS can be broken by an adversary who compromised $S+1$ nodes.*

Proof. Suppose an adversary has compromised the nodes $N_1, \dots, N_{S+1}$, which means he knows the distinct elements $r_{N_1}, \dots, r_{N_{S+1}}$ and the polynomials $f(x, r_{N_i}) \bmod p$ for $1 \leq i \leq S+1$. Applying Theorem 5.1.6 delivers the desired polynomial $f(x, y)$. □

Example 13. Suppose we have a network of size $n = 10$, which uses the Blom KPS with $S = 3$, $p = 11$ and parameters

$$\begin{pmatrix} 6 & 10 & 6 & 10 \\ 10 & 7 & 9 & 1 \\ 6 & 9 & 8 & 7 \\ 10 & 1 & 7 & 5 \end{pmatrix},$$

where the entry in the ith row and jth column represents $a_{(i-1)(j-1)}$. Let the network contain four compromised nodes N_1, N_2, N_3 and N_4, whose publicly known identifiers are respectively $r_{N_1} = 0$, $r_{N_2} = 2$, $r_{N_3} = 4$ and $r_{N_4} = 6$. That means the adversary knows the polynomials

$$\begin{aligned} y_1(x) &= f(x, r_{N_1}) = 10x^3 + 6x^2 + 10x + 6, \\ y_2(x) &= f(x, r_{N_2}) = 3x^3 + 2x^2 + 2x + 9, \\ y_3(x) &= f(x, r_{N_3}) = 6x^3 + 2x^2 + 4x + 1, \\ y_4(x) &= f(x, r_{N_4}) = 6x^3 + x^2 + 9x. \end{aligned}$$

It is easy to verify this by hand or by using part of the code for the general Blom KPS in Appendix A.2. Polynomial interpolation then leads to

$$\begin{aligned} f(x, y) = {} & 5x^3y^3 + 7x^3y^2 + x^3y + 10x^3 + 7x^2y^3 + 8x^2y^2 + 9x^2y \\ & + 6x^2 + xy^3 + 9xy^2 + 7xy + 10x + 10y^3 + 6y^2 + 10y + 6, \end{aligned}$$

as shown in Appendix A.2. This polynomial corresponds precisely to the parameter matrix above, which means that the adversary is now able to intercept all communications in the network.

To end this section, we finally prove that compromising S nodes does not reveal any information about the common keys of uncompromised nodes. Of course, the same then also holds when less nodes have been compromised.

Proposition 5.1.8. *The Blom KPS is S-unconditionally secure.*

Proof. Suppose that an adversary has compromised the nodes $N_1, \dots, N_S$, which means he knows the polynomials $f(x, r_{N_k}) \bmod p$ for every $1 \leq k \leq S$. We will show that this information is consistent with any possible value of the common key K_{ij} of two uncompromised nodes N_i and N_j. Let $K \in \mathbb{F}_p$ be chosen at random and define the polynomial

$$g(x, y) = f(x, y) + (K - K_{ij}) \prod_{1 \leq k \leq S} \frac{(x - r_{N_k})(y - r_{N_k})}{(r_{N_i} - r_{N_k})(r_{N_j} - r_{N_k})}.$$

We note that g is symmetric, because both terms in its definition are symmetric. Also, $g(x, r_{N_k}) = f(x, r_{N_k})$ for all $1 \leq k \leq S$, and

$$g(r_{N_i}, r_{N_j}) = f(r_{N_i}, r_{N_j}) + K - K_{ij} = K.$$

Thus, for any possible value K of the key K_{ij} there is a polynomial g that is consistent with the information known to the adversary. It is therefore impossible for the adversary to rule out any values of K_{ij}. □

The major drawback of the Blom KPS is the sharp security threshold which must be specified, namely the security parameter S. However, it is optimal with respect to key storage, because it has been proven in [5] that any S-unconditionally secure KPS requires a key storage of at least $S + 1$.

5.2 Key distribution patterns

We now discuss the first combinatorial approaches for key predistribution without any computational overhead, dating from 1987 in [30]. We still consider a network with n nodes $N_1, \ldots, N_n$ and we would still like to obtain a complete network. The general idea is as follows: a trusted authority chooses m elements $K_1, \ldots, K_m$ from an additive abelian group G and assigns a different subset of these keys to each node, such that any pair of nodes has some keys in common. The latter requirement allows each two nodes to determine a common key for communication.

Definition 5.2.1. *A* **key distribution pattern** *(KDP) is a public $m \times n$-matrix M with binary entries, which specifies which nodes are to receive which keys. Namely, the node N_j is given the key K_i if and only if $M(i, j) = 1$.*

We denote $S_j = \{\, 1 \leq i \leq m \mid M(i, j) = 1 \,\}$ for any $1 \leq j \leq n$, i.e. S_j is the index set of the keys assigned to the node N_j. For any set of nodes N, we define

$$S(N) = \bigcap_{N_j \in N} S_j \quad \text{and} \quad K_N = \sum_{j \in S(N)} K_j$$

if $S(N) \neq \emptyset$, where the addition takes place in G. We call K_N the **group key**, because each node in N can compute it independently. Indeed, $S(N)$ contains precisely those indices of the keys that every node in N stores.

Note that sets N consisting of only two nodes N_i and N_j are actually sufficient for our purposes and in that case we will denote $K_{ij} = K_{\{N_i, N_j\}}$ as usual. However, the original framework of KDPs permits the construction of group keys for larger node collections, so we will work in this extended setting.

Proposition 5.2.2. *Let N and P be two sets of nodes. Then, an adversary who compromised every node in P can compute K_N if and only if*

$$S(N) \subseteq \bigcup_{N_j \in P} S(N_j). \tag{5.1}$$

Proof. If there exists a node $N_j \in N \cap P$, then the adversary can determine K_N by testing all summations of the keys of N_j, so we may assume that $N \cap P = \emptyset$. It is clear that the adversary can compute K_N if (5.1) holds, because the compromised nodes collectively hold all the required keys. Conversely, if (5.1) does not hold, then there is an element $i \in S(N)$ such that

$$i \notin \bigcup_{N_j \in P} S(N_j).$$

Since K_N is a sum of which one of the terms is K_i, the adversary has no information about the group key's value. □

Example 14. Suppose that $n = 4$, $m = 6$ and the KDP is

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

We then have $S_3 = \{2, 4, 6\}$ and $S_4 = \{3, 5, 6\}$, hence $K_{34} = K_6$. Similarly, we obtain $K_{12} = K_1$, $K_{13} = K_2$, $K_{14} = K_3$, $K_{23} = K_4$ and $K_{24} = K_5$.

It is possible to extend Example 14 to an arbitrary number of n nodes. Indeed, it is always possible to construct an $\binom{n}{2} \times n$-matrix such that any two nodes in the resulting KDP have exactly one common key, which acts as the group key. Such a KDP requires every node to store $n - 1$ keys and is perfectly resilient to node compromise. However, this is just the complete pairwise KPS from Example 2 in disguise, so we have not gained anything. We therefore try to find certain combinatorial properties for the KDP such that it becomes perfectly resilient against the compromise of S nodes where $1 \leq S \leq n$, similar to the Blom KPS.

Definition 5.2.3. *A* **Fiat–Naor S-KDP** *is a KDP whose rows are all possible binary vectors of length n with at most S zeros.*

Note that a Fiat–Naor S-KDP is thus an $m \times n$-matrix, where $m = \sum_{i=0}^{S} \binom{n}{i}$. By definition, any set of at least $n - S$ nodes is assigned a unique key.

Proposition 5.2.4. *Given a Fiat–Naor S-KDP, there exists a group key for any set of nodes N, such that N is perfectly resilient to the compromise of at most S other nodes.*

Proof. Let P be a set of compromised nodes such that $N \cap P = \emptyset$ and $|P| \leq S$. That means $|P^{\mathrm{c}}| \geq n - S$, so there exists a key K_i that is only given to the nodes in P^{c}. Since $N \subseteq P^{\mathrm{c}}$, all nodes in N were assigned K_i and we see that (5.1) is not satisfied. □

Example 15. For $n = 6$ and $S = 1$, we have $m = \binom{6}{0} + \binom{6}{1} = 7$ and the Fiat–Naor 1-KDP is then the $m \times n$-matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Suppose that $P = \{N_1, N_2, N_3\}$ is a set of uncompromised nodes. Their group key is $K_P = K_1 + K_2 + K_3 + K_4$ and no other node can compute this value, since none of them is assigned K_2, K_3 and K_4 simultaneously.

Definition 5.2.5. *A* **Mitchell–Piper** (t, S)**-KDP** *is a KDP that yields a group key for every set N of exactly t nodes, such that N is perfectly resistent to the compromise of at most S nodes.*

We show that such KDPs exist with the **probabilistic method**. This approach was initiated by Paul Erdős and is covered in full in [1]. The general idea is as follows: in order to prove the existence of a combinatorial structure with certain properties, one constructs an appropriate probability space and shows that a randomly chosen element in this space has the desired properties with positive probability.

Definition 5.2.6. *A set system $(X, \mathscr{B})$ is called a (t, S)***-cover-free family** *if for any disjoint $N, P \subset \mathscr{B}$ with $|N| = t$ and $|P| = S$ we have*

$$\bigcap_{B_i \in N} B_i \not\subseteq \bigcup_{B_j \in P} B_j.$$

*For brevity, we write that $(X, \mathscr{B})$ is a (t, S)-*CFF(m, n).

The next theorem links cover-free families and Mitchell–Piper KDPs.

Theorem 5.2.7. *Let M be an KDP of dimensions $m \times n$. Then, M is a Mitchell–Piper (t, S)-KDP if and only if M is the incidence matrix of a (t, S)-*CFF(m, n).

Proof. Let M be a Mitchell–Piper (t, S)-KDP. Consider the set system $(X, \mathscr{B})$ where $X = \{K_1, \ldots, K_m\}$ and $\mathscr{B} = \{S_1, \ldots, S_n\}$. The definitions imply that the incidence matrix of $(X, \mathscr{B})$ is precisely M. Also, $(X, \mathscr{B})$ is a (t, S)-CFF(m, n), since for any $N, P \subset \mathscr{B}$ where $|N| = t$ and $|P| = S$ we have

$$\bigcap_{S_i \in N} S_i = S(N) \not\subseteq \bigcup_{S_j \in P} S_j$$

because of Proposition 5.2.2. The converse follows directly from Proposition 5.2.2. □

This connection with cover-free families is very interesting, because there are many known constructions for the latter in the literature. However, we restrict ourselves to a non-constructive existence proof as mentioned earlier.

Theorem 5.2.8. *A (t, S)-*CFF(m, n) *exists if*

$$m > \frac{(t+S)\log n}{-\log p_{t,S}} \quad \textit{where} \quad p_{t,S} = 1 - \frac{t^t S^S}{(t+S)^{t+S}}.$$

Proof. Let M be an $m \times n$-matrix whose columns are labeled $1, \ldots, n$. An entry of M is defined to be 1 with probability ρ and 0 with probability $1 - \rho$. We will later determine a specific optimal value for $0 < \rho < 1$. Suppose that $N, P \subseteq \{1, \ldots, n\}$, where $|N| = t$, $|P| = S$ and $N \cap P = \emptyset$. We say that a row i of M satisfies the property $\gamma(N, P, i)$ if the entries of the columns in respectively N and P are all 1 and 0. Next, we define the random variable X as

$$X(N, P) = \begin{cases} 0 & \text{if } \gamma(N, P, i) \text{ is satisfied for a certain row } i, \\ 1 & \text{otherwise.} \end{cases}$$

We observe the following equivalent statements:

(1) M is the incidence matrix of a (t, S)-CFF(m, n)
(2) for all N and P, we have $\bigcap_{B_i \in N} B_i \not\subseteq \bigcup_{B_j \in P} B_j$
(3) for all N and P, there exists an $x_l \in \bigcap_{B_i \in N} B_i$ such that $x_l \notin B_j$ for all $B_j \in P$
(4) for all N and P, $\gamma(N, P, l)$ is satisfied for a certain $1 \leq l \leq m$, i.e. $X(N, P) = 0$.

For fixed subsets N and P, the probability that $\gamma(N, P, i)$ does not hold for any $1 \leq i \leq m$ is equal to

$$\mathbb{E}[X(N, P)] = \left(1 - \rho^t (1 - \rho)^S\right)^m.$$

In order to minimize $\mathbb{E}[X(N,P)]$, we have to maximise $1-\rho^t(1-\rho)^S$, so we should choose ρ such that

$$t\rho^{t-1}(1-\rho)^S - \rho^t S(1-\rho)^{S-1} = 0,$$

that is $\rho = t/(t+S)$. We define $I = \left\{ (N,P) \,\middle|\, |N| = t, |P| = S, N \cap P = \emptyset \right\}$ and

$$X = \sum_I X(N,P).$$

The above equivalences imply that M is the incidence matrix of a (t,S)-CFF(m,n) if and only if $X = 0$. It is sufficient to choose values for all parameters such that $\mathbb{E}[X] = \mathbb{P}(X=1) < 1$, since then $\mathbb{P}(X=0) > 0$. We compute

$$\begin{aligned} \mathbb{E}[X] &= \sum_I \mathbb{E}[X(N,P)] = \binom{n}{t}\binom{n-t}{S}\left(1-\rho^t(1-\rho)^S\right)^m \\ &= \binom{n}{t}\binom{n-t}{S}\left(1-\frac{t^t S^S}{(t+S)^{t+S}}\right)^m \le n^{t+S}\left(1-\frac{t^t S^S}{(t+S)^{t+S}}\right)^m. \end{aligned}$$

Writing $p_{t,S}$ as in the statement, we obtain that $\mathbb{E}[X] < 1$ if and only if

$$(t+S)\log n + m \log p_{t,S} < 0,$$

which is precisely the desired condition after reordering. □

5.3 Random key predistribution schemes

After the discovery of many elegant combinatorial KPSs, such as the Blom KPS, the interest in key predistribution faded. The evolution of wireless technologies in recent years led to a resurgence of research and completely different approaches. In 2002, Eschenauer and Gligor presented the first random KPS in [15]. Their scheme is pretty straightforward, but provides good trade-offs for many WSN scenarios.

Definition 5.3.1. *The* **Eschenauer–Gligor KPS** *works as follows. A key pool K of keys is generated from the space of all possible keys. Each node is independently assigned a uniformly random subset of k distinct keys from K.*

If nodes have more than one key in common, then they should select a single one at random to use for communication. For further analysis later on, we denote d for the maximum number of common keys that two nodes may use to secure their communications. Thus, in general $q \le d \le k$, and for the Eschenauer–Gligor KPS we have $d = q = 1$. We first compute the probability that two nodes are connected.

Proposition 5.3.2. *For the Eschenauer–Gligor KPS with key pool K, we have*

$$\mathrm{Pr}_1 = 1 - \frac{\binom{|K|-k}{k}}{\binom{|K|}{k}}.$$

Proof. Suppose that two nodes N_i and N_j respectively store the key sets S_i and S_j. Then, the probability that they share a key is equal to

$$\mathrm{Pr}_1 = 1 - \mathbb{P}(S_i \cap S_j = \emptyset).$$

If $S_i \cap S_j = \emptyset$, then the k keys in S_j must have been picked from $K \setminus S_i$, for which there are $\binom{|K|-k}{k}$ ways. There are $\binom{|K|}{k}$ ways to pick S_i, so the result follows. □

The networks that result from the Eschenauer–Gligor KPS are not necessarily complete, since $\mathrm{Pr}_1 < 1$. Even worse, they may be disconnected. This is demonstrated in Appendix A.3. Based on the results of Erdős and Rényi on random graphs in [14], it is however possible to choose the parameters $|K|$ and n in such a way that a network becomes connected with a high probability. It is even stated in [20] that random KPSs provide key graphs with good expansion with high probability.

In their paper [15], Eschenauer and Gligor do not give a formula for fail_s. Instead, they simulate a network with $n = 1000$, $k = 40$ and $|K| = 10{,}000$, and observe that only 50% of the keys from the key pool were used to secure links: 30% were used to secure a single link, 10% to secure two links and 5% to secure three links. Thus, the compromise of a single key compromises one other link with probability 0.1.

Proposition 5.3.3. *For the Eschenauer–Gligor KPS with key pool K, we have*

$$\mathrm{fail}_s = 1 - \left(1 - \frac{k}{|K|}\right)^s \quad \textit{for all } 1 \leq s \leq n-2.$$

Proof. Let N_i and N_j be two uncompromised nodes that share a key K_{ij}. Suppose that the adversary has compromised the nodes N_l for $1 \leq l \leq s$, which respectively store the uniformly random sets $S_l \subset K$ of size k. Note that

$$\mathbb{P}(K_{ij} \notin S_1) = \frac{\binom{|K|-1}{k}}{\binom{|K|}{k}} = 1 - \frac{k}{|K|}.$$

Denoting $S = S_1 \cup \cdots \cup S_s$, we obtain

$$\mathrm{fail}_s = \mathbb{P}(K_{ij} \in S) = 1 - \mathbb{P}(K_{ij} \notin S) = 1 - \mathbb{P}(K_{ij} \notin S_1)^s,$$

where we used the independence of the sets S_l in the last equality. □

Since the Eschenauer–Gligor KPS uses the same key pool K for all nodes, the security of the network gradually erodes as keys from K are compromised by an adversary. That motivated the development of random KPSs with better resilience. In [9], Chan et al. extended the Eschenauer–Gligor KPS.

Definition 5.3.4. *The* q**-composite KPS** *is identical to the Eschenauer–Gligor KPS, except that it requires nodes to have at least $q > 1$ common keys in order to be connected and $d = k$.*

Suppose that two nodes N_i and N_j have c common keys $K_1, \ldots, K_c$, where $q \leq c \leq k$. Their communication key K_{ij} can then be computed with an appropriate function h such as a hash function (see [39] for more on hashes), that is

$$K_{ij} = h(K_1, \ldots, K_c),$$

such that an adversary needs to learn $c > 1$ keys in order to compromise the link. Intuitively, the resilience in the q-composite KPS is thus better than in the Eschenauer–Gligor KPS where $q = d = 1$. However, the connectivity is worse for the same key pool size and key storage, because nodes are less likely connected in the case $q > 1$. We will formally prove the latter statement first.

Proposition 5.3.5. *In a q-composite scheme with key pool K, we have*

$$\Pr_1 = 1 - \sum_{i=0}^{q-1} p(i) \quad \textit{where} \quad p(i) = \frac{\binom{|K|-k}{k-i}\binom{k}{i}}{\binom{|K|}{k}}$$

is the probability that two nodes share exactly i keys.

Proof. Two nodes N_1 and N_2 do not have a secure connection if they share i keys where $0 \leq i \leq q-1$. In order to compute the probabilities $p(i)$, we fix the i keys from N_1 that N_2 shares. That means N_2 has $k - i$ keys chosen from the $|K| - k$ keys that are unknown to N_1. There are $\binom{|K|-k}{k-i}$ ways to do this, out of the $\binom{|K|}{k}$ ways to choose keys for N_2. The result follows from the fact that there are $\binom{k}{i}$ ways to fix the i keys from N_1. □

Note that the formula in Proposition 5.3.5 agrees with the one in Proposition 5.3.2. Indeed, if $q = 1$ then we obtain

$$\Pr_1 = 1 - p(0) = 1 - \frac{\binom{|K|-k}{k}}{\binom{|K|}{k}}.$$

We now present a formula for the resilience in the q-composite KPS, following [18]. Actually, this formula for fail_s holds for any random KPS with a construction similar to the q-composite KPS. Specifically, it applies to all random KPSs where

(1) each node is allocated k keys, which are selected independently, uniformly at random and without replacement from a key pool;

(2) nodes may only establish a common key if they share at least $q \geq 1$ keys;

(3) if two nodes share more than d keys where $q \leq d \leq k$, then they should randomly pick d of their keys for the computation of their communication key;

(4) the function for producing the communication key from c common keys is such that an adversary must know all c keys to compromise the link.

Proposition 5.3.6. *Any random KPS that fulfils the above conditions (1)-(4) has a resilience given by*

$$\begin{aligned}\text{fail}_s = \frac{1}{\Pr_1} \sum_{c=q}^{d} \left(1 - \sum_{i=1}^{c} (-1)^{i-1} \binom{c}{i} \frac{\binom{|K|-i}{k}^s}{\binom{|K|}{k}^s} \right) p(c) + \\ \frac{1}{\Pr_1} \left(1 - \sum_{i=1}^{d} (-1)^{i-1} \binom{d}{i} \frac{\binom{|K|-i}{k}^s}{\binom{|K|}{k}^s} \right) \sum_{c=d+1}^{k} p(c).\end{aligned}$$

Proof. Consider a pair of uncompromised nodes with c common keys $K_1, \ldots, K_c$, where $q \leq c \leq d$. The probability that all these keys are known to an adversary who compromised s nodes $N_1, \ldots, N_s$ is

$$\mathbb{P}\big(\{K_1, \ldots, K_c\} \subset S \big) = 1 - \mathbb{P}\Big(\bigcup_{i=1}^{c} \{K_i \notin S\} \Big),$$

where S has the same meaning as in the proof of Proposition 5.3.3. By the law of inclusion and exclusion, we get

$$\begin{aligned}\mathbb{P}\big(\{K_1, \ldots, K_c\} \subset S \big) &= 1 - \sum_{i=1}^{c} (-1)^{i+1} \binom{c}{i} \mathbb{P}\big(\{K_1, \ldots, K_i\} \not\subset S\big) \\ &= 1 - \sum_{i=1}^{c} (-1)^{i+1} \binom{c}{i} \mathbb{P}\Big(\bigcap_{j=1}^{s} \{K_1, \ldots, K_i\} \not\subset S_j \Big) \\ &= 1 - \sum_{i=1}^{c} (-1)^{i+1} \binom{c}{i} \frac{\binom{|K|-i}{k}^s}{\binom{|K|}{k}^s}.\end{aligned}$$

The probability of the randomly chosen uncompromised nodes is $p(c)/\Pr_1$, so

$$\text{fail}_s = \frac{1}{\Pr_1} \sum_{c=q}^{d} \left(1 - \sum_{i=1}^{c} (-1)^{i-1} \binom{c}{i} \frac{\binom{n-i}{k}^s}{\binom{n}{k}^s} \right) p(c)$$

for $q \leq c \leq d$. If $d < c \leq k$, then the probability of two connected nodes sharing c keys is also $p(c)/\Pr_1$, but only d keys will be used for the computation of a common key. We already know that

$$\mathbb{P}\big(\{K_1, \ldots, K_d\} \subset S \big) = 1 - \sum_{i=1}^{d} (-1)^{i+1} \binom{d}{i} \frac{\binom{|K|-i}{k}^s}{\binom{|K|}{k}^s}$$

from the previous calculating, so we obtain for $d < c \leq k$ that

$$\mathrm{fail}_s = \frac{1}{\Pr_1}\left(1 - \sum_{i=1}^{d} (-1)^{i-1} \binom{d}{i} \frac{\binom{n-i}{k}^s}{\binom{n}{k}^s}\right) \sum_{c=d+1}^{k} p(c).$$

Adding the two results gives the final formula for fail_s. □

We can easily verify that this formula agrees with the one given in Proposition 5.3.3: for $q = d = 1$, we obtain

$$\begin{aligned}\mathrm{fail}_s &= \frac{1}{\Pr_1}\left(1 - \frac{\binom{|K|-1}{k}^s}{\binom{|K|}{k}^s}\right) p(1) + \frac{1}{\Pr_1}\left(1 - \frac{\binom{|K|-1}{k}^s}{\binom{|K|}{k}^s}\right) \sum_{c=2}^{k} p(c) \\ &= 1 - \left(1 - \frac{k}{|K|}\right)^s,\end{aligned}$$

where the last equality follows from $\sum_{c=1}^{k} p(c) = \Pr_1$ by definition of $\Pr_1$. The formula for the resilience of the q-composite KPS also follows immediately.

Corollary 5.3.7. *For the q-composite KPS with key pool K, we have*

$$\mathrm{fail}_s = \frac{1}{\Pr_1} \sum_{c=q}^{k} \left(1 - \sum_{i=1}^{c} (-1)^{i-1} \binom{c}{i} \frac{\binom{|K|-i}{k}^s}{\binom{|K|}{k}^s}\right) p(c).$$

Proof. Setting $d = k$ in Proposition 5.3.6, we observe that the summation from $c = k+1$ to $c = k$ in the second term vanishes, leaving only the first term. □

5.4 Combinatorial designs

The major drawback of random key predistribution is that it cannot guarantee with absolute certainty that the network satisfies desired properties, such as connectedness or good expansion. Deterministic KPSs, on the other hand, do have this advantage, which often facilitates the analysis. This provoked a second rise of combinatorial key predistribution [28]. Many of the proposed deterministic KPSs are based on combinatorial designs that have the property of being a configuration.

Definition 5.4.1. *A design $(X, \mathscr{B})$ where $|X| = n$ and $|\mathscr{B}| = m$ is called an (n, m, r, k)*-**configuration** *if it is r-regular and k-uniform, and any pair of distinct points occurs in at most one block.*

Configurations allow some flexibility, because their graphs are not necessarily complete, contrary to the more restrictive KDPs.

Proposition 5.4.2. *The graph of an (n, m, r, k)-configuration is $k(r-1)$-regular.*

Proof. Let v be a vertex in the graph of a configuration, which corresponds to the block B. By definition, B is of size k and any point $x \in B$ appears in $r-1$ other blocks. Hence, it follows that $\deg(v) = k(r-1)$. □

Proposition 5.4.3. *For an (n, m, r, k)-configuration, we have $\varepsilon \approx k(r-1)/2$ if m is large.*

Proof. We know from Proposition 5.4.2 that any point is $k(r-1)$-regular. Thus, for every $S \subset V$ we have $E(S, S^c) = |S|k(r-1) - 2E(S, S)$. We now try to approximate $E(S, S)$. Let v_i and v_j be two vertices in S, and denote S_i and S_j for the key sets of their corresponding nodes N_i and N_j. We write $S_i = \{K_1, \dots, K_k\}$ and define the random variables

$$X_l = \begin{cases} 1 & \text{if } K_l \in S_j, \\ 0 & \text{otherwise,} \end{cases}$$

for $1 \le l \le k$. Then $X = \sum_{l=1}^{k} X_l$ is the number of edges between v_i and v_j. Since $\mathbb{P}(K_l \in S_j) = (r-1)/(m-1)$, the linearity of expectation gives

$$\mathbb{E}[X] = \sum_{l=1}^{k} \mathbb{P}(K_l \in S_j) = \frac{k(r-1)}{m-1}.$$

The expected number of edges with endpoints in S is thus $\binom{|S|}{2}\frac{k(r-1)}{m-1}$, so we obtain

$$\begin{aligned} \varepsilon &\approx \min\left\{ k(r-1) - \frac{2}{|S|}\binom{|S|}{2}\frac{k(r-1)}{m-1} \;\middle|\; S \subset V, 1 \le |S| \le \frac{m}{2} \right\} \\ &= \min\left\{ k(r-1)\left(1 - \frac{|S|-1}{m-1}\right) \;\middle|\; S \subset V, 1 \le |S| \le \frac{m}{2} \right\} \\ &= k(r-1)\left(1 - \frac{\lfloor \frac{m}{2} \rfloor - 1}{m-1}\right) \approx \frac{k(r-1)}{2}, \end{aligned}$$

The approximation in the last line holds for large m, because $(\lfloor \frac{m}{2} \rfloor - 1)/(m-1)$ converges to $1/2$ as $m \to \infty$. □

In practice $k(r-1)/2 > 1$, so KPSs that are based on configurations have a large expected expansion coefficient. However, the following example illustrates that good expansion is not guaranteed.

Example 16. Consider the $(6, 6, 2, 2)$-configuration whose graph is given in Fig. 5.1. Since the graph is disconnected, we have $\varepsilon = 0$, so the configuration is no expander.

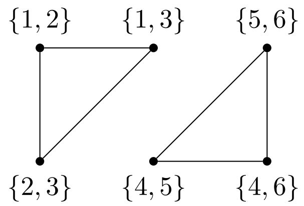


Figure 5.1: A disconnected configuration graph.

Example 16 shows that we require connected configuration graphs if we want to construct KPSs that are based on configurations. In order to guarantee this connectedness, two new classes of configurations were introduced: μ-common intersection designs and strongly regular graphs.

Definition 5.4.4. *An (n, m, r, k)-configuration $(X, \mathscr{B})$ is called a μ-***common intersection design** *if for any two disjoint blocks B_i and B_j we have*

$$|\{\, B_k \in \mathscr{B} \mid B_i \cap B_k \neq \emptyset \text{ and } B_j \cap B_k \neq \emptyset \,\}| \geq \mu.$$

μ-common intersection designs were introduced by Lee and Stinson in [23, 24], specifically for key predistribution purposes. The idea is that nodes, which are disconnected in the key graph, may benefit from having at least μ common adjacent nodes, since they are then able to communicate via two hops. The following proposition further supports the use of μ-common intersection designs for building KPSs.

Proposition 5.4.5. *The graph of any μ-common intersection design has $\varepsilon \geq 1$.*

Proof. Let $S \subset V$ be chosen such that $1 \leq |S| \leq |V|/2$. It is sufficient to show that $|E(S, S^{\mathrm{c}})| \geq |S|$, because we then have

$$\varepsilon \geq \min \left\{ \frac{|S|}{|S|} \;\middle|\; S \subset V,\, 1 \leq |S| \leq \frac{|V|}{2} \right\} = 1$$

as desired. Suppose for a contradiction that $|E(S, S^{\mathrm{c}})| < |S|$. That means there exists a vertex $v \in S$ that is not adjacent to S^{c}. Denoting the set of vertices adjacent to v by B, we get that $E(B, S^{\mathrm{c}}) \subseteq E(S, S^{\mathrm{c}})$, so

$$|E(B, S^{\mathrm{c}})| \leq |E(S, S^{\mathrm{c}})| < |S| \leq |S^{\mathrm{c}}|.$$

This implies the existence of a vertex $w \in S^{\mathrm{c}}$ that is not adjacent to B. But then v and w do not have a common neighbour, which is impossible by definition of a μ-common intersection design. □

We now give a brief introduction to the important class of strongly regular graphs, restricting ourselves to the tools we need to find a lower bound for the expansion coefficient of such graphs. More interesting results can be found in [16].

Definition 5.4.6. *An* (n, r, μ, ν)**-strongly regular graph** *is an* r*-regular graph with* n *vertices such that any two distinct vertices have* μ *common adjacent vertices if they are not adjacent and* ν *common adjacent vertices if they are adjacent.*

A strongly regular graph may be regarded as the graph of a μ-common intersection design $(X, \mathscr{B})$ with the extra condition

$$|\{\, B_k \in \mathscr{B} \mid B_i \cap B_k \neq \emptyset \text{ and } B_j \cap B_k \neq \emptyset \,\}| \geq \nu$$

for any two blocks B_i and B_j such that $B_i \cap B_j \neq \emptyset$.

Example 17. Two easy examples of strongly regular graphs are C_5 and the well-known Petersen graph, which is depicted in Fig. 5.2. Indeed, their parameters are respectively $(5, 2, 1, 0)$ and $(10, 3, 1, 0)$.

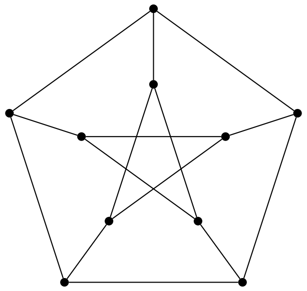


Figure 5.2: The Petersen graph is strongly regular.

The parameters of strongly regular graphs are not independent: the following proposition establishes a so-called **feasibility condition**, which is an equation that must be satisfied by the graph's parameters.

Proposition 5.4.7. *For any* (n, r, μ, ν)*-strongly regular graph* G*, we have*

$$r(r - \nu - 1) = \mu(n - r - 1).$$

Proof. Let $v \in V(G)$, and write B for its set of adjacent vertices and C for its set of non-adjacent vertices. By definition, we have $|B| = r$ and $|C| = n - r - 1$. The desired equality follows by counting the edges between B and C in two ways. On the one hand, every vertex in B is adjacent to ν vertices in B, hence adjacent to $r - \nu - 1$ vertices in C. Therefore, $|E(B, C)| = r(r - \nu - 1)$. On the other hand, every vertex in C is adjacent to μ vertices in B, so we also have $|E(B, C)| = \mu(n - r - 1)$. □

Example 18. We can easily check that the feasibility condition from Proposition 5.4.7 holds for C_5 and the Petersen graph. Indeed, we respectively obtain

$$\begin{aligned}&2 \cdot (2-0-1) = 2 = 1 \cdot (5-2-1) \quad \text{and}\\ &3 \cdot (3-0-1) = 6 = 1 \cdot (10-3-1).\end{aligned}$$

By constructing a list of parameters that meet this feasibility condition, we can narrow down the possible candidates for strongly regular graphs. We will present another feasibility condition later on.

Lemma 5.4.8. *For any graph G, the eigenvectors with different eigenvalues are orthogonal with respect to the scalar product.*

Proof. Let A be the adjacency matrix of G. Suppose that $Au = \alpha u$ and $Av = \beta v$ with $\alpha \neq \beta$. Since A is symmetric, we get that

$$\beta u^T v = u^T Av = (v^T Au)^T = \alpha u^T v.$$

This is only possible if $u^T v = 0$, i.e. when u and v are orthogonal. □

Theorem 5.4.9. *The non-trivial eigenvalues of an (n, r, μ, ν)-strongly regular graph are solutions of the following equation with variable x:*

$$x^2 - (\nu - \mu)x - (r - \mu) = 0.$$

Proof. Let A be the adjacency matrix of a strongly regular graph G. The entry a_{vw} of A^2 is the number of walks of length two from $v \in V(G)$ to $w \in V(G)$. In a strongly regular graph, this number only depends on whether v and w are equal, adjacent or distinct and non-adjacent, so we get that

$$A^2 = rI + \nu A + \mu(J - I - A) = (\nu - \mu)A + (r - \mu)I + \mu J,$$

where I is the identity matrix of dimension n and J is the matrix of dimension n whose entries are all equal to 1. For any eigenvector u of A with eigenvalue $\theta \neq r$, we then obtain

$$\theta^2 u - (\nu - \mu)\theta u - (r - \mu)u = A^2 u - (\nu - \mu)Au - (r - \mu)Iu = \mu Ju = 0,$$

where the last equality follows from Proposition 3.2.5 and Lemma 5.4.8. □

Corollary 5.4.10. *For an* (n, r, μ, ν)*-strongly regular graph, we have*

$$\varepsilon \geq \frac{r}{2} - \frac{\nu - \mu + \sqrt{(\nu - \mu)^2 + 4(r - \mu)}}{4}.$$

Proof. Theorem 5.4.9 implies that the second largest eigenvalue of the graph is

$$\lambda_2 = \frac{\nu - \mu + \sqrt{(\nu - \mu)^2 + 4(r - \mu)}}{2}.$$

By Theorem 3.2.9, we have $\varepsilon \geq (r - \lambda_2)/2$ and that gives the desired result. □

Example 19. Filling in the parameters of the Petersen graph in the inequality from Corollary 5.4.10 gives $\varepsilon \geq 1$. If we let S be set of the vertices that form the 'outer ring' C_5, then $\varepsilon \leq |E(S, S^c)|/|S| = 5/5 = 1$. We may thus conclude that $\varepsilon = 1$.

The following proposition yields a second feasibility test, which is very useful in practice. Although it does not give us an explicit construction method for strongly regular graphs, it considerably reduces the list of possible parameters.

Proposition 5.4.11. *For an* (n, r, μ, ν)*-strongly regular graph, the two evaluations of the following expression are integral:*

$$\frac{1}{2}\Big((n - 1) \pm \frac{2r + (n - 1)(\nu - \mu)}{\sqrt{(\nu - \mu)^2 + 4(r - \mu)}}\Big).$$

Proof. We know from Theorem 5.4.9 that the non-trivial eigenvalues of a (n, r, μ, ν)-strongly regular graph are

$$\theta_1 = \frac{\nu - \mu - \sqrt{D}}{2} \quad \text{and} \quad \theta_2 = \frac{\nu - \mu + \sqrt{D}}{2},$$

where $D = (\nu - \mu)^2 - 4(r - \mu)$. Let m_1 and m_2 be the multiplicities of θ_1 and θ_2 respectively. Since $r + m_1\theta_1 + m_2\theta_2 = \mathrm{Tr}(A) = 0$ and $m_1 + m_2 = n - 1$, we obtain

$$m_1 = \frac{(n - 1)\theta_2 + r}{\theta_2 - \theta_1} \quad \text{and} \quad m_2 = \frac{(n - 1)\theta_1 + r}{\theta_1 - \theta_2}.$$

Substituting the values of θ_1 and θ_2 gives the expression in the statement. □

Example 20. Consider $(n, r, \mu, \nu) = (11, 4, 2, 0)$. The equality from Proposition 5.4.7 is fulfilled, because $4 \cdot (4 - 0 - 1) = 12 = 2 \cdot (11 - 4 - 1)$. However, $(\nu - \mu)^2 + 4(r - \mu) = 12$ is not a square, so the evaluation of the expressions in Proposition 5.4.11 cannot be integral. Therefore, a $(11, 4, 2, 0)$-strongly regular graph does not exist.

Discussing known constructions for strongly regular graphs would lead us too far, so we refer the interested reader to [16] and the references in [20].

Chapter 6

Conclusions

We have seen that a high expansion coefficient ε is certainly desirable in a network. Therefore, this invariant should be taken into account in the design of KPSs with WSN applications. Its use is further justified by the fact that many existing KPSs, which provide good trade-offs, already yield networks with good expansion.

However, there is still some room for improvement. In practice, it is namely very difficult to compute ε for large networks and even when its exact value is known for the network's key graph, the expansion of the intersection graph cannot be predicted if the communication graph is modelled as a random graph. Fortunately, ε can be approximated with an upper and lower bound, and the other issue can of course be resolved in applications where the position of deployed nodes is traceable. A possibly more concerning limitation of ε is that it only reflects the weakest point of a network, without providing any information about the network's structure elsewhere. This is illustrated in Fig. 6.1: the three cases are equally evaluated with $\varepsilon = 0$, although the network's topologies are completely different.

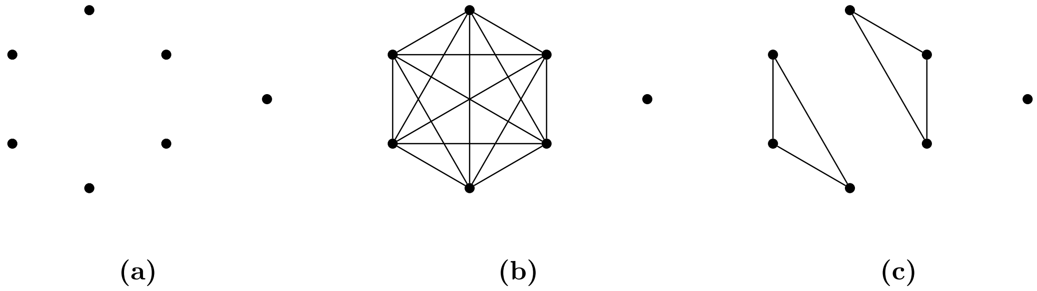


Figure 6.1: Three cases where $\varepsilon = 0$.

Especially situations like Fig. 6.1b, where only a few nodes are disconnected from the rest of the network, which otherwise has good expansion, may be perfectly acceptable in networks with hundreds or thousands of nodes. It therefore seems interesting to investigate whether something can be gained with graph invariants that also take into account the number of connected components, for example a weighted average of the expansion coefficients of all connected components.

The study of key predistribution gave birth to many interesting proposals for KPSs, mostly targeted at WSN applications. We presented several of the most trend-setting ones, both deterministic and random. While some KPSs in the literature achieve in practice good trade-offs between key storage, connectivity and resilience, a better understanding of these compromises is absolutely required. Indeed, the added value of many new KPSs is only motivated with a comparison against a limited number of previous proposals and some simulations, for example [6]. Even worse, papers like [38] essentially base a KPS on an expander graph construction, assign unique keys to all edges and then claim to have proposed a valuable KPS without much further ado. This approach might suffice for engineering disciplines, but from a mathematical point of view, it is questionable whether much is added to our knowledge of key predistribution.

Recent research of KPS construction techniques and a deeper exploration of desirable network features on a mathematical basis (e.g. [19, 20, 22]) are definitely a step in the right direction. It is for example suggested in [20] that KPSs based on hypergraphs with good expansion, of which currently little is known, may be very promising.

Appendix A

Implementations in Matlab

The code in this appendix is written by the author in Matlab R2016a, but should also be usable in older versions. Before presenting the code for zig-zag products and several KPSs, we provide a way to plot weighted graphs and networks in general. These functions require at least Matlab R2015b and will be used in multiple places later on.

```
function [ P ] = WeightedGraph( A )
%WEIGHTEDGRAPH Plot the weighted graph of the provided adjacency matrix.
fig=figure; clf
fig.Color='white';
fig.Position(3)=500;
fig.Position(4)=500;
axes('position',[0 0 1 1]);
P=plot(graph(A));
P.NodeLabel={};
P.NodeColor='black';
P.EdgeColor='black';
W=graph(A).Edges.Weight;
for i=1:size(W,1)
    if W(i)>1
        labeledge(P,i,W(i));
    end
end
axis off square tight;
%Uncomment the following line to export the figure as a png file.
%print('weightedgraph','-dpng','-r600')
end
```

Note that we had to fall back on weighted graphs, because Matlab R2016a does not support multiple edges. Although not very pleasing from an aesthetic point of view, these graphs can certainly help to visually investigate the structure of large adjacency matrices, and they are particularly useful to quickly discover isomorphic graphs due to the way Matlab constructs them. The following function outputs a graph where each edge is labeled with the common key of its endpoints.

```
function [ P ] = PlotNetwork( A,L )
%PLOTNETWORK Plot the network of the provided adjacency matrix A (whose
%entries are the node's common keys increased by 1) and node labels L.
fig=figure;
fig.Color='white';
axes('position',[0 0 1 1]);
P=plot(graph(A));
P.NodeColor='black';
P.EdgeColor='black';
P.NodeLabel=L;
W=graph(A).Edges.Weight;
for i=1:size(W,1)
    labeledge(P,i,W(i)-1);
end
axis off square tight;
%Uncomment the following lines to export the figure as a 500x500 png file.
% fig.Position(3)=500;
% fig.Position(4)=500;
% print('Network','-dpng','-r600')
end
```

It is crucial to note that all entries in the input adjacency matrix have to be increased by one, since Matlab would otherwise not plot an edge between nodes who have 0 as a common key.

A.1 Zig-zag products

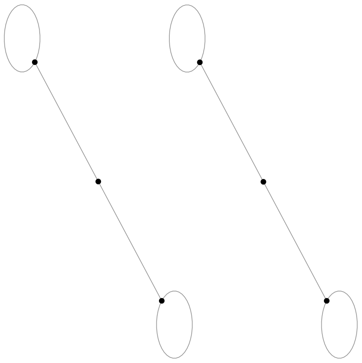
All notation is inherited from Section 4.2; except the use of an apostrophe is replaced by enumeration, due to the specific meaning of that symbol in Matlab. We will illustrate the usage of every function with the multigraphs and labelings from Example 8. Every adjacency matrix is also accompanied by a graph plot.

```
>> A=[2 1;1 2];
B=[1 0 1;0 1 1;1 1 0];
L1=[1 2 3;3 1 2];
L2=[1 2 3;2 1 3];
```

The matrices that represent the labeling should be constructed as follows: the ith row contains the labels of the edges incident with vertex v_i; the labels are added according to the ordering of the vertex set, and a random ordering should be fixed on all multiple edges between two vertices.

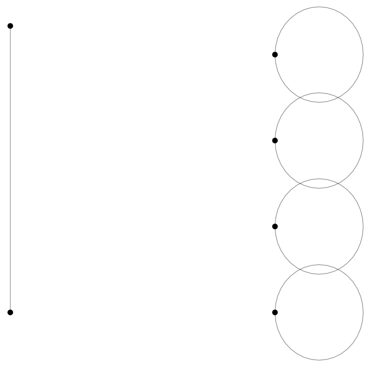
```
function [ Z ] = Zmatrix( A,B )
%ZMATRIX Determine Z based on the adjacency matrices of G and H.
[n,~]=size(A);
Z=kron(eye(n),B);
```

```
>> Z = Zmatrix(A,B)

1      0      1      0      0      0
0      1      1      0      0      0
1      1      0      0      0      0
0      0      0      1      0      1
0      0      0      0      1      1
0      0      0      1      1      0
```

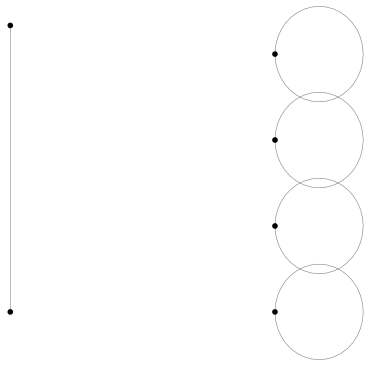


```
function [ N ] = Nmatrix( A,L )
%NMATRIX Determine N based on the adjacency matrix of G and the labeling.
r=sum(A(1,:));
[n,~]=size(A);
N=zeros(n*r);
for v=1:n
    for w=v:n
        for k=1:A(v,w)
            labv=L(v,k+sum(A(v,1:w-1)));
            labw=L(w,k+sum(A(w,1:v-1)));
            i=(v-1)*r+labv;
            j=(w-1)*r+labw;
            N(i,j)=1;
        end
    end
end
N=N+triu(N,1)';
```

```
>> N1 = Nmatrix(A,L1)

1     0     0     0     0     0
0     1     0     0     0     0
0     0     0     0     0     1
0     0     0     1     0     0
0     0     0     0     1     0
0     0     1     0     0     0
```

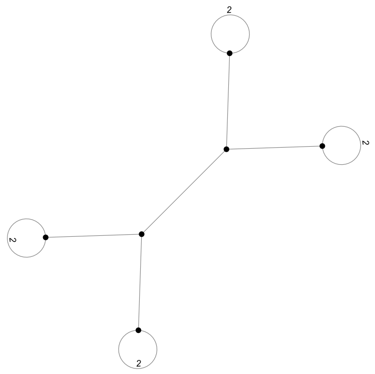


```
>> N2 = Nmatrix(A,L2)

1     0     0     0     0     0
0     1     0     0     0     0
0     0     0     0     1     0
0     0     0     1     0     0
0     0     1     0     0     0
0     0     0     0     0     1
```

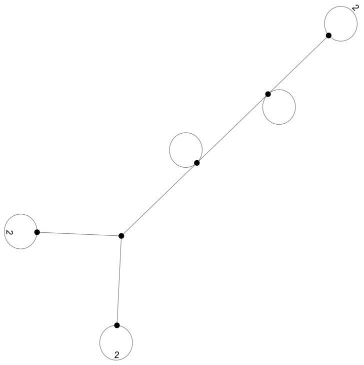


```
function [ R ] = Replace( A,B,L )
%REPLACE Determine the replacement product of G and H with labeling L.
Z=Zmatrix(A,B);
N=Nmatrix(A,L);
R=Z+N;
end
```

```
>> R1 = Replace(A,B,L1)

2     0     1     0     0     0
0     2     1     0     0     0
1     1     0     0     0     1
0     0     0     2     0     1
0     0     0     0     2     1
0     0     1     1     1     0
```

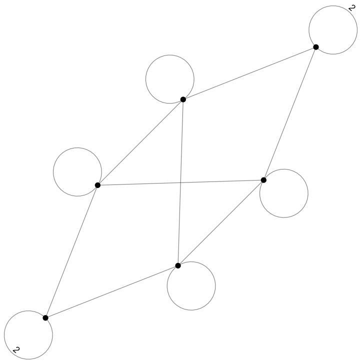


```
>> R2 = Replace(A,B,L2)

2      0      1      0      0      0
0      2      1      0      0      0
1      1      0      0      1      0
0      0      0      2      0      1
0      0      1      0      1      1
0      0      0      1      1      1
```

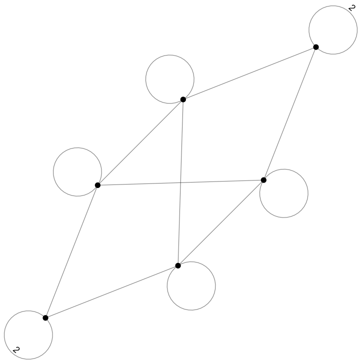


```
function [ M ] = ZigZag( A,B,L )
%ZIGZAG Determine the zig-zag product of G and H with the labeling L.
Z=Zmatrix(A,B);
N=Nmatrix(A,L);
M=Z*N*Z;
end
```

```
>> M1 = ZigZag(A,B,L1)

1      0      1      1      1      0
0      1      1      1      1      0
1      1      2      0      0      0
1      1      0      1      0      1
1      1      0      0      1      1
0      0      0      1      1      2
```



```
>> M2 = ZigZag(A,B,L2)

1      0      1      0      1      1
0      1      1      0      1      1
1      1      2      0      0      0
0      0      0      2      1      1
1      1      0      1      1      0
1      1      0      1      0      1
```



All previously displayed graph plots are made with the function `WeightedGraph`.

A.2 Blom key predistribution scheme

We inherit all notation from Section 5.1 and demonstrate how Example 11 was obtained.

```
function [ A ] = BlomSimple( n )
%BLOMSIMPLE Plot a network of size n using the Blom KPS with S=1.
p=n;
while isprime(p)==0
    p=p+1;
end
r=mod(randperm(p,n),p);
A=randi([0 p-1],1,3);
syms x y;
f(x,y)=A(1)+A(2)*(x+y)+A(3)*x*y;
K=zeros(n);
for i=1:n
    for j=i+1:n
        K(i,j)=mod(f(r(i),r(j)),p)+1;
    end
end
PlotNetwork(K+K',r);
end
```

For simplicity, the function uses for p the smallest prime such that $n \leq p$, but the code can be simplified if p is provided by the user. The parameters a, b and c are outputted for reference.

```
>> BlomSimple(10)

4      1      6
```

The output graph is shown in Fig. A.1. Every node N is labeled with r_N, so it is immediately clear where N_i ($r_{N_i} = 3$), N_j ($r_{N_j} = 6$) and N_k ($r_{N_k} = 8$) are located. We see that the edge labels correspond to our manually calculated values for the common keys, namely $K_{ij} = 0$, $K_{ik} = 5$ and $K_{jk} = 9$.

The implementation of the general Blom KPS requires a small trick, because Matlab does not allow symbolic indices for the matrix entries a_{ij} in the definition of the polynomial f. We can solve this problem by noting that $f(x, y)$ can be written as

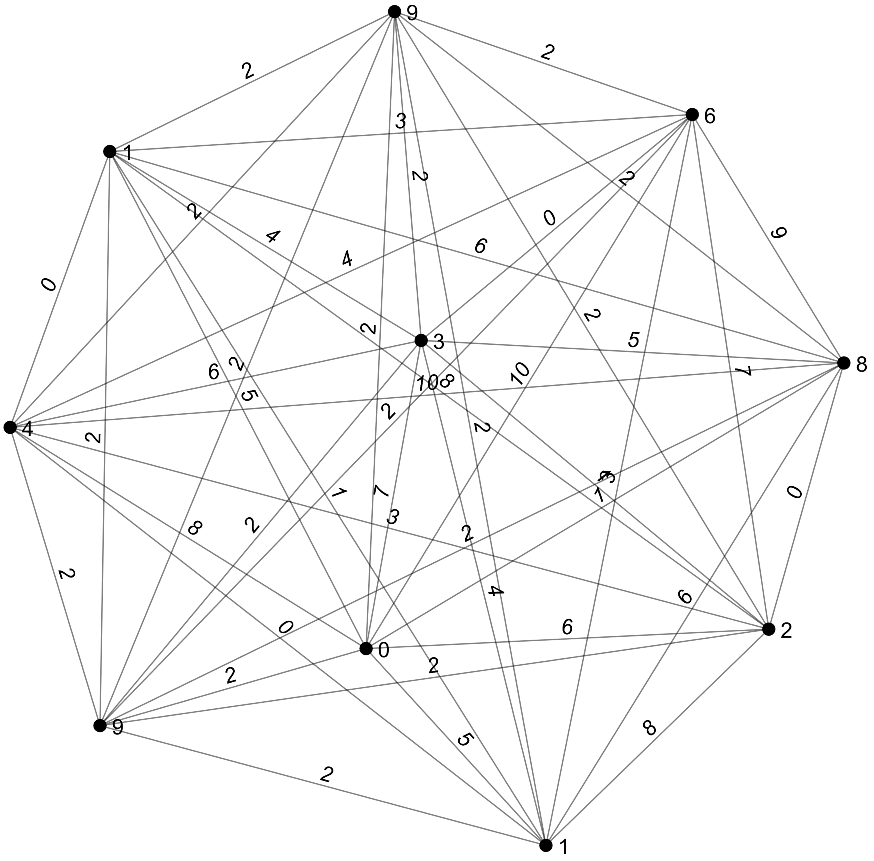


Figure A.1: A network using the Blom KPS with $S = 1$.

the sum of all entries in the square matrix

$$\begin{pmatrix} a_{00} & a_{01} & \cdots & a_{0S} \\ a_{10} & a_{11} & \cdots & a_{1S} \\ \vdots & \vdots & \ddots & \vdots \\ a_{S0} & a_{S1} & \cdots & a_{SS} \end{pmatrix} \star \begin{pmatrix} x^0 & x^0 & \cdots & x^0 \\ x^1 & x^1 & \cdots & x^1 \\ \vdots & \vdots & \ddots & \vdots \\ x^S & x^S & \cdots & x^S \end{pmatrix} \star \begin{pmatrix} y^0 & y^1 & \cdots & y^S \\ y^0 & y^1 & \cdots & y^S \\ \vdots & \vdots & \ddots & \vdots \\ y^0 & y^1 & \cdots & y^S \end{pmatrix},$$

where $\star$ denotes the element-wise multiplication.

```
function [ a ] = Blom( n,S )
%BLOM Plot a network of size n using the Blom KPS.
p=n;
while isprime(p)==0
    p=p+1;
end
r=mod(randperm(p,n),p);
a=randi([0 p-1],S+1);
```

```
a=triu(a)+triu(a,1)';
k=repmat((0:S)',1,S+1);
l=repmat((0:S),S+1,1);
syms x y;
f(x,y)=sum(sum(a.*x.^k.*y.^l));
K=zeros(n);
for i=1:n
    for j=i+1:n
        K(i,j)=mod(f(r(i),r(j)),p)+1;
    end
end
PlotNetwork(K+K',r);
end
```

```
>> Blom(10,3)

 6    10     6    10
10     7     9     1
 6     9     8     7
10     1     7     5
```

The coefficients a_{ij} are displayed as a matrix for reference. The resulting network is shown in Fig. A.2.

```
function [ P ] = Interpolation( X,Y,p )
%INTERPOLATION Apply Lagrange's polynomial interpolation in Fp for the
%values in X and respective polynomials whose coefficients are given
%row-wise in Y (from low to high degree).
syms x y;
n=length(X);
P=0;
for i=1:n
    Q=1;
    for j=1:n
        if j~=i
            [~,q,~]=gcd(X(i)-X(j),p);
            Q=Q*(y-X(j))*q;
        end
    end
    P=P+sum(Y(i,:).*x.^(0:n-1))*Q;
end
P=expand(mod(P,p));
end
```

```
>> X=[0 2 4 6]; Y=[6 10 6 10; 9 2 2 3; 1 4 2 6; 0 9 1 6];
>> Interpolation(X,Y,11)

5*x^3*y^3 + 7*x^3*y^2 + x^3*y + 10*x^3 + 7*x^2*y^3 + 8*x^2*y^2 + 9*x^2*y
```

```
+ 6*x^2 + x*y^3 + 9*x*y^2 + 7*x*y + 10*x + 10*y^3 + 6*y^2 + 10*y + 6
```

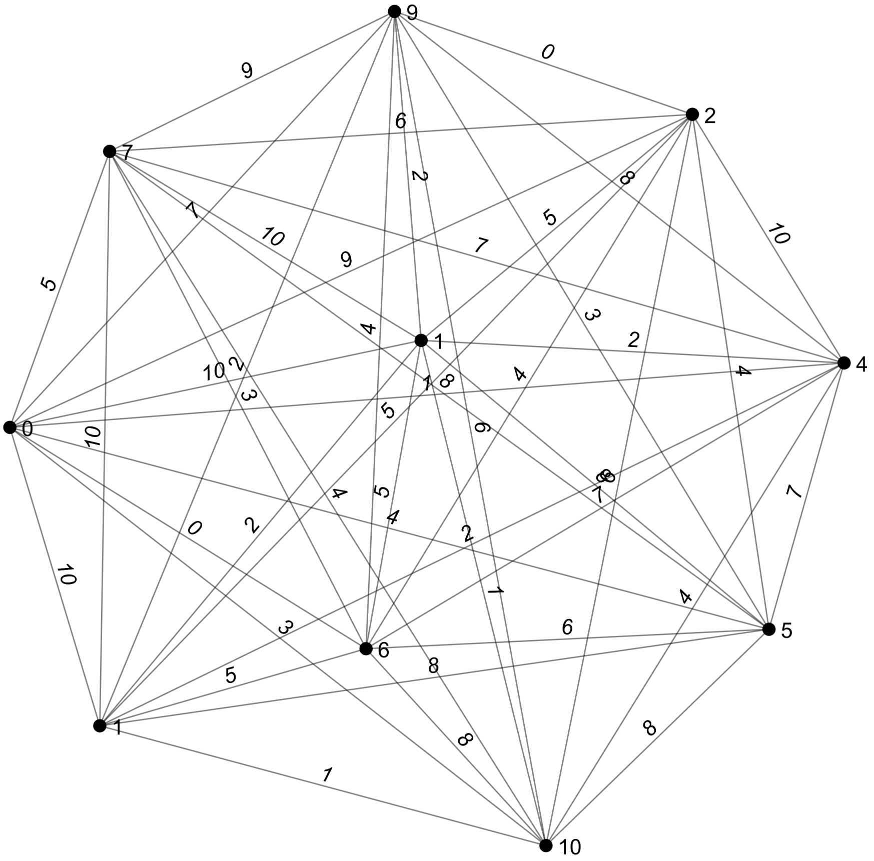


Figure A.2: A network using the Blom KPS with $S = 3$.

A.3 Eschenauer–Gligor key predistribution scheme

```
function [ S ] = EschenauerGligor( n,k,m )
%ESCHENAUERGLIGOR Plot a network of size n using the Eschenauer-Gligor KPS
%with a key pool of size m.
S=zeros(n,k);
for i=1:n
    S(i,:)=randperm(m,k);
end
M=zeros(n);
for i=1:n
    for j=i+1:n
        I=intersect(S(i,:),S(j,:));
        if isempty(I)==0
            r=randi(length(I));
            M(i,j)=I(r)+1;
        end
    end
end
PlotNetwork(M+triu(M)',{});
end
```

In order to illustrate the built-in randomness of the Eschenauer–Gligor KPS, we apply it on four networks with the same parameters $n = 10$, $k = 5$ and $|K| = 100$, using the notation from Section 5.3. We see in Fig. A.3 that the resulting communication networks may become disconnected.

```
>> EschenauerGligor( 10,5,100 )
```

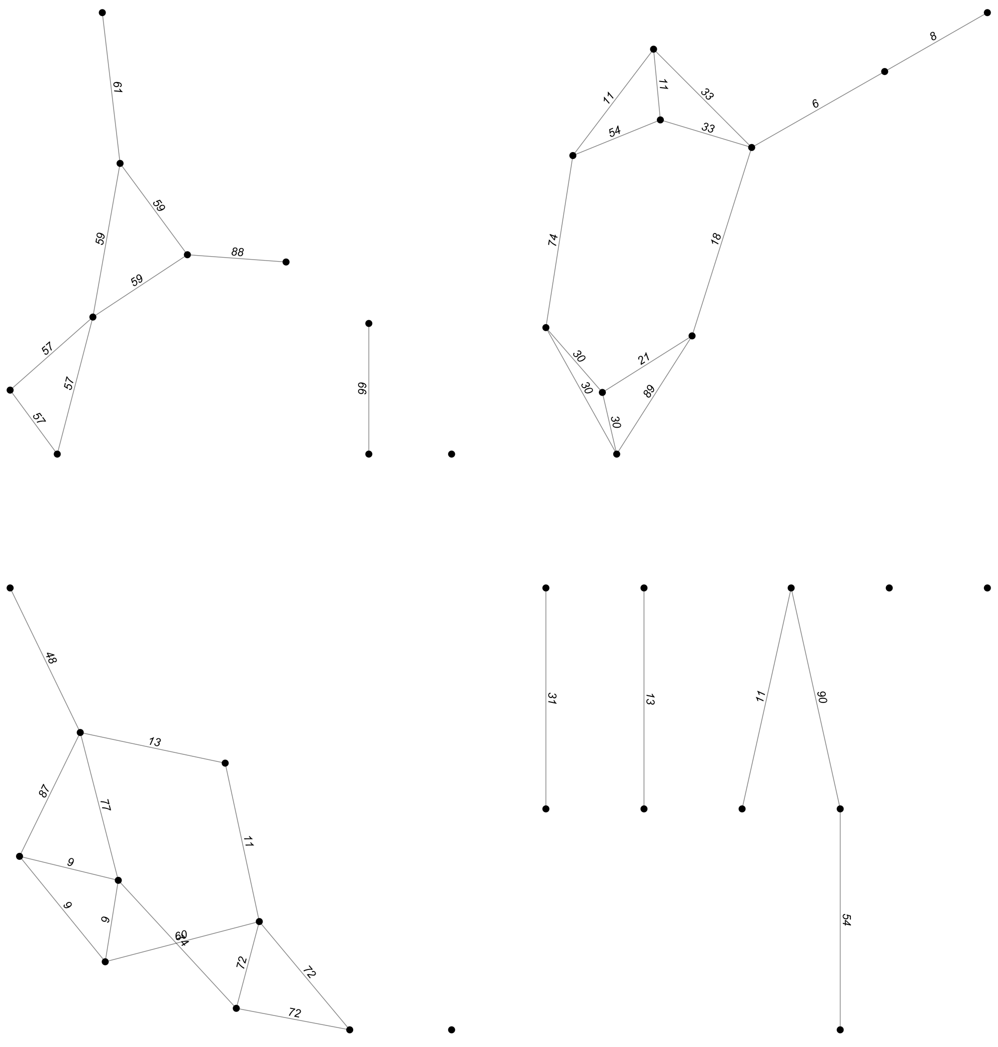


Figure A.3: Four networks using the Eschenauer–Gligor KPS.

Bibliography

[1] N. Alon and J.H. Spencer. *The Probabilistic Method.* Discrete Mathematics and Optimization. John Wiley & Sons, Hoboken, New Jersey, fourth edition, 2016.

[2] T.M. Apostol. *Introduction to Analytic Number Theory.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1976.

[3] 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, *3rd International Conference on Information Theoretic Security*, volume 5155 of *Lecture Notes in Computer Science*, pages 54–69, Calgary, August 2008. Springer-Verlag, Berlin, Heidelberg.

[4] R. Blom. An optimal class of symmetric key generation systems. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology. EUROCRYPT 1984*, volume 209 of *Lecture Notes in Computer Science*, pages 335–338. Springer-Verlag, Berlin, Heidelberg, 1985. Springer Link.

[5] C. Blundo, A. De Santis, and A. Herzberg et al. Perfectly-secure key distribution for dynamic conferences. In *Advances in Cryptology, CRYPTO' 92*, volume 740 of *Lecture Notes in Computer Science*, pages 471–486. Springer-Verlag, August 1992. Springer Link.

[6] S.A. Camtepe, B. Yener, and M. Yung. Expander graph based key distribution mechanisms in wireless sensor networks. In IEEE, editor, *IEEE International Conference on Communications*, IEEE International Conference on Communications, pages 2262–2267. IEEE, 2006. IEEE Link.

[7] P. Cara. Discrete wiskunde. Course notes, Vrije Universiteit Brussel, 2012.

[8] P. Cara. Lineaire algebra, volumes I and II. Course notes, Vrije Universiteit Brussel, 2012.

[9] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 197–214. IEEE Computer Society, Washington DC, May 2003.

[10] W. Dargie and C. Poellabauer. *Fundamentals of Wireless Sensor Networks. Theory and Practice.* Wireless Communications and Mobile Computing. John Wiley & Sons, West Sussex, 2010.

[11] G. Davidoff, P. Sarnak, and A. Valette. *Elementary Number Theory, Group Theory, and Ramanujan Graphs.* Cambridge University Press, Cambridge, 2003.

[12] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[13] W. Du, J. Deng, Y.S. Han, and P.K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 42–51, Washington D.C., October 2003.

[14] P. Erdős and A. Rényi. On the evolution of random graphs. *Bulletin of the International Statistical Institute*, 38(4):343–347, 1960.

[15] L. Eschenauer and V.D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 18–22, Washington DC, November 2002.

[16] C. Godsil and G. Royle. *Algebraic Graph Theory*. Graduate Texts in Mathematics. Springer-Verlag, 2001.

[17] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, October 2006.

[18] E. Kendall, M. Kendall, and W.S. Kendall. A generalised formula for calculating the resilience of random key predistribution schemes. Cryptology ePrint Archive, Report 2012/426, 2012. http://eprint.iacr.org/2012/426.

[19] M. Kendall and K.M. Martin. On the role of expander graphs in key predistribution schemes for wireless sensor networks. In F. Armknecht and S. Lucks, editors, *WEWoRC 2011. Research in Cryptology*, volume 7242 of *Lecture Notes in Computer Science*, pages 62–82. Springer-Verlag, Berlin, Heidelberg, 2012. Springer Link.

[20] M. Kendall and K.M. Martin. Graph-theoretic design and analysis of key predistribution schemes. *Designs, Codes and Cryptography*, 81(1):11–34, October 2016. Springer Link.

[21] N. Koblitz. *A Course in Number Theory and Cryptography*. Graduate Texts in Mathematics. Springer–Verlag, New York, second edition, 1994.

[22] M. Krebs and A. Shaheen. *Expander Families and Cayley Graphs: A Beginner's Guide*. Oxford University Press, Oxford, 2011.

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

[24] J. Lee and D.R. Stinson. Common intersection designs. 14(4):251–269, 2006.

[25] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[26] G.A. Margulis. Explicit constructions of concentrators. *Problems of Information Transmission*, 9(4):325–332, October–December 1973.

[27] K.M. Martin. On the applicability of combinatorial designs to key predistribution for wireless sensor networks. In Y.M. Chee, C. Li C., S. Ling, H. Wang, and C. Xing, editors, *Coding and Cryptology*, volume 5557 of *Lecture Notes in Computer Science*, pages 124–145. Springer, Berlin, Heidelberg, 2009. Springer Link.

[28] K.M. Martin. The rise and fall and rise of combinatorial key predistribution. In A. Biryukov, G. Gong, and D.R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 92–98. Springer, Berlin, Heidelberg, 2011. Springer Link.

[29] F. Martincic and L. Schwiebert. Introduction to wireless sensor networking. In I. Stojmenovic, editor, *Handbook of Sensor Networks. Algorithms and Architectures*, Parallel and Distributed Computing, chapter 1, pages 1–40. John Wiley & Sons, Hoboken, New Jersey, 2005.

[30] C.J. Mitchell. Key storage in secure networks. *Discrete Applied Mathematics*, 21(3):215–228, October 1988.

[31] T. Newe, V. Cionca, and D. Boyle. Security for wireless sensor networks, configuration aid. In S.C. Mukhopadhyay and H. Leung, editors, *Advances in Wireless Sensors and Sensor Networks*, volume 64 of *Lecture Notes in Electrical Engineering*, pages 1–24. Springer-Verlag, Berlin, Heidelberg, 2010. Springer Link.

[32] A. Nilli. On the second eigenvalue of a graph. *Discrete Mathematics*, 91:207–210, August 1991.

[33] H.K. Patil and S.A. Szygenda. *Security for Wireless Sensor Networks using Identity-Based Cryptography*. CRC Press, Boca Raton, 2013.

[34] M. Pinsker. On the complexity of a concentrator. In *7th International Telegraffic Conference*, pages 318/1–318/4, Stockholm, June 1973.

[35] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, January 2002.

[36] E. Sabbah and K. Kang. Security in wireless sensor networks. In S. Misra, I. Woungang, and S.C. Misra, editors, *Guide to Wireless Sensor Networks*, Computer Communications 491 and Networks, chapter 19, pages 491–512. Springer-Verlag, London, 2009.

[37] R.R. Selmic, V.V. Phoha, and A. Serwadda. *Wireless Sensor Networks. Security, Coverage, and Localization*. Springer, 2016.

[38] H. Shafiei, A. Mehdizadeh, A. Khonsari, and M. Ould-Khaoua. A combinatorial approach for key-distribution in wireless sensor networks. In IEEE, editor, *IEEE GLOBECOM 2008*, IEEE Global Telecommunications Conference. IEEE, 2008. IEEE Link.

[39] D.R. Stinson. *Cryptography, Theory and Practice*. Discrete Mathematics and its Applications. Chapman and Hall, Boca Raton, third edition, 2006.

[40] L. Trevisan. Pcp and hardness of approximation, notes for lecture 11. Course notes, U.C. Berkeley, 2006.

[41] Y. Zhou, Y.G. Fang, and Y.C. Zhang. Securing wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 10(3):6–28, 2008.

Index

(n, m, r, k)-configuration, 54
(n, r, μ, ν)-strongly regular graph, 57
(t, S)-cover-free family, 48
S-unconditionally secure, 40
μ-common intersection design, 56
k-uniform set system, 9
q-composite KPS, 52
r-regular, 7, 9
r-uniform hypergraph, 8
t-(n, k, λ) design, 9

adjacency matrix, 14
adjacency operator, 15
adjacent vertex, 6

bipartite graph, 7
block, 9
Blom KPS, 41, 43
bounded away from zero, 12

Cayley graph, 8
cloud, 29
combinatorial design, 9
communication graph, 7
complement, 6
complete graph, 6
complete pairwise KPS, 5
connected graph, 6
constant on clouds, 33
cycle, 6
cycle graph, 12

degree, 7, 9
diameter, 7
distance, 7

edge, 6
eigenvalue of a graph, 14
endpoint, 6
Eschenauer–Gligor KPS, 50
expander family, 12
expander graph, 12
expansion coefficient, 11

feasibility condition, 57
Fiat–Naor S-KDP, 47

graph, 6
group key, 46

hopping, 4
hyperedge, 8
hypergraph, 8

incidence matrix, 10
incident, 6
intersection graph, 7
isomorphic graph, 6

key distribution, 1
key distribution pattern, 46
key graph, 7
key predistribution, 1
key predistribution scheme, 1
KPS, 1

labeling, 29

labeling at v, 29
Laplacian operator, 18
length, 6
loop, 8

Mitchell–Piper (t, S)-KDP, 48
multigraph, 8
multiplicity, 8
multiset, 8

node, 1
node compromise, 5

orientation, 18

path, 6
perfect resilience, 5
point, 9
power of a multigraph, 38

probabilistic method, 48

Ramanujan graph, 27
replacement product, 29

security parameter, 40
set system, 9
spectral gap, 17
square of a multigraph, 38
symmetric multiset, 8

trivial eigenvalue, 27

vertex, 6

walk, 8
wireless sensor network, 3
WSN, 3

zig-zag product, 31