ALBANIAN JOURNAL
OF MATHEMATICS
Volume 2, Number 3, Pages 235–239
ISSN 1930-1235: (2008)

# SIERPIŃSKI GASKET-BASED GRAPHS IN CODING THEORY

MONIKA KOTOROWICZ

## 1. Introduction

In this paper we build a family of hierarchical graphs based on the triangle (Sierpiński gasket-based graphs) and calculate their important characteristics, such as average degree, average shortest path length, small-world graph family characteristics. Then we present stream ciphers defined on a finite automaton corresponding to this family.

## 2. Basic network characteristics

Our family of graphs $\{\Lambda_k\}_{k\in\mathbb{N}}$ is generated in an hierarchical way (see [1]). Here $k = 1, 2, 3, \ldots$ denotes the level of the hierarchy understood as the step of the construction. The initial graph $\Lambda_1$ is the complete graph of order 3. At each step of the construction we join 3 graphs of level $k-1$ (called units) in a way shown in Figure 1.

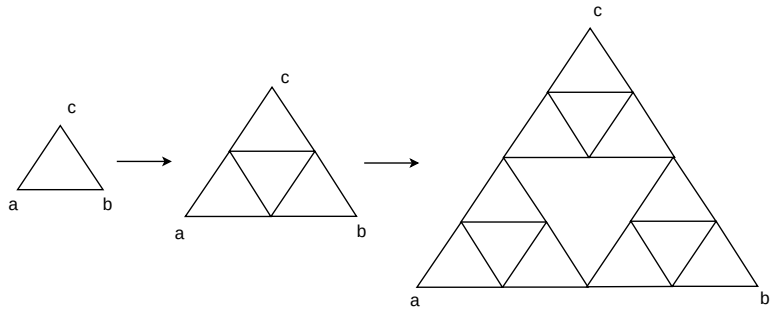


Figure 1. Construction of the graph $\Lambda_3$

Each graph has 3 external vertices of special meaning. Units of the same level are attached to them to form the unit of a higher level. In the figures we denote them by $a, b, c$. The rest of vertices are called internal.

The result is a family of Sierpiński gasket-based graphs $\{\Lambda_k\}_{k\in\mathbb{N}}$ with no loops and no multiple edges. By $V_k$ and $E_k$ we denote the sets of vertices and edges of $\Lambda_k$, respectively. One can find the *order* $|V_k|$ and the *size* $|E_k|$ of $\Lambda_k$:

$$|V_k| = \frac{3}{2}(3^{k-1}+1), \quad |E_k| = 3^k.$$

2.1. **Average degree.** Let $n_k(v)$ stand for the number of edges ending at a vertex $v \in V_k$. Clearly, $n_k(v) = 2$ for each external vertex and $n_k(v) = 4$ for each internal one. So

$$\langle n_k \rangle \stackrel{\text{def}}{=} \frac{1}{|V_k|} \sum_{v \in V_k} n_k(v) = \frac{3 \cdot 3 + 4 \cdot (|V_k| - 3)}{|V_k|}$$

which tends to 4 when $k \to \infty$.

2.2. **Average shortest-path length.** As Figure 2 suggests, it is convenient to introduce the following notations. The graph of level $k$, $k > 1$, consists of 3 subgraphs of level $k-1$

$$\Lambda_k = \Lambda^a_{k-1} \cup \Lambda^b_{k-1} \cup \Lambda^c_{k-1}.$$

Every vertex $v \in V_k$ has a label determining its place in the graph

$$v = \{\alpha_1 \alpha_2 \dots \alpha_k\}, \alpha_i \in \{a, b, c\}.$$

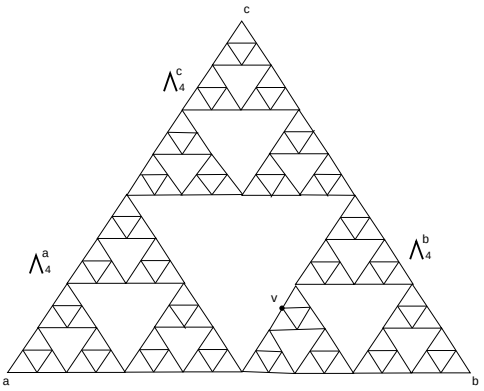


FIGURE 2. The graph of level 5

Each symbol corresponds to the choice of the triangle of the previous level. Notice that every vertex, besides the external ones, has two labels. For example the vertex $v$ in Figure 2 can be labelled by $\{bacca\}$ or $\{bacac\}$. The distance $\rho_k(v, \gamma)$ between $v$ and $\gamma \in \{a_k, b_k, c_k\}$, measured in terms of the number of edges along the path in $\Lambda_k$, is

$$\rho_k(v, \gamma) = (1 - \delta_{\alpha_k \gamma}) + \sum_{j=1}^{k-1} 2^{j-1}(1 - \delta_{\alpha_{k-j} \gamma}), \quad \delta_{\alpha_i \gamma} = \begin{cases} 1 & \alpha_i = \gamma, \\ 0 & \alpha_i \neq \gamma. \end{cases}$$

Let $v \in \Lambda^a_{k-1}$ and $w \in \Lambda^b_{k-1}$. Then the distance between $v$ and $w$ is

$$\rho_k(v, w) \leq \rho_{k-1}(v, b) + \rho_{k-1}(w, a).$$

Thus, the average shortest-path length $\rho_k$ is

$$\rho_k = \frac{\sum_{v,w \in V_k} \rho_k(v, w)}{\frac{1}{2}|V_k|(|V_k| - 1)} = \Theta(2^k).$$

2.3. **Small world graph family.** In the last years, small-world networks have been studied intensively, see [2, 3]. The family of graphs $\{\Lambda_k\}$ is a small world graph family if the diameter of $\Lambda_k$ (i. e. the maximal distance between the two vertices in $\Lambda_k$) scales logarithmically or slower with the graph size, that is,

$$\exists C > 0 \quad \mathrm{diam}\Lambda_k \leq C \log_{\langle n_k \rangle} |V_k|.$$

In our model, one has $\mathrm{diam}\Lambda_k = 2^{k-1}$ so it is not the small world graph family. We need this important information to our application in cryptography.

## 3. Cryptographical application on Sierpiński gasket-based graphs

To adapt our model described in previous section to our cryptographical application ([4, 5]) we need to make some changes. For every pair of distinct vertices connected with a simple edge we replace this edge with a pair of directed edges with opposite directions. Moreover, for every pair of distinct external vertices we add two directed edges with opposite directions (Figure 3). So every vertex $v \in V_k$ has the same number of input and output edges (equal to 4). Our graph is 4-regular.

From now on, $\Lambda_k$ and other notation stand for the changed model. Notice that the order of $\Lambda_k$ has not changed but the size of $\Lambda_k$ has ($|E_k| = 2(3^k + 3)$). Of course, the average shortest-path length and the diameter of $\Lambda_k$ has changed too. But they are still the powers of 2. So new $\{\Lambda_k\}$ is not a small world graphs family.

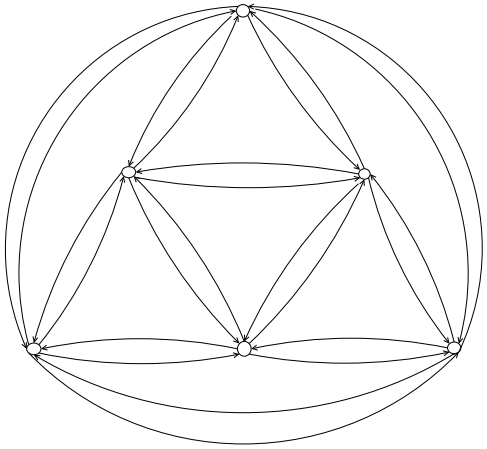

Figure 3. The graph of level 2

In this paper we use the conventional cryptographical notation. The ordinary information (called plaintext) will be transformed into an encrypted, unintelligible information (a ciphertext) by the cryptographical algorithm with a password (a key).

The vertices and the edges represent the states and the transition between these states in an automaton, respectively.

3.1. **Encryption scheme.** Let $\mathbb{F}_3^k$ be a vector space over a finite field $\mathbb{F}_3 = \{a, b, c\}$. Every vector $v = [\alpha_1\alpha_2 \ldots \alpha_k], \alpha_i \in \mathbb{F}_3$ is considered to be a vertex label in $\Lambda_k$. We identify every label $v \in \mathbb{F}_3^k$ with a plaintext or a ciphertext of length $k$. Notice that

every label points to exactly one vertex but every vertex (besides external ones) has two labels.

An encryption scheme on our graph model relies on special colouring of edges. We need to attribute a colour to each edge $e \in E_k$ in such a way that no two adjacent edges of the same direction (starting at the same vertex or ending in it) share the same colour.

**Lemma 1.** *Let $\{\Lambda_k\}_{k\in\mathbb{N}}$ be a graphs family presented above. For every $k = 1, 2, \ldots$ there exists a 4-colouring of edges such that for every vertex $v \in V_k$ any pair of edges starting (or ending) at $v$ has not the same colour. And there is a representative of each colour in the set of edges starting (or ending) at each vertex $v$.*

For example Figure 4 presents edges colouring in $\Lambda_2$. We identify a set of colours with elements of $\mathbb{Z}_4 = \{0, 1, 2, 3\}$. A path in $\Lambda_k$ between $v, w \in V_k$ is represented by a finite sequence of colours $[c_0, c_1, \ldots, c_m], c_i \in \mathbb{Z}_4$.

Let $c$ be a colour of edge from a vertex $v$ to a vertex $w$. By $c^{-1}$ we denote the colour of edge from $w$ to $v$.

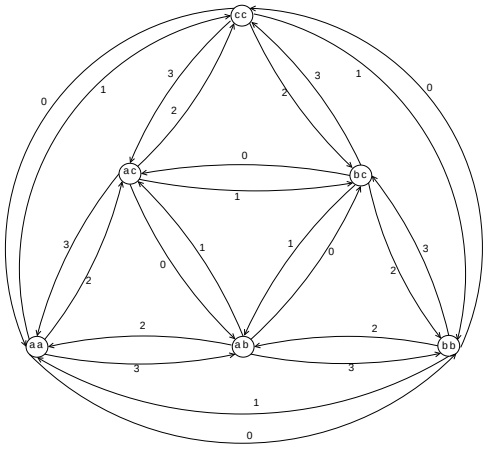


FIGURE 4. The vertex labeling and edges colouring in $\Lambda_2$

Let $p$ be a plaintext we encrypt to a ciphertext $c$. Let $v_p$ and $v_c$ be a vertices representing the plaintext and the ciphertext, respectively. The key $k = [c_0, c_1, \ldots, c_m, \eta], c_i \in \mathbb{Z}_4$, $i = 0, 1, \ldots, m$, $c_{i+1} \neq c_i^{-1}$, $\eta \in \{0, 1\}$ in an encryption procedure consisting of two parts. The first one $[c_0, c_1, \ldots, c_m]$ is the path between $v_p$ and $v_c$. The second part $\eta$ defines which one of two possible labels at $v_p$ concerns a plaintext. The space $\mathbb{F}_3^k$ is totally ordered, so we put $\eta = 0$ for the first label and $\eta = 1$ for the second one. This information is necessary to a decryption procedure.

Notice that $v_p$ is a starting state in an automaton. Every password leads us to some $v_c$ and all states (vertices) of such automaton are accepting ones.

The encryption scheme is to start in a vertex $v_p$ and pass on the graph along the path defined by a password $k$ (Figure 5). In each step of the algorithm the transition

function $f : V_k \times \mathbb{Z}_4 \to V_k$ appoints a next vertex according to the following scheme

$$\begin{array}{rcl} f(v_p, c_0) & = & v_1 \\ f(v_1, c_1) & = & v_2 \\ & \dots & \\ f(v_m, c_m) & = & v_c. \end{array} \tag{1}$$

At the end of this procedure we will reach the vertex $v_c$.

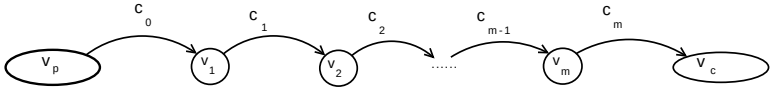


FIGURE 5. The encryption algorithm

As was mentioned above, the family $\{\Lambda_k\}_{k\in\mathbb{N}}$ is not a small world graphs family. It means that short paths between every pair of distinct vertices cannot exist. So if we choose a sufficiency long password we will be sure that ciphertext cannot be decrypted by other passwords of small length.

Each graph $\Lambda_k$ is connected and the average shortest-path length is a power of 2. Hence our algorithm is a stream cipher. Each step depends only on the state of the system after the previous step.

3.2. **Decryption procedure.** A decryption procedure bases on the inverse function $f^{-1} : V_k \times \mathbb{Z}_4 \to V_k$. The function $f^{-1}$ for $v_i \in V_k$ and $c \in \mathbb{Z}_4$ returns a vertex $v_j \in V_k$ such that $f(v_j, c) = v_i$ (one of predecessors of $v_i$ indicated by an edge of colour $c$). Knowing the ciphertext $v_c$ and the key $k = [c_0, c_1, \dots, c_m, \eta], c_i \in \mathbb{Z}_4, \eta \in \{0, 1\}$ one can obtain the vertex $v_p$ in the following decryption scheme

$$\begin{array}{rcl} f^{-1}(v_c, c_m) & = & v_m \\ f^{-1}(v_m, c_{m-1}) & = & v_{m-1} \\ & \dots & \\ f^{-1}(v_1, c_0) & = & v_p. \end{array} \tag{2}$$

Every internal vertex $v_p$ has two labels, so we have to choose the proper one using the information $\eta$ of the key $k$.

## REFERENCES

[1] Clauset A., Moore C., Newman M., *Structural Inference of Hierarchies in Networks*, In: Proceedings of the 23rd International Conference on Machine Learning, Workshop on "Statistical Network Analysis", Springer Lecture Notes in Computer Science (Pittsburgh, June 2006), also arXiv:physics/0610051v1 [physics.soc-ph] 9 Oct 2006

[2] Newman M.E.J., SIAM Review, 2003, **45**, 167 – 256

[3] Barrat A., Weigh M., Eur. Phys. J. B, 2000, **13**, 547–560

[4] V. Ustimenko, *CRYPTIM: Graphs as Tools for Symmetric Encryption*, in Lecture Notes in Computer Science, Springer, v. 2227, 278-287.

[5] V. Ustimenko, *On the extremal graph theory for directed graphs and its cryptographical applications* In: T. Shaska, W.C. Huffman, D. Joener and V. Ustimenko, Advances in Coding Theory and Cryptography, Series on Coding and Cryptology, vol. 3, 181-200 (2007).