# ECOLE NORMALE SUPERIEURE

**Black Box Cryptanalysis of Cryptographic Primitives**

Claus Peter SCHNORR
Serge VAUDENAY

LIENS - 95 - 28

Département de Mathématiques et Informatique

CNRS URA 1327

# Black Box Cryptanalysis of Cryptographic Primitives

**Claus Peter SCHNORR***
**Serge VAUDENAY**

**LIENS - 95 - 28**

November 1995

Laboratoire d'Informatique de l'Ecole Normale Supérieure
45 rue d'Ulm 75230 PARIS Cedex 05

Tel : (33)(1) 44 32 00 00
Adresse électronique : vaudenay@dmi.ens.fr

*Universität Frankfurt - FB Math./Inf.
Postfach 111932 60054 Frankfurt a.M. Germany

Adresse électronique : schnorr@informatik.uni-frankfurt.de

# Black Box Cryptanalysis of Cryptographic Primitives

Claus Peter Schnorr[1] Serge Vaudenay[2]

[1] Universität Frankfurt — FB Math./Inf.
Postfach 111932
60054 Frankfurt a.M. — Germany
Schnorr@informatik.uni-frankfurt.de

[2] Ecole Normale Supérieure — DMI***
45, rue d'Ulm
75230 Paris cedex 5 — France
Serge.Vaudenay@ens.fr

**Abstract**

We analyse the security of a cryptographic primitive on the basis of the geometry of its computation graph. We assume the computation graph of the primitive to be given whereas the boxes sitting on the vertices of this graph are unknown and random, i.e. they are black boxes. We formalize and study a family of attacks which generalize exhaustive search and the birthday paradox. We establish complexity lower bounds for this family and we apply it to compression functions based on the FFT network.

Cryptographic primitives for encryption, hashing and pseudo-random generation are judged according to efficiency and security. It is a major problem to prove security in a realistic model, in particular since reasonable complexity lower bounds seem to be impossible to prove for the model of boolean networks. By black box cryptanalysis we can prove interesting complexity lower bounds for a family of attacks that comprises exhaustive search and the birthday paradox. Black box cryptanalysis complements differential and linear cryptanalysis [5, 7] which analyse particular boxes from the point of view of their non-linearity.

We assume the undirected *computation graph* $G = (V, E)$ to be given as well as an interpretation $I$ which associates to each vertex $v$ a random box $I(v)$ which is uniformly distributed and independent for distinct vertices $v$. Since all boxes are random, we cut out the role of particular boxes, and thus we cryptanalyse the geometry of the computation graph. A solution for the interpretation $I$ is an assignment of values, in some finite alphabet $Z$, to the edges in $E$ that is compatible with all the boxes $I(v)$. We study resolution algorithms that resolve the boxes $I(v)$ in some particular order of the vertices. Its complexity is the

***$\textit{Laboratoire d'Informatique de l'Ecole Normale Supérieure}$, research group affiliated with the CNRS maximal number of local solutions for the subsets of resolved boxes that appear during the process of resolution. The complexity of the graph $G$ is the minimal of these complexities over all orders of the vertices (actually we take logarithms to basis $\#Z$ of the cardinalities of these solution sets). We establish lower bounds on the complexity of computation graphs which in some cases are nearly sharp.

A particular case of black box cryptanalysis has been studied in [11]. This was done in context with the application to the hash functions based on the computation graph of the Fast Fourier Transform [10] known as the butterfly graph. Here we prove lower bounds in the general black box cryptanalysis context. A full formal study is also available in [15].

# 1 The black box cryptanalysis

## 1.1 Computation graphs with random boxes

Let $G = (V, E)$ be a finite, undirected, loop-free graph with vertex set $V$ and edge set $E$. For this paper all graphs are finite and loop-free. A computation along $G$ associates to each edge a value in some finite *alphabet* $Z$. The box associated to vertex $v$ defines the set of local solutions $I(v) \subset Z^{E(v)}$ for $E(v)$, the set of edges adjacent to $v$. Thus $I(v)$ is the set of assignments of values in $Z$ to the edges in $E(v)$ that are admissible for the box. For the degree $d(v) = \#E(v)$ of vertex $v$ we must have $\#I(v) \leq \#Z^{d(v)}$. However, if the box is non-trivial $\#I(v)$ must be smaller than $\#Z^{d(v)}$. E.g. if vertex $v$ has $i$ "input edges" then $\#I(v) = \#Z^i$ since the input values determine the output values. We associate with each vertex $v$ some fixed value for $\#I(v)$ and we call $\mathrm{df}(v) := \log_{\#Z} \#I(v)$ the *degree of freedom* of vertex $v$. In the following all logarithms have the basis $\#Z$ and log means $\log_{\#Z}$. For a subset $U \subset V$ we let $E(U)$ denote the set of edges adjacent to the vertices in $U$.

**Definition 1.** A *computation graph* $(G^{\mathrm{df}}, Z)$ consists of an undirected graph $G = (V, E)$, a real valued function $\mathrm{df}(v)$ satisfying $0 \leq \mathrm{df}(v) \leq d(v)$ and an alphabet $Z$. An *interpretation* $I$ is a map which associates with each vertex $v$ a set $I(v) \subset Z^{E(v)}$ of local solutions so that $\mathrm{df}(v) = \log \#I(v)$.

The graph $G^{\mathrm{df}}$ is undirected and so is the computation flow. A solution for the interpreted computation graph is a tuple $t \in Z^E$, i.e. a tuple on $E$, rather than a function mapping inputs to outputs. A tuple $t$ on $E$ is a *solution* for $G^{\mathrm{df}}$ if, for all vertices $v$, the restriction $t_{|E(v)}$ is in $I(v)$. To stress the undirected nature of $G^{\mathrm{df}}$ it was called *equation graph* rather than computation graph in [15].

We are not interested in particular boxes and interpretations, so we consider random interpretations $I$. In the following we assume that all probability distributions for $I$ have the following two properties:

**Local uniformity.** *For all $v \in V$ and $t \in Z^{E(v)}$ :* $\Pr[t \in I(v)] = \# Z^{df(v)-d(v)}$.

**Independence.** *The sets $I(v)$ are independent for distinct vertices $v$.*

Examples of possible distributions are

- the uniform distribution over all $I$ so that $I(v)$ is a subset of $Z^{E(v)}$ with $\#I(v) = \#Z^{\mathrm{df}(v)}$;
- the uniform distribution over all $I$ so that $I(v)$ defines a function of edge values of some $\mathrm{df}(v)$ edges in $E(v)$ to the other $d(v) - \mathrm{df}(v)$ edge values;
- the uniform distribution over all $I$ so that $I(v)$ defines a *multipermutation* on $Z^{E(v)}$ , see [14, 15].

With a computation graph $G^{\mathrm{df}}$ having vertex set $V$ we associate its *quadratic form* in $\mathbb{Z}[V]$ which we also denote $G^{\mathrm{df}}$ and which is defined by the symmetric matrix

$$G^{\mathrm{df}}_{v,w} = \begin{cases} -\frac{1}{2} & \text{if } v \neq w \text{ and } \{v,w\} \in E \\ \mathrm{df}(v) & \text{if } v = w \\ 0 & \text{otherwise.} \end{cases}$$

We identify a subset $U \subset V$ with the vector in $\{0,1\}^V$ that assigns to vertex $v$ the value 1 iff $v \in U$, and thus $G^{\mathrm{df}}(U) = \sum_{v,w \in U} G^{\mathrm{df}}_{v,w}$. Let $\mathrm{Int}(U) = \{\{v,w\} \in E \mid v,w \in U\}$ be the set of interior edges of $U$. It can easily be seen that

$$G^{\mathrm{df}}(U) \;=\; \sum_{v \in U} \mathrm{df}(v) - \#\mathrm{Int}(U) \;=\; \#E(U) + \sum_{v \in U} (\mathrm{df}(v) - d(v))$$

where the latter equality comes from $\sum\limits_{v \in U} d(v) = \#E(U) + \#\mathrm{Int}(U)$.

If $\mathrm{df}(v) = \frac{1}{2} d(v)$ holds for all vertices $v$ then for any subset $U$ of vertices $G^{\mathrm{df}}(U)$ equals to half the number of edges of the perimeter of $U$ (i.e. the edges between $U$ and $V - U$). We call this the *locally invertible* case as all the boxes look like permutations with the same number of inputs and outputs.

## 1.2 Resolution and complexity of a computation graph

For two subsets $E', E'' \subset E$ of edges and corresponding sets of tuples $X' \subset Z^{E'}$, $X'' \subset Z^{E''}$ we define the *join*

$$X' \bowtie X'' \;=\; \{t \in Z^{E' \cup E''} \mid t_{|E'} \in X',\, t_{|E''} \in X''\}$$

to be the set of all tuples $t$ on $E' \cup E''$ with restrictions to $E'$ in $X'$ and to $E''$ in $X''$. This operation $\bowtie$ is similar to the join used in the relational database theory. The join is associative and commutative and the set of *global* solutions on $E$ turns out to be the join of all $I(v)$.

We extend the interpretation $I$ to arbitrary subsets of vertices using that the join is associative and commutative. For $U \subset V$ let $I(U)$ be the join of all $I(v)$ with $v \in U$, i.e.

$$I(U) = \{t \in Z^{E(U)} \mid t_{|E(v)} \in I(v) \quad \text{for all} \quad v \in U\} \ .$$

$I(U)$ is the set of *local* solutions for $U$. Obviously $I(U \cup W) = I(U) \bowtie I(W)$ holds for arbitrary subsets $U$ and $W$ of vertices.

A (resolution) algorithm specifies the order of all the $\bowtie$ operations starting from the $I(v)$ with $v \in V$. We describe such an algorithm as a term with $\bowtie$ and all the $v$ in $V$. E.g. the term $(v_1 \bowtie v_2) \bowtie (v_3 \bowtie v_4)$ means that we first form $I(v_1) \bowtie I(v_2)$, $I(v_3) \bowtie I(v_4)$ and then the join of these two sets. There is a natural notion of subterm, the above term has the subterms $v_1 \bowtie v_2$, $v_3 \bowtie v_4$. We write $B \leq A$ if $B$ is subterm of $A$.

**Definition 2.** A (resolution) *algorithm* $A$ for the graph $G$ is a term $A$ with $\bowtie$ and all $v$ in $V$. Its *length* is the number of subterms of $A$, including $A$ and all $v \in V$.

For an arbitrary subterm $B$ of an algorithm $A$ let $V(B)$ denote the set of vertices occuring in $B$. So $I(V(B))$ is the result or the set of local solutions of the subterm $B$.

We define the logarithmic *complexity* $C_I(A)$ of an algorithm $A$ to be the maximal logarithmic size of the result of a subterm $B \leq A$:

$$C_I(A) \;=\; \log \max_{B \leq A} \#I(V(B)) \ .$$

The logarithmic complexity roughly corresponds to a *work load* $\#Z^{C_I(A)}$. Counting only the size of the largest intermediate result is justified since the length of algorithms will be small and the costs for a join operation corresponds to the cardinality of the operands.

The *average complexity* $C(A^{\mathrm{df}})$ of algorithm $A$ is defined to be

$$C(A^{\mathrm{df}}) \;=\; \log \ \mathrm{Exp}_I \max_{B \leq A} \#I(V(B)) \ .$$

As the distribution of $I$ is locally uniform with $r$ the expected value $\mathrm{Exp}_I$ depends on df and so does $C(A^{\mathrm{df}})$.

Let the *complexity* $C(G^{\mathrm{df}})$ of a computation graph $G^{\mathrm{df}}$ be the minimum of $C(A^{\mathrm{df}})$ over all algorithms $A$ for $G^{\mathrm{df}}$.

In most of cases, we are only interested in getting *one* solution of $G^{\mathrm{df}}$. If there are many solutions we can decrease the complexity by restricting the resolution process to random subsets of all the $I(v)$. Thus, for a given mapping

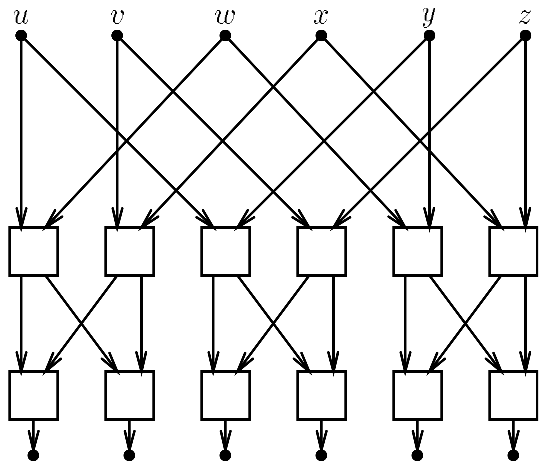


Figure 1: A powerful non-linear algorithm.

$\mathrm{df}'$ lower than df and a given interpretation $I$ of $(G^{\mathrm{df}}, Z)$, we consider a random sub-interpretation $I^{\mathrm{df}'}$ and a distribution defined by picking independently and uniformly random subsets $I^{\mathrm{df}'}(v)$ of $I(v)$ of size $\#Z^{\mathrm{df}'(v)}$. This means to consider an computation graph with $\mathrm{df}'$ instead of df. Note that the construction of $I^{\mathrm{df}'}$ preserves local uniformity and independence. In the following we will consider computation graphs with only one expected solution.

This formalization of black box cryptanalysis generalizes the use of exhaustive search and the birthday paradox locally on the intermediate values. Trying to solve exhaustively $f(x) = a$ for a given function $f$ with the unknown $x$ is formalized by the algorithm $a^0 \bowtie f^1$. (Here, degrees of freedom $\mathrm{df}(v)$ are denoted by a superscript on $v$). Trying to find *one* solution of $f(x, y) = a$ picking $x$ and $y$ at random can be formalized by $a^0 \bowtie f^1$ too, where the degree of freedom of $f^2$ has been decreased. Also, the use of the birthday paradox to get one solution of $f(x) = g(y)$ can be formalized by $f^{\frac{1}{2}} \bowtie g^{\frac{1}{2}}$.

In [11] we only considered *linear* resolution algorithms i.e. algorithms of the form $v_1 \bowtie (v_2 \bowtie (\ldots))$. It is tempting to believe that linear algorithms are already most powerful and that nothing can be gained from non-linear ones. The following counterexample shows this is not the case. Consider the network of figure 1 representing the computation graph of a supposed one-way function which maps 6 inputs $u$, $v$, $w$, $x$, $y$ and $z$ onto 6 outputs, all 24-bits long. The direction of the edges in figure 1 indicates the underlying network for computing the function. It is straightforward to imagine a linear attack to invert the function with logarithmic complexity 3, that is within work load $2^{72}$: guessing $u$, $w$ and $y$ by exhaustive search, one can solve the leftmost third of the graph from $u$ and $w$

and get $v$ and $x$, then the rightmost third from $w$ and $y$ and get $z$, then check if this is really a solution of the whole graph. It is easy to see that 3 is the lowest complexity for linear algorithms. On the other hand, one can solve the leftmost third from $u$ and $w$ and *independently* solve the rightmost third from $w$ and $y$ then join the two sets of partial solutions to get a set with rank 2 and check whether it contains a global solution. This is done with a logarithmic complexity 2, that is with work load $2^{48}$.

### 1.3 An approximative expression for the complexity

***Lemma 3.*** *Every subset $U \subset V$ of vertices satisfies* $\log \mathit{Exp}_I \#I(U) = G^{df}(U)$ .

*Proof.* We note that $\mathrm{Exp}_I \#I(U) = \sum_t \Pr[t \in I(U)]$ where the sum is over all tuples $t$ on $E(U)$. The event $\{t \in I(U)\}$ is the intersection of all the independent events $\{t_{|E(v)} \in I(v)\}$ for $v \in U$. There are $\#Z^{\#E(U)}$ many tuples on $E(U)$. Local uniformity and independence of the distribution for $I$ yields

$$\log \mathrm{Exp}_I \#I(U) = \#E(U) + \sum_{v \in U} (\mathrm{df}(v) - d(v))$$

which, as we have already seen, equals to $G^{\mathrm{df}}(U)$. □

***Theorem 4.*** *Every algorithm $A$ for $G^{df}$ with length $|A|$ satisfies*

$$\max_{B \le A} G^{df}(V(B)) \le C(A^{df}) \le \log|A| + \max_{B \le A} G^{df}(V(B)) .$$

*Proof.* By the definition of $C(A^{\mathrm{df}})$, and since a maximum of non-negative values is lower than their sum we have

$$C(A^{\mathrm{df}}) = \log \mathrm{Exp}_I \max_{B \le A} \#I(V(B)) \le \log \mathrm{Exp}_I \sum_{B \le A} \#I(V(B))$$

$$= \log \sum_{B \le A} \mathrm{Exp}_I \#I(V(B)) \le \log(|A| \cdot \max_{B \le A} \mathrm{Exp}_I \#I(V(B)))$$

$$= \log|A| + \max_{B \le A} \log \mathrm{Exp}_I \#I(V(B)) = \log|A| + \max_{B \le A} G^{\mathrm{df}}(V(B))$$

where the last equality comes from Lemma 3.
The first inequality of the claim is straightforward by Lemma 3. □

As an immediate consequence of Theorem 4 the expression

$$C'(G^{\mathrm{df}}) := \min_A \max_{B \le A} G^{\mathrm{df}}(V(B))$$

is a close approximation for the complexity $C(G^{\mathrm{df}})$. Interestingly $C'(G^{\mathrm{df}})$ does not depend on the alphabet $Z$. Now Theorem 4 implies the

***Corollary 5.*** $C'(G^{df}) \le C(G^{df}) \le \log \ell + C'(G^{df})$, *where $\ell$ is the length of an optimal algorithm.*

## 1.4 The spectral approach

We consider a locally invertible graph $G^{\frac{1}{2}d}$ so that $\mathrm{df}(v) = \frac{1}{2}d(v)$. Its quadratic form turns out to have properties similar to the Laplacian operator. In this context, lower bounds on the complexity can be proven in a similar way as in the expander graphs theory using a well-known link with the spectral values [12, 2, 1]. In this section let $G$ be an undirected graph with $n$ vertices and let $\lambda_1 \leq \ldots \lambda_n$ be the eigenvalues of the quadratic form $G^{\frac{1}{2}d}$.

***Lemma 6.*** *If the graph $G$ is connected then $\lambda_1 = 0$, $\lambda_2 > 0$, and every set $U$ of $c$ vertices satisfies $G^{\frac{1}{2}d}(U) \geq \lambda_2\, c\,(1 - c/n)$.*

*Proof.* 0 is an eigenvalue since $G^{\frac{1}{2}d}(U) = 0$ holds for all connected components $U$ of $G$. The fact that the quadratic form is positive and that $\lambda_2 > 0$, if $G$ is connected, is an easy algebra exercise left to the reader. We note that $\lambda_2$ is the smallest eigenvalue of the quadratic form in the hyperplane $V^{\perp}$ orthogonal to the set $V$ of all the vertices (*i.e.* the vector having all coordinates 1).

For an orthonormal basis of eigenvectors $v_1, \ldots, v_n$ with $v_1 = \frac{1}{\sqrt{n}}V$ we have (the dot $\cdot$ denotes the scalar product)

$$G^{\frac{1}{2}d}(U) = \sum_{i=1}^{n} \lambda_i (U \cdot v_i)^2 \geq \lambda_2 \sum_{i=2}^{n} (U \cdot v_i)^2 = \lambda_2((U \cdot U) - (U \cdot v_1)^2).$$

Now, the claim follows from $U \cdot U = c$ and $U \cdot v_1 = \frac{c}{\sqrt{n}}$ . $\square$

Then we get a lower bound:

***Theorem 7.*** *If the graph $G$ is connected then $C(G^{\frac{1}{2}d}) \geq \frac{2\lambda_2}{9} n$ .*

*Proof.* Let $A$ be an algorithm for $G^{\frac{1}{2}d}$ such that $C'(G^{\frac{1}{2}d}) = \max_{B \leq A} G^{\frac{1}{2}d}(V(B))$. Lemma 6 yields $C'(G^{\frac{1}{2}d}) \geq \max_{B \leq A} \lambda_2 \#(V(B))(1 - \#(V(B))/n)$. Let $x$ be an arbitrary integer between 0 and $n = \#V$. Every minimal subterm $B$ with the property that $\#(V(B)) \geq x$ also satisfies $\#(V(B)) \leq 2x$ since it can best be the join of two subterms which each contain $x - 1$ vertices. For such a subterm we have

$$\#(V(B))(1 - \#(V(B))/n) \geq \min_{x \leq y \leq 2x} y(1 - y/n)$$

and thus by Corollary 5

$$C(G^{\frac{1}{2}d}) \geq C'(G^{\frac{1}{2}d}) \geq \max_{0 \leq x \leq n} \min_{x \leq y \leq 2x} \lambda_2\, y(1 - y/n) = \frac{2\lambda_2}{9} n. \qquad \square$$

### 1.5 The symmetric approach

Similarly, we can apply a theorem due to Babai and Szegedy [4] used in context with Cayley graphs. We reformulate it in our context and refer to [4] for the proof. Let $G$ be an arbitrary undirected, edge-transitive graph with $n$ vertices, let $d$ be the harmonic mean of the degrees of the vertices and let $\delta$ be the average distance between two vertices.

***Lemma 8.*** *Every set $U$ of $c$ vertices satisfies $G^{\frac{1}{2}d}(U) \geq \frac{d}{2\delta}c\left(1-\frac{c}{n}\right)$.*

We remind that a graph is edge-transitive if every edge can be mapped onto any other one by the action of a graph automorphism. In such a graph, the vertices can only have one or two possible degrees. For two degrees $d_1$ and $d_2$, the harmonic mean is $\frac{2}{1/d_1+1/d_2}$.

Then, a straightforward application of the proof of Theorem 7 to Lemma 8 shows:

***Theorem 9.*** *Every undirected, edge-transitive graph $G$ satisfies $C(G^{\frac{1}{2}d}) \geq \frac{d}{9\delta}n$.*

## 2 Parallel FFT hashing

Two previous proposals of a cryptographic hash function based on the FFT network [8, 9] have been broken in [3, 13]. Then, by a joint effort, a family of hash function based on the same graph has been proposed in [10] and discussed in [11]. We will now prove the conjectures announced in the latter paper. Interestingly, the FFT network has been used by Massey for the SAFER encryption function [6].

Let $G_{k,s}$ be the graph defined by the set of vertices

$$V = \{v_{i,j}; 0 \leq i < 2^{k-1}, 0 \leq j \leq s\}$$

and the set of edges

$$E = \{\{v_{i,j}, v_{i,j+1}\}, \{v_{i,j}, v_{i\oplus 2^{j \bmod k-1},j+1}\}; 0 \leq i < 2^{k-1}, 0 \leq j < s\}.$$

$G_{k,s}$ is roughly the graph of the FFT network for $2^k$ values extended to $s+1$ layers. Considering all vertices as boxes with two inputs coming from a lower layer and two outputs going to a higher layer, this corresponds to a function with $2^k$ inputs entering in layer 0 and $2^{k-1}$ outputs going out from layer $s$. Given two message blocks $m$ and $m'$ with $2^{k-1}$ values, we let the $i$th value of each block enter to vertex $v_{i,0}$ and write the first output of $v_{i,s}$ as the $i$th value of the output string $h$. The mapping $(m, m') \mapsto h$ defines a compression function (see the example of $G_{3,2}$ illustrated on figure 2). Thus, we propose to study the family of the compression functions defined by $G_{k,s}$ and a distribution of interpretations $I$ defining boxes

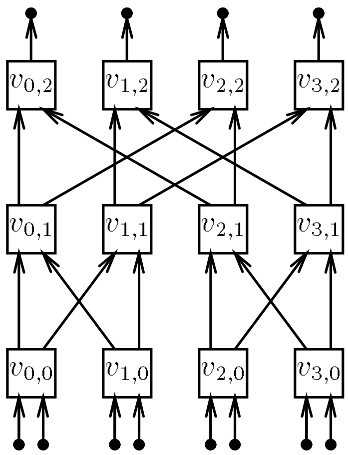


Figure 2: The $G_{3,2}$ compression functions family.

on the vertices. In [10], it is suggested to use the uniform distribution on all the multipermutations. As the purpose of the present paper is to study the graph properties we refer to the original papers for more informations.

For all the possible $I$ which make a function, we need to define $(s+1)2^{k-1}$ boxes. Choosing an alphabet with cardinality $q$, the number of bits to encode the input is $n = 2^k \log q$ whereas the length of the description of the function (that is the interpretation) is $(s+1)2^k q^2 \log q$. The family is quite huge, but we hope to find an interesting smaller sub-family in which the following analysis will be possible too. For instance if we take the same box for all the vertices $i, j$ and concatenate these boxes with independent random permutations along the inner edges of $G_{k,s}$ we decrease the length of the interpretation to $s^{2k} \log q!$ and we preserve local uniformity and independence. The aim of this study is to find the minimal $s$ for optimal security in context with black box cryptanalysis.

The one-wayness of the compression function relies on the hardness of finding, for given $m$ and $h$, one $m'$ such that $(m, m') \mapsto h$. It is formalized by the computation graph $G_{k,s}$ together with

$$\mathrm{df} : v_{i,j} \mapsto \begin{cases} 1 & \text{if } j = 0 \text{ or } j = s \\ 2 & \text{otherwise} \end{cases}$$

as one input of all $v_{i,0}$ and one output of all $v_{i,s}$ are already known that is with $\mathrm{df} = \frac{1}{2}d$. The exhaustive search consists in joining all $v_{i,0}$ to guess $m'$ and joining successively all the other vertices layer by layer. This has complexity $2^{k-1}$. So, we are interested in the *ratio* $C(G_{k,s}^{\frac{1}{2}d})/2^{k-1}$.

## 2.1 The upper bounds

***Theorem 1.*** *For $k-1 \leq s \leq 2k-2$ we have* $C\left(G_{k,s}^{\frac{1}{2}d}\right) \leq 2^{k-2}\left(1+2^{2-s}\right)$.

Thus, for $s < 2k-2$ there is an attack faster than exhaustive search. We conjecture that this inequality is in fact an equality for $s = 2k-2$, that is to say the exhaustive search is the best black box attack on $G_{k,2k-2}$.

*Proof.* First we show that $C\left(G_{k,k-1}^{\frac{1}{2}d}\right) \leq 2^{k-2}$. For this we guess the first $2^{k-2}$ inputs, that is we join the first $2^{k-2}$ vertices $v_{i,0}$. This allows to compute half of the edges, namely all edges adjacent to $v_{i,j}$ for $i < 2^{k-2}$. Then, the degree of freedom of all $v_{i,k-1}$ becomes 0, so that we can compute the other half edges backward and solve the graph. This has complexity $2^{k-2}$.

We can do similar things on $G_{k,s}$: we guess the first $2^{k-2}$ inputs and solve half of the vertices from layer 0 to layer $k-1$. Then, all connected subgraphs from layer $k-1$ to layer $s$ are isomorphic to $G_{k',k'-1}$ with $k' = s-k+2$ which can iteratively be solved within a complexity $2^{k'-2} = 2^{s-k}$. After having solved all of these subgraphs, the backward processing in $G_{k,s}$ finishes the resolution within complexity $2^{k-2} + 2^{k-s}$. □

In the following sections, we show how to apply the different approaches to find lower bounds. Finally, we prove that the ratio for $s = 2k-2$ is lower bounded by a constant $\frac{2}{3}$.

## 2.2 The lower bounds

### 2.2.1 Use of the spectral approach.

We use the notation introduced in sections 1.4 and 2.

***Lemma 2.*** $\lambda_2\left(G_{k,s}^{\frac{1}{2}d}\right) = \begin{cases} 4\sin^2\frac{\pi}{2(2k-1)} & \text{if } k \leq s < 2k-1 \\ 4\sin^2\frac{\pi}{2(s+1)} & \text{if } 2k-1 \leq s. \end{cases}$

The proof involves insipid tricky calculus which are not relevant here. It can be found in [15]. Then Theorem 7 implies:

***Corollary 3.*** $C\left(G_{k,s}^{\frac{1}{2}d}\right) \geq 2^{k-1}\begin{cases} \frac{8(s+1)}{9}\sin^2\frac{\pi}{2(2k-1)} & \text{if } k \leq s < 2k-1 \\ \frac{8(s+1)}{9}\sin^2\frac{\pi}{2(s+1)} & \text{if } 2k-1 \leq s. \end{cases}$

This suggests to use $s = 2k-1$ to get optimal security. The lower bound of the ratio is here equivalent to $\frac{2\pi^2}{9s}$.

### 2.2.2 Use of the symmetric approach.

Though $G_{k,s}$ is not edge-transitive, it is possible to use the symmetric approach for the case $s = k$. If we contract layers 0 and 1 following the rule that adjacent edges are merged, we get the same result (*i.e.* isomorphic) as if we add symmetrical edges between the first and the last edges of $G_{k-1,2k-5}$. The obtained graph $G'_{k-1,2k-5}$ turns out to be edge-transitive for $k \geq 3$. This graph has degree $d = 4$. Its average distance is given in [15]:

***Lemma 4.*** *The average distance of $G'_{k-1,2k-5}$ is at least $\frac{3}{2}(k-2)$.*

This allows to prove a technical corollary we mention for completeness:

***Corollary 5.*** *Letting $C_{lin}\left(G_{k,k}^{\frac{1}{2}d}\right)$ denote the linear complexity of $G_{k,k}^{\frac{1}{2}d}$, that is*

$$C_{lin}\left(G_{k,k}^{\frac{1}{2}d}\right) = \min_A C(A^{df})$$

*for all linear algorithm $A$, we obtain for $k \geq 3$*

$$C_{lin}\left(G_{k,k}^{\frac{1}{2}d}\right) \geq \frac{2^{k-1}}{3}\left(1 - \left(\frac{1}{k-2}\right)^2\left(3 + \frac{2}{2^{k-2}}\right)^2\right).$$

This establishes the lower bound $\frac{1}{3}$ for the ratio. Unfortunately, we could not prove a similar result for the general complexity following this approach.

*Proof.* (sketch) Any linear algorithm $A$ can be rewritten without increasing its complexity into an algorithm such that for any subterm $B$ which involves a vertex $v_{i,j}$ for $j = 0,\ 1,\ s-1$ or $s$, every of the other vertices which are merged with $v_{i,j}$ are either already involved in $B$ or will all be joined immediately after. In such an algorithm, there is at least one out of four consecutive subterms $B$ which has all of its merged *classes complete*, i.e. closed with respect to merging. Let $B'$ be the merged image of such a $B$ in $G'_{k-1,2k-5}$. The completeness property of $B$ implies $G_{k,k}(B) = G'_{k-1,2k-5}(B')$. Thus, using the Babai-Szegedy theorem together with the previous lemma, we obtain

$$G_{k,k}^{\frac{1}{2}d}(V(B)) \geq \frac{4}{3(k-2)}\#(V(B'))\left(1 - \frac{\#(V(B'))}{(2k-4)2^{k-2}}\right).$$

Noticing that

$$\#(V(B)) - 3 \cdot 2^{k-1} \leq \#(V(B')) \leq \#(V(B)).$$

and that at least one out of four consecutive $B$ satisfies the *completeness* property, we obtain the result. □

### 2.2.3 Use of the flow approach.

In the particular case of the graph $G_{k,2k-2}$, we can make an independent method dedicated to this graph based on the min-cut max-flow theorem.

***Lemma 6.*** *Let $V_0$ and $V_s$ be respectively the first and the last layer of $G_{k,2k-2}$. For every function $r$ from $V_0 \cup V_s$ to $\mathbb{Z}$ such that $|r(v)| \leq 2$ and $\sum_v r(v) = 0$ there exists a flow $f$ with source $s$, that is to say a function from the set $\bar{E}$ , the set of directed edges, to $\mathbb{Z}$ such that for all $v$ and $w$:*

1. $f(v,w) = -f(w,v)$,
2. $|f(v,w)| \leq 1$,
3. $\sum_u f(u,v) = \begin{cases} r(v) & \text{if } v \in V_0 \cup V_s \\ 0 & \text{otherwise.} \end{cases}$

*Proof.* Let $\{v_{i,j}, v_{i',j+1}\}$ be an edge with $j+1 \leq k-1$. We define

$$f(v_{i,j}, v_{i',j+1}) = \frac{1}{2} \underset{v_{i'',0}}{\text{Mean}}\ r(v_{i'',0})$$

where the arithmetic mean is taken over all vertices $v_{i'',0}$ such that there exists a straight path in $G_{k,2k-2}$ from $v_{i'',0}$ to $v_{i,j}$. $f(v_{i,j}, v_{i',j+1})$ is thus equal to half the mean of all $r(v_{i'',0})$ the incoming flow in $v_{i,j}$ from previous layers which is equally spread into its two outgoing edges. Hence, the incoming flow in all $v_{i,k-1}$ will be a constant. Defining $f(v_{i',j+1}, v_{i,j}) = -f(v_{i,j}, v_{i',j+1})$, it is easy to show that the above three conditions are satisfied for all edges before the layer $k-1$.

Similarly, for $j \geq k-1$, we define $f(v_{i,j}, v_{i',j+1})$ to be the half of the incoming flow in $v_{i',j+1}$ from the next layers starting at $V_s$. The conditions are satisfied for all edges after the layer $k-1$. The flow coming from all the upper layers to the layer $k-1$ is also equally spread into all the vertices. Then, due to the condition that the sum of all $r(v)$ is 0, the third condition is also satisfied in the layer $k-1$. □

Using the previous lemma we show:

***Lemma 7.*** *For any set $U$ in $G_{k,2k-2}$, if $c = \#(U \cap (V_0 \cup V_s))$ where $V_0$ is the first layer and $V_s$ is the last one, we have $G^{\frac{1}{2}d}_{k,2k-2}(U) \geq 2^{k-1} - |2^{k-1} - c|$.*

*Proof.* We note that $G^{\frac{1}{2}d}_{k,2k-2}(U) = G^{\frac{1}{2}d}_{k,2k-2}(V-U)$ (this comes from the fact that $V$ is in the kernel of the quadratic form). Thus, eventually replacing $U$ by $V-U$, we can assume $c \leq 2^{k-1}$. Now, for $v \in V_0 \cup V_s$, we can define $r(v)$ to be 2 if $v \in U$ and $r(v) = -\frac{2c}{2^k - c}$ otherwise. Since $s$ verifies the conditions of the lemma, there exists a flow with source $s$, hence with capacity $2c$. We note that $2G^{\frac{1}{2}d}_{k,2k-2}(U)$ is equal to the cardinality of the border of $A$, which is a cut for the flow. Hence, the min-cut max-flow theorem says $2G^{\frac{1}{2}d}_{k,2k-2}(U) \geq 2c$. □

***Corollary 8.*** $C\left(G^{\frac{1}{2}d}_{k,2k-2}\right) \geq \frac{2}{3}\,2^{k-1}$.

*Proof.* In a similar way than in the proof of the theorem 7, we get

$$C\left(G^{\frac{1}{2}d}_{k,2k-2}\right) \geq \max_{0\leq x\leq 2^k} \min_{x\leq y\leq 2x} \left(2^{k-1} - |2^{k-1} - x|\right)$$

which is equal to $\frac{2}{3}2^{k-1}$. □

A similar method applied on $G_{k,k-1}$ (basically, in taking only $V_0$ into account in the source of the flow) enables to prove:

***Corollary 9.*** $C\left(G^{\frac{1}{2}d}_{k,k-1}\right) \geq \frac{1}{3}\,2^{k-1}$.

This confirms the partial result obtained by the symmetric approach.

These results establish a constant ratio between the upper bounds and the lower bounds. Actually, the same method applied in the linear case proves the equality between the bounds which proves the conjecture in [11]. We conjecture that the upper bounds are still the real complexities in the general case for $s = 2k-2$. It shows one has to choose $s = 2k-2$ to get the optimal security for the $G_{k,s}$ compression function family.

# 3 Possible extensions and conclusion

The analysis on cryptographic primitives proposed here can be extended in a more general context. To allow several edge-domains to exist together in the same primitive, we can add the notion of edge-degree of freedom in the definition of the computation graphs. This would be the logarithm (in any basis) of the cardinality of the domain. We mention all the results still hold if we replace $d(v)$ by the sum of all the adjacent edge-degrees. We can also allow the value of an edge to be involved in more than two different vertices replacing the notion of graph by the notion of hypergraph.

We have proposed a new frame for the study of the security of cryptographic primitives based on a computation graph. We showed that the complexity of resolving a computation graph is related to the local expansion properties of the graph. This theory enables to prove the one-wayness of a family of compression functions with respect to the black box attacks. It can be applied to the compression functions based on the FFT network. It turns out that the function is the most secure possible (in context with the black box cryptanalysis) for a doubled FFT network, that is for $s = 2k-2$.

# Acknowledgment

We thank László Babai and Jacques Stern for helpful discussions.

# References


[1] N. Alon. Eigenvalues and Expanders. in *Combinatorica*, vol. 6, pp. 83–96, 1896.

[2] N. Alon, V. D. Milman. $\lambda_1$, Isoperimetric Inequalities for Graphs, and Superconcentrators. In *Journal of Combinatorical Theory*, vol. B 38, pp. 73–88, 1985.

[3] T. Baritaud, H. Gilbert, M. Girault. FFT Hashing is not Collision-free. In *Advances in Cryptology EUROCRYPT'92*, Balatonfüred, Hungary, Lectures Notes in Computer Science 658, pp. 34–44, Springer-Verlag, 1993.

[4] L. Babai, M. Szegedy. Local Expansion of Symmetrical Graphs. In *Combinatorics, Probability and Computing*, vol. 1, pp. 1–11, 1992.

[5] E. Biham, A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[6] J. L. Massey. SAFER K-64: a Byte-oriented Block-ciphering Algorithm. In *Fast Software Encryption – Proceedings of the Cambridge Security Workshop*, Cambridge, United Kingdom, Lectures Notes in Computer Science 809, pp. 1–17, Springer-Verlag, 1994.

[7] M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In *Advances in Cryptology CRYPTO'94*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 839, pp. 1–11, Springer-Verlag, 1994.

[8] C. P. Schnorr. FFT-Hashing: an Efficient Cryptographic Hash Function. Presented at the CRYPTO'91 Conference. Unpublished.

[9] C. P. Schnorr. FFT-Hash II, Efficient Cryptographic Hashing. In *Advances in Cryptology EUROCRYPT'92*, Balatonfüred, Hungary, Lectures Notes in Computer Science 658, pp. 45–54, Springer-Verlag, 1993.

[10] C. P. Schnorr, S. Vaudenay. Parallel FFT-Hashing. In *Fast Software Encryption – Proceedings of the Cambridge Security Workshop*, Cambridge, United Kingdom, Lectures Notes in Computer Science 809, pp. 149–156, Springer-Verlag, 1994.

[11] C. P. Schnorr, S. Vaudenay. Black Box Cryptanalysis of Hash Networks based on Multipermutations. In *Advances in Cryptology EUROCRYPT'94*, Perugia, Italy, Lectures Notes in Computer Science 950, pp. 47–57, Springer-Verlag, 1995.

[12] R. M. Tanner. Explicit Concentrators from Generalized $N$-gons. In *SIAM Journal of Algebraic Discrete Methods*, vol. 5, pp. 287–293, 1984.

[13] S. Vaudenay. FFT-Hash II is not yet Collision-free. In *Advances in Cryptology CRYPTO'92*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 740, pp. 587–593, Springer-Verlag, 1993.

[14] S. Vaudenay. On the Need for Multipermutations: Cryptanalysis of MD4 and SAFER. In *Fast Software Encryption*, Leuven, Belgium, Lectures Notes in Computer Science 1008, pp. 286–297, Springer-Verlag, 1995.

[15] S. Vaudenay. *La Sécurité des Primitives Cryptographiques*, Thèse de Doctorat de l'Université de Paris 7, Technical Report LIENS-95-10 of the Laboratoire d'Informatique de l'Ecole Normale Supérieure, 1995.