Rubik's for Cryptographers

Christophe Petit*and Jean-Jacques Quisquater
UCL Crypto Group

e-mails: christophe.petit@uclouvain.be, jjq@uclouvain.be

January 19, 2011

Abstract

Hard mathematical problems are at the core of security arguments in cryptography. In this paper, we study mathematical generalizations of the famous Rubik's cube puzzle, namely the factorization, representation and balance problems in non-Abelian groups. These problems arise naturally when describing the security of Cayley hash functions, a class of cryptographic hash functions with very interesting properties. The factorization problem is also strongly related to a famous long-standing conjecture of Babai, at the intersection of group theory and graph theory. A constructive proof of Babai's conjecture would make all Cayley hash functions insecure, but on the other hand it would have many positive applications in graph theory and computer science. In this paper, we classify existing attacks against Cayley hash functions and we review known results on Babai's conjecture. Despite recent cryptanalytic progress on particular instances, we show that the factorization, representation and balance problems presumably remain good sources of cryptographic hard problems. Our study demonstrates that Cayley hash functions deserve further interest by the cryptography community.

Disclaimer. *This paper contains essentially no new result but it rather collects and organizes all the results that were independently found by two distinct scientific communities on the same problems. Between September 2009 and May 2010, the first author gave a sequence of talks to a cryptographic audience, entitled "Hash functions and Cayley graphs: the end of the story?". Surprisingly, many cryptographers seemed to either ignore the beautiful Cayley hash construction, or believe that it had been definitively broken. The very positive feedback received after these talks motivated us to write this survey and to complete it with known results on Babai's conjecture.*

1 Introduction

Presumably hard mathematical problems stand at the core of modern cryptography. A typical security proof for a cryptographic protocol relates its resistance against a particular attack to the hardness of some mathematical problem. Very few problems survived the thorough analysis of scientists, the most established ones being the integer factoring problem and the discrete logarithm problem on finite fields and elliptic curves. Other problems have been suggested, related for example to hyperelliptic curves, lattices [59], error-correcting codes [46] or multivariate

*Research Fellow of the Belgian Fund for Scientific Research (F.R.S.-FNRS) at Université catholique de Louvain (UCL).

polynomial equations [51]. They are currently less trusted than the three previous ones but they might join or replace them in the future.

The Rubik's cube is a famous 3D mechanical puzzle. It is notoriously "hard", but of course not in the cryptographic sense. Computer programs solve it instantaneously, and even human champions need less than ten seconds. The Rubik's cube has a strong mathematical structure: the set of its configurations is a subgroup of some finite permutation group. Solving the Rubik's cube amounts to solving a *factorization problem* in this subgroup.

To any finite (multiplicative) group $\mathcal{G}$ and any set $\mathcal{S}$ of elements generating this group, we can associate the problem of *factoring* any element of the group as a "short" product of elements from $\mathcal{S}$. Hardness results on this problem are only known for a few combinations of groups and generating sets. For some reasons that will be made clear below, the factorization problem is very easy in the case of the Rubik's cube. On the other hand, it is equivalent to the discrete logarithm problem in Abelian groups [6], and the related problem of finding *the shortest* factorization is NP-hard for permutation groups [27, 35].

The factorization problem in non-Abelian groups may also be seen as an *explicit* version of a conjecture of Babai stating that the diameter of any Cayley graph of a non-Abelian simple group is "small" [2]. This famous conjecture has recently been proved for a few groups but using *non explicit* techniques [33, 25, 32]. On the other hand, explicit factorization algorithms are known in all finite simple non-Abelian groups but only for particular generators [4, 36, 39, 55, 60, 38].

The factorization problem in non Abelian groups was introduced to the cryptography community via *Cayley hash functions*, a class of cryptographic hash functions based on Cayley graphs. Hash functions are a very important cryptographic primitive, used for digital signatures, message authentication codes and many other applications. Although a few hash functions are based on mathematical problems [21, 45], the most popular ones have an *ad hoc* design somehow similar to a block cipher. Recent attacks on the standard SHA-1 prompted NIST to launch a competition for a new secure hash algorithm [1].

At Eurocrypt'91, Zémor introduced a hash function based on a Cayley graph of the group $SL(2, \mathbb{F}_p)$ [67]. The main security properties of this function are strongly related to the corresponding factorization problem and to the related *representation* and *balance* problems. Besides its nice mathematical structure, the function has the advantages of reasonably good efficiency and natural parallelism. Unfortunately, its factorization problem was solved by Tillich and Zémor, who then proposed new parameters in the group $SL(2, \mathbb{F}_{2^n})$ [64, 62]. Thirteen years later, the design was rediscovered and new parameters coming from LPS and Morgenstern graphs were suggested [44, 49, 19, 54]. Recently, the LPS, Morgenstern and Tillich-Zémor hash function have been broken as well, giving the feeling to the cryptography community that all Cayley hash functions are necessarily insecure.

In this paper, we show that the factorization, representation and balance problems in non Abelian groups still appear as potentially hard problems for general parameters. We first review and classify known attacks against particular Cayley hash function proposals. We show that the techniques used for these particular parameters can hardly be used against more general functions. We then cover the progress on Babai's conjecture. We show that despite 20 years of active research, constructive proofs of the conjecture are only known for a few particular parameters. Finally, we propose a set of parameters leading to both secure and efficient cryptographic hash functions. Our study demonstrates that the Cayley hash function design is still particularly appealing and that it deserves further interest by the cryptography community.

The paper is organized as follows. In Section 2, we recall the Cayley hash function design and we define the balance, representation and factorization problems. In Section 3, we review

the cryptanalysis of Cayley hash functions. In Section 4, we review known results on Babai's conjecture. We propose a new cryptanalytic challenge in Section 5 and we conclude in Section 6.

2 Cayley hash functions

In this section, we first review the construction of hash functions based on Cayley graphs. We then define the balance, representation and factorization problems, and we justify that they are potentially hard. We finally explicit the connection with the Rubik's cube.

2.1 Construction and main features

In cryptography, a *hash function* is a function that takes as inputs bitstrings of arbitrary length and that returns bitstrings of fixed, finite, small length. Such a function is typically required to be collision resistant, second preimage resistant and preimage resistant.

Definition 1 *Let $n \in \mathbb{N}$ and let $H : \{0,1\}^* \to \{0,1\}^n : m \to h = H(m)$. The function H is said to be* [47]

- ***collision resistant*** *if it is "computationally hard" to find $m, m' \in \{0,1\}^*$, $m' \neq m$, such that $H(m) = H(m')$.*
- ***second preimage resistant*** *if given $m \in \{0,1\}^*$, it is "computationally hard" to find $m' \in \{0,1\}^*$, $m' \neq x$, such that $H(m) = H(m')$;*
- ***preimage resistant*** *if given $h \in \{0,1\}^n$, it is "computationally hard" to find $m \in \{0,1\}^*$ such that $h = H(m)$;*

Remark. The words "computationally hard" can be understood in two different ways. From a practical point of view, they mean that no big cluster of computers can perform the task. A computational complexity of 2^{80} operations is currently considered out of reach [29]. On the other hand, from a theoretical point of view, it means that no probabilistic algorithm that runs in time polynomial in n can succeed in performing the task for large values of the parameter n with a probability larger than the inverse of some polynomial function of n [30].

Given a (multiplicative) group $\mathcal{G}$ and a subset $\mathcal{S} = \{s_1, ..., s_k\}$ thereof, their *Cayley graph* is a k-regular graph that has one vertex for each element of $\mathcal{G}$ and one edge between two vertices v_1 and v_2 if and only if the corresponding group elements g_{v_1}, g_{v_2} satisfy $g_{v_2} = g_{v_1} s_i$ for some $s_i \in \mathcal{S}$. We can build a hash function from this graph as follows. The message m is first written as a string $m = m_1...m_N$ where $m_i \in \{1, ..., k\}$. Then the group product

$$h = s_{m_1} s_{m_2} ... s_{m_N}$$

is computed and it is mapped onto a bitstring. A hash function constructed this way is called a *Cayley hash function*. The initial and final transformations do not influence the security. In the remaining of the paper, we will consider hash functions as functions from $\{1, ..., k\}^*$ to $\mathcal{G}$.

Classical hash functions like SHA are designed in a very different way: they mix pieces of the message again and again until the result looks sufficiently random. Somehow, the "block-cipher-like" design of these functions looks like a sack of nodes that discourages its study outside the cryptography community. In contrast, Cayley hash functions have a clear, simple and elegant mathematical design. As we will see below, their main security properties are strongly related

to interesting mathematical problems with a history of 20 years. Moreover, the computation of a Cayley hash value can be very easily parallelized: large messages can be cut into various pieces distributed to different computing units, and the associativity of the group ensures that the final result can be recovered from all partial products. Efficiency depends on the group and the generators used. Cayley hash functions are rather slow to compute for most parameters, but in some contexts they perform better than SHA-1 [23]. Malleability properties [9] are another drawback. For example, given the hash value of m and m', it is possible to compute the hash value of $m||m'$. However, heuristic additional design can solve this problem [52].

The first instance of a Cayley hash function was introduced by Zémor at Eurocrypt'91 [67]. It uses the group $SL(2, \mathbb{F}_p)$ and the set $\mathcal{S} = \{\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)\}$ where p is a prime number of 160 bits. Soon after, Tillich and Zémor cryptanalysed this scheme and replaced its parameters by $\mathcal{G} = SL(2, \mathbb{F}_{2^n})$ and the set $\mathcal{S} = \{\left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} X & X+1 \\ 1 & 1 \end{smallmatrix}\right)\}$ where $\mathbb{F}_{2^n} \approx \mathbb{F}_2[X]/(p(X))$ and $p(X)$ is an irreducible polynomial of degree about 160 over $\mathbb{F}_2$ [64, 62]. In both cases, $\mathcal{S}$ contains two elements with "small" coefficients to accelerate the matrix multiplications.

In 2007, Charles et al. rediscovered the design and suggested using the Lubotzky-Philips-Sarnak (LPS) Ramanujan graphs for their optimal expanding properties [44, 19]. For this construction, the group $\mathcal{G}$ is $PSL(2, \mathbb{F}_p)$ where p is a prime of 160 bits and the set $\mathcal{S}$ contains all the $q+1$ elements with reduced norm equal to some small prime q. Morgenstern Ramanujan graphs seemed appealing for the same reasons [49, 54]. They use $\mathcal{G} = PSL(2, \mathbb{F}_{2^n})$ with $\mathbb{F}_{2^n} \approx \mathbb{F}_2[X]/(p(X))$ and $p(X)$ is an irreducible polynomial of even degree about 160. The set $\mathcal{S}$ contains the 3 elements of reduced norm $1+X$. For both LPS and Morgenstern, the sets $\mathcal{S}$ have a lot of symmetry since they contain exactly *all* the elements with the same (small) norms.

As we will see in Section 3, the particular choices for the generators in Zémor, Tillich-Zémor, LPS and Morgenstern hash functions have facilitated their cryptanalysis.

2.2 Balance, factorization and representation problems

We now introduce the mathematical problems at the core of the security of Cayley hash functions.

Definition 2 *Let $\mathcal{G}$ be a group and let $\mathcal{S} = \{s_1, ..., s_k\} \subset \mathcal{G}$ be a set generating this group. Let $L \in \mathbb{Z}$ be "small".*

- ***Balance problem:*** *Find an "efficient" algorithm that returns two words $m_1...m_\ell$ and $m'_1...m'_{\ell'}$ with $\ell, \ell' < L$, $m_i, m'_i \in \{1, ..., k\}$ and $\prod s_{m_i} = \prod s_{m'_i}$.*
- ***Representation problem:*** *Find an "efficient" algorithm that returns a word $m_1...m_\ell$ with $\ell < L$, $m_i \in \{1, ..., k\}$ and $\prod s_{m_i} = 1$.*
- ***Factorization problem:*** *Find an "efficient" algorithm that given any element $g \in \mathcal{G}$, returns a word $m_1...m_\ell$ with $\ell < L$, $m_i \in \{1, ..., k\}$ and $\prod s_{m_i} = g$.*

Remark. Again, the word "small" can be understood in two different ways. Messages larger than a few gigabytes can hardly make sense in practice. On the other hand, from a theoretical point of view, "small" means polylogarithmic in the size of the group, considering a family of groups with increasing sizes. The word "efficient" means the opposite of "computationnally hard".

Without the length constraint, the representation problem would be trivial since $s^{ord(s)} = 1$ for any $s \in \mathcal{G}$. With the stronger requirement of finding a product of *minimal* length, it becomes

NP-hard [27, 35]. The factorization problem was described by Lubotzky as a "non-commutative analog of the discrete logarithm problem ([43], p.102). Indeed, both the representation and the factorization problems are equivalent to the discrete logarithm problem in Abelian groups if we forbid trivial solutions [6]. On the other hand, the balance problem becomes trivial in Abelian groups.

In general, the factorization problem is at least as hard as the representation problem, which is at least as hard as the balance problem. Clearly, a Cayley hash function is collision resistant if and only if the balance problem is hard; it is second preimage resistant only if the representation problem is hard; it is preimage resistant if and only if the corresponding factorization problem is hard. In the remaining of this paper, we will freely move between the security properties of Cayley hash functions and the hardness of the corresponding problems.

The balance, representation and factorization problems are related to famous problems in group theory. The simplest of these problems is Dixon's [26]: given a group $\mathcal{G}$ and a set $\mathcal{S}$ of randomly chosen elements, what is the probability that those elements generate the group? The answer is now known for all finite non Abelian simple groups [37, 42]. When the elements of $\mathcal{S}$ generate $\mathcal{G}$, it becomes natural to ask for the *diameter* of the corresponding Cayley graph. A logarithmic lower bound $\log_{|\mathcal{S}|} |\mathcal{G}|$ can be easily derived, but we don't know whether the bound is tight in general.

A large source of graphs with logarithmic diameter is provided by *expander graphs* [34]. Roughly speaking, an expander graph is a regular graph such that any set of its vertices has a comparatively large set of neighbors. Expander graphs are very important for computer science, with a wide range of applications. An intense research effort in the last ten years recently culminated in proving that for any non-Abelian finite simple group, there exists a symmetric set of generators such that the corresponding Cayley graph is an expander [16].

Since we cannot hope for logarithmic diameter in general, another problem that has often been considered is the problem of finding *optimal* sets of generators: generators providing Cayley graphs with diameters as close as possible from the lower bound [4, 38]. Finally, Babai conjectured that the diameter of any undirected Cayley graphs of any non Abelian simple group is polylogarithmic in the size of the group [2].

Babai's conjecture has been one of the most challenging open problems in group theory. The factorization problem can be seen as providing a *constructive* proof of Babai's conjecture, and it is therefore at least as hard as proving it. "Small" factorizations always exist in a Cayley graph with logarithmic or polylogarithmic diameter, but they are not necessarily computed by an efficient algorithm. Babai's conjecture has recently been proved in many special linear groups [33, 25, 32]. Under some wide condition on the generator sets, these groups are even expanders [11, 13]. Unfortunately, the proofs of these results use non constructive techniques from combinatorics and representation theory. Babai and Hayes [3] also proved the conjecture for *almost all* generators sets of the symmetric group, but the core of the proof, a Chebyshev argument, is non constructive.

Constructive proofs of the conjecture are known for all finite simple non-Abelian groups but only for *particular* sets of generators [4, 36, 39, 55, 60, 38]. We sketch the proofs of these results in Section 4. They do not generalize to arbitrary sets of generators. To the best of our knowledge, the only groups where explicit factorizations can be computed for more than marginal sets of generators are the groups $PSL(2, \mathbb{Z}/p^k\mathbb{Z})$ and $SL(2, \mathbb{Z}/p^k\mathbb{Z})$ for "small" p [24, 28]. We will explain in Section 3.2 what makes these groups "particular".

After 20 years of research by the mathematics, computer science and cryptography communities, the hardness of the factorization problem in general is still a widely open problem.

The challenge for cryptographers is to find groups $\mathcal{G}$ and sets $\mathcal{S}$ for which the group operations with elements of $\mathcal{S}$ are efficient but the factorization, representation and balance problems are difficult to solve.

2.3 A "toy" example: the Rubik's cube

We now explicit the link between the factorization problem and the Rubik's cube. Let E be the set of all possible configurations of the Rubik's cube, including configurations obtained by disassembling and reassembling it. The permutation group $\mathcal{G}$ on E acts naturally on the cube: to each $g \in \mathcal{G}$ we can associate the image by g of the initial configuration of the cube. The *Rubik's* group is the subgroup $\mathcal{G}_R$ that is generated by the 6 elementary rotations of the faces. The Rubik's group has order $|\mathcal{G}_R| = \frac{1}{12} 12!8!3^8 2^{12}$ and it is isomorphic to $(\mathbb{Z}_3^7 \times \mathbb{Z}_2^{11}) \rtimes ((A_8 \times A_{12}) \rtimes \mathbb{Z}_2)$ where $\times$ and $\rtimes$ are respectively the direct and semi-direct group products [20]. Solving the Rubik's cube amounts to solving the factorization problem for the group $\mathcal{G}_R$ and the set $\mathcal{S}$ containing the 6 rotations of the faces.

3 Cryptanalysis of Cayley hash functions

In this section, we review known attacks on the balance, representation and factorization problems. We first describe generic attacks on Merkle-Damgård hash functions, subgroup attacks, trapdoor attacks and lifting attacks. Then we go to more elaborate cryptanalysis and we finally explain why these problems are still worth studying in our sense.

3.1 Generic attacks on Merkle-Damgård hash functions

Like any hash function, Cayley hash functions are susceptible to exhaustive search attacks solving the factorization problem in time roughly $|\mathcal{G}|$, and to birthday attacks [65] solving the balance problem in time roughly $|\mathcal{G}|^{1/2}$. Moreover, Cayley hash functions are a particular case of Merkle-Damgård hash functions [22]. The "compression function" $H : \mathcal{G} \times \{1, ..., k\} \to \mathcal{G}$ sends an intermediary product $s_{m_1} s_{m_2} ... s_{m_n}$ and a k-digit m_{n+1} to the next intermediary product $s_{m_1} s_{m_2} ... s_{m_{n+1}}$. Because this compression function can be efficiently inverted by exhaustive search, the factorization problem can be solved in time roughly $|\mathcal{G}|^{1/2}$ with a meet-in-the-middle attack [58]. Since $|\mathcal{G}_R| \approx 2^{65.2}$, the Rubik's cube can already be solved by these simple techniques.

3.2 Subgroup attacks

The group structure of Cayley hashes opens the door to even more efficient attacks. Let us suppose that $\mathcal{G}$ has a subgroup tower decomposition $\mathcal{G} = \mathcal{G}_0 \supset \mathcal{G}_1 \supset \mathcal{G}_2 \supset ... \supset \mathcal{G}_N = \{1\}$, and that $|G_i|/|G_{i+1}|$ is "small" for all i. Given $\mathcal{S} = \{s_1, ..., s_k\}$, the representation problem can be solved as follows. We generate random products of the s_i until we get an element $s_1^{(1)} \in \mathcal{G}_1$, and we repeat the operations until we get a set $\mathcal{S}^{(1)} = \{s_1^{(1)}, ..., s_{k'}^{(1)}\}$ that can generate all the elements of $\mathcal{G}_2$. We then recursively repeat the procedure starting from the group $\mathcal{G}_1$ and the set $S^{(1)}$. A representation with the elements of $\mathcal{S}$ can be obtained by substitutions. The complexity of this attack is roughly $\max_i |\mathcal{G}_i|/|\mathcal{G}_{i+1}|$, but it can be reduced to $\max_i (|\mathcal{G}_i|/|\mathcal{G}_{i+1}|)^{1/2}$ using a meet-in-the-middle strategy as follows. We obtain $s_1^{(1)} \in G_1$ more efficiently if we generate random products g_j of the s_i and random products h_j of the s_i^{-1} until getting a couple $(g_j, h_{j'})$ such that $s_1^{(1)} := g_j h_{j'}^{-1} \in \mathcal{G}_1$. These attacks can be extended to solve the factorization problem as well.

Subgroup attacks were first introduced by Camion against an early scheme of Bosset [10, 17]. At Crypto'00, Steinwandt et al. attacked the Tillich-Zémor hash function as follows. Assuming $n = n_1 n_2$, the group $SL(2, \mathbb{F}_{2^{n_1}})$ and its conjugates are subgroups of $SL(2, \mathbb{F}_{2^n})$. Matrices of these subgroups have "small" orders, and they can be easily identified since their traces belong to $\mathbb{F}_{2^{n_1}}$ [61]. The "level by level" resolution method for the Rubik's cube is also a subgroup attack: each level can be associated to the subgroup of the Rubik's group containing all the permutations that preserve the levels solved so far. Since the order of $\mathcal{G}_R$ is very smooth, many other subgroup attacks could be constructed against the Rubik's cube. Finally, we observe that the factorization algorithms of Dinai [24] for the groups $SL(2, \mathbb{Z}/p^k\mathbb{Z})$ is a subgroup attack in essence, with the subgroup tower $SL(2, \mathbb{Z}/p^k\mathbb{Z}) \supset SL(2, \mathbb{Z}/p^{k-1}\mathbb{Z}) \supset ... \supset SL(2, \mathbb{Z}/p\mathbb{Z}) \supset \{I\}$. The case of $PSL(2, \mathbb{Z}/p^k\mathbb{Z})$ is similar [28].

Subgroup attacks decompose the factorization, representation and balance problems into smaller similar problems in the left, right or bilateral quotients of $\mathcal{G}_i$ by $\mathcal{G}_{i+1}$. Solving these smaller problems is sufficient to solve the original problems. The condition that $|\mathcal{G}_i|/|\mathcal{G}_{i+1}|$ is "small" for all i is sufficient but not necessary. In fact, if the quotient of some $\mathcal{G}_i$ by $\mathcal{G}_{i+1}$ has a "nice" and "manageable" structure, like an Abelian additive group or the multiplicative group of a "not too large" finite field, the problems can be solved in that quotient much more efficiently than by exhaustive or birthday searches. In [56], Petit et al. studied the diagonal and triangular subgroups of $SL(2, \mathbb{F}_{2^n})$ and all their conjugates. For the Tillich-Zémor hash function, they showed that finding two messages hashing to the same subgroup conjugate to the triangular subgroup of $SL(2, \mathbb{F}_{2^n})$ was not harder than finding a collision for the whole function.

3.3 Trapdoor attacks

A *trapdoor attack* assumes a particular situation where the person who chooses the group $\mathcal{G}$ and the set $\mathcal{S}$ has an incentive to cheat. In [61], Steinwandt et al. gave the following trapdoor attack on the Tillich-Zémor hash function. They generate random products of $\left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} X & X+1 \\ 1 & 1 \end{smallmatrix}\right)$ over $\mathbb{F}_2[X]$ (without modular reductions) and compute the trace of the resulting matrix. Then, they choose as modular polynomial p a divisor of the trace that has a sufficiently large degree. Therefore, the matrix has trace 0 modulo p hence it is of order 2 in the group $SL(2, \mathbb{F}_2[X]/(p(X))$. Keeping the factorization of the random matrix secret, they are therefore able to compute a solution to the representation problem even if nobody else can do so.

3.4 Lifting attacks

Modular reductions are essential in the hash functions of Zémor, Tillich-Zémor, LPS and Morgenstern. Without modular reductions, the elements of $\mathcal{S}$ would generate a free group. The outputs would "grow" indefinitely with the length of the message. Moreover, they would belong to a subset of a matrix ring with *unique factorization*, and the message digits could be recovered one by one from right to left. Thanks to modular reductions, some information is lost in the products, the group generated by $\mathcal{S}$ is no longer free and factorization is no longer trivial. The goal of a lifting attack is to "undo" the mixing work performed by the reductions.

Lifting attacks have been the most powerful technique against Cayley hash functions. They were first used by Tillich and Zémor against Zémor hash function [64]. The crucial observation for their attack is that any matrix of $SL(2, \mathbb{Z}_+)$ is a product of $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$. Indeed, the well-known *Euclidean algorithm* on integers can be written in matrix form

$$\begin{pmatrix} a_{i-2} \\ a_{i-1} \end{pmatrix} = \begin{pmatrix} 1 & q_{i-1} \\ & 1 \end{pmatrix} \begin{pmatrix} 1 & \\ q_i & 1 \end{pmatrix} \begin{pmatrix} a_i \\ a_{i+1} \end{pmatrix}$$

and moreover $\left(\begin{smallmatrix} 1 & q \\ 0 & 1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)^q$ and $\left(\begin{smallmatrix} 1 & 0 \\ q & 1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)^q$. The factorization problem is solved as follows: given a matrix $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in SL(2, \mathbb{F}_p)$, a matrix $\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) \in SL(2, \mathbb{Z}_+)$ that reduces to M modulo p is selected. If $A \leq B$ the Euclidean algorithm is applied to (A, B), else it is applied to (C, D). The length of the factorization is the sum of the partial quotients. Tillich and Zémor argue that this sum is "small" on average and that "large" sums are "unlikely" to appear. Independently, Larsen [39] provided an algorithm that returns factorizations of length $O(\log p \log\log p)$ in time polynomial in $\log p$ and with a constant probability.

The cryptanalysis of Zémor hash function is particularly simple because the set of matrices generated by $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ is *dense* in $SL(2, \mathbb{Z}_+)$ (actually it is equal to $SL(2, \mathbb{Z}_+)$). This observation led Tillich and Zémor to propose a new scheme with $\mathcal{G} = SL(2, \mathbb{F}_{2^n})$ and $\mathcal{S} = \{\left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} X & X+1 \\ 1 & 1 \end{smallmatrix}\right)\}$. They showed that the *density* of the set generated by $\mathcal{S}$ in $SL(2, \mathbb{F}_2[X])$ is about 2^{-n}. Given a matrix $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in SL(2, \mathbb{F}_{2^n})$, it seems therefore harder to find a matrix $\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) \in SL(2, \mathbb{F}_2[X])$ that reduces to M modulo p, and can be written as a product of $\left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} X & X+1 \\ 1 & 1 \end{smallmatrix}\right)$. We will see in Section 3.6 that this scheme was nevertheless broken by a more elaborate lifting attack.

A lifting strategy was also used by Tillich and Zémor against the LPS hash function [63]. In this attack, the elements of $PSL(2, \mathbb{F}_p)$ are lifted to elements of $SL(2, \mathbb{Z}[i])$ where $i^2 = -1$. Unlike in the attack against Zémor hash function, the lifts of the generators do not generate the whole $SL(2, \mathbb{Z}[i])$, but only a subset Ω of very small density. However, the lifting attack is still possible because Ω has a very simple characterization. Since Ω contains all the elements of norm q, Ω contains exactly all the elements whose norms are powers of q. Tillich and Zémor solve the representation problem by lifting the identity to Ω, which amounts to solving the *norm equation*

$$(\lambda + wp)^2 + 4(xp)^2 + 4(yp)^2 + 4(zp)^2 = q^e$$

with $\lambda, w, x, y, z, e \in \mathbb{Z}$ (once the identity is lifted, factoring it becomes trivial). The equation is solved as follows: they arbitrarily fix $e = 2e'$ with $q^{e'} > 4p^2$, and $\lambda + wp = q^{e'} - 2mp^2$ for some m. The norm equation can be "simplified by $4p^2$", resulting in an equation of the form

$$x^2 + y^2 + z^2 = N$$

for some N depending on m. Finally, the last equation is solved by generating random values for z, checking that the resulting equation $x^2 + y^2 = N' := N - z^2$ has a solution (a sufficient condition is that all the prime factors of N' congruent to 3 modulo 4 appear an even number of times in the factorization of N'), and finally solving this equation with the continued fraction method (or equivalently, with the *Euclidean algorithm*). A similar attack was developed against the Morgenstern hash function [53].

3.5 Preimages for LPS and Morgenstern hash functions

The cryptanalysis of LPS hash function was extended to solve the factorization problem [53]. Following the approach of Tillich and Zémor, finding preimages for the LPS hash function amounts to solving the norm equation

$$(A\lambda + wp)^2 + (B\lambda + xp)^2 + (C\lambda + yp)^2 + (D\lambda + zp)^2 = q^e$$

where A, B, C, D are fixed and $\lambda, w, x, y, z, e \in \mathbb{Z}$. For $A = 1, B = C = D = 0$ this equation particularizes the previous one, but the general equation seems much harder to solve. Petit et al. therefore introduced a two-steps strategy that combines ideas from lifting attacks and subgroup attacks. First, they write any matrix as a product of diagonal matrices and the elements of S.

The same ideas apply to Morgenstern hash function. Second, they compute preimages of any diagonal matrix by solving the norm equation

$$(A\lambda + wp)^2 + (B\lambda + xp)^2 + (yp)^2 + (zp)^2 = q^e.$$

Their method is to first fix λ to satisfy the equation modulo p, then w and x to satisfy it modulo p^2, and finally y and z to satisfy it over the integers. The resulting algorithm is probabilistic. The authors provide good arguments (but no definite proof) that it finishes in polynomial time and produces messages of logarithmic length.

3.6 Cryptanalysis of the Tillich-Zémor hash function

Despite the partial attacks described above, the Tillich-Zémor hash function resisted 15 years of cryptanalysis attempts until it was definitely broken by Grassl et al. [31] and Petit and Quisquater [55]. An important observation for both attacks is that the hardness of the factorization, representation and balance problem does not change if we replace the generators $\left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} X & X+1 \\ 1 & 1 \end{smallmatrix}\right)$ by $\left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right)^{-1} \left(\begin{smallmatrix} X+1 & 1 \\ 1 & 0 \end{smallmatrix}\right) \left(\begin{smallmatrix} X & 1 \\ 1 & 0 \end{smallmatrix}\right) = \left(\begin{smallmatrix} X+1 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. The new matrices are strongly related to the *Euclidean algorithm* in $\mathbb{F}_2[X]$ since an iteration of this algorithm can be written in a matrix form

$$\begin{pmatrix} a_{i-1} \\ a_i \end{pmatrix} = \begin{pmatrix} q_i & 1 \\ 1 & \end{pmatrix} \begin{pmatrix} a_i \\ a_{i+1} \end{pmatrix}.$$

Mesirov and Sweet [48] proved that for any irreducible polynomial $p \in \mathbb{F}_2[X]$, there exists a polynomial $q \in \mathbb{F}_2[X]$ such that all partial quotients appearing in the execution of the algorithm on p and q are X or $X+1$. Their proof implicitly contains an algorithm computing this q. In this attack, Grassl et al. apply this algorithm to the polynomial p defining the field, to obtain a preimage m to some matrix $\left(\begin{smallmatrix} p & q \\ c & d \end{smallmatrix}\right) = \left(\begin{smallmatrix} 0 & q \\ c & d \end{smallmatrix}\right) \bmod p$. They finally show how simple manipulations of this message lead to a collision.

The attack is reminiscent of the lifting attack. The density obstacle mentioned in Section 3.4 is bypassed by lifting a matrix $\left(\begin{smallmatrix} 0 & q \\ c & d \end{smallmatrix}\right)$ to $SL(2, \mathbb{F}_2[X])$ without constraining the values of q, c and d modulo p. The key tool for the lifting step is Mesirov and Sweet's algorithm.

Petit and Quisquater extended Grassl et al.'s attack to solve both the representation and the factorization problems. First, they observe that another simple manipulation of the preimage of $\left(\begin{smallmatrix} 0 & q \\ c & d \end{smallmatrix}\right)$ leads to a preimage of a matrix $\left(\begin{smallmatrix} 1 & 0 \\ \alpha_0 & 1 \end{smallmatrix}\right)$ for some $\alpha_0 \in \mathbb{F}_{2^n}$. This last matrix has order 2, leading to a solution to the representation problem. Second, they show how to write any matrix as a product of elements of S and of matrices of the form $\left(\begin{smallmatrix} 1 & 0 \\ \alpha & 1 \end{smallmatrix}\right)$. Finally, they show how to compute any matrix of this form from a small set of precomputed matrices $\left(\begin{smallmatrix} 0 & q_i \\ c_i & d_i \end{smallmatrix}\right)$, and they provide two precomputing algorithms. When n is prime, one of them produces explicit polylogarithmic factorizations (of length $O(n^3)$) in deterministic time $O(n^3)$. The algorithm recursively finds preimages of $\left(\begin{smallmatrix} 0 & b_1^{2^i} \\ c_i & d_i \end{smallmatrix}\right)$ for some $b_1, c_i, d_i \in \mathbb{F}_{2^n}, 1 \leq i \leq 2n$, from which it deduces preimages of $\left(\begin{smallmatrix} 1 & 0 \\ X+b_1^{2^i} & 1 \end{smallmatrix}\right)$ and then preimages of any matrix of the form $\left(\begin{smallmatrix} 1 & 0 \\ \alpha & 1 \end{smallmatrix}\right)$.

3.7 Further attacks and secure (?) instances

Subgroup attacks are easy to prevent by choosing the group $\mathcal{G}$ carefully. Lifting attacks seem more difficult to thwart since they have become more and more sophisticated. However, simple modifications of the generators have been suggested to counter existing attacks, and the resulting functions remain safe today.

We have seen that the parameters $s_0 = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right), s_1 = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ of Zémor hash function made the lifting strategy particularly easy. Tillich and Zémor suggested to replace them by s_0^2, s_1^2 or by s_0^4, s_1^4 [64]: these parameters are still safe today. After cryptanalysing the LPS and Morgenstern hash function, Tillich and Zémor [63] and Petit et al. [53] also suggested small modifications in the generators that would make the functions safe to their attacks.

From an efficiency point of view, the group $SL(2, \mathbb{F}_{2^n})$ appears as the most interesting one. The Tillich-Zémor hash function was broken by an elaborate lifting attack. Its key ingredient is Mesirov and Sweet's algorithm that is specific to quotients X and $X+1$. Despite some attempts to extend this algorithm [18, 41, 8] (see also the more general surveys [7, 40]), a simple substitution of X by some small power of X in one of the matrices of the Tillich-Zémor function would already make it safe today. The results of Lauder [41] tend to show that the only other generator sets to which this cryptanalysis can be extended should contain more than two *Euclidean algorithm matrices*. More precisely, those sets are

$$\mathcal{S} := \left\{ \begin{pmatrix} t_i & 1 \\ 1 & 0 \end{pmatrix} | t_i \in G \right\}$$

where G is one of the following additive groups:

$$\begin{array}{ll} < X, X^2 + X >, & < X, X^3 + X^2 + X, X^4 + X >, \\ < X+1, X^2 + 1 >, & < X+1, X^3 + 1, X^4 + X^3 + X + 1 >, \\ < X, X^2 + X + 1 >, & < X+1, X^3 + X + 1, X^4 + X^3 + X^2 + 1 >, \\ < 1, X^2 + X + 1, X^3 + 1 >, & < X, X^3 + X^2 + 1, X^4 + X^2 + X + 1 >, \end{array}$$

$$\begin{array}{l} < X+1, X^3 + X^2 + 1, X^5 + X + 1, X^6 + X^5 + X^2 + 1 >, \\ < X, X^3 + 1, X^5 + X^4 + 1, X^6 + X^4 + X + 1 > . \end{array}$$

Of course, we can also obtain other insecure instances by conjugating all the elements of an insecure generator set by the same matrix. Similarly, no change of variable (replacing X by another polynomial in the definition of the generators) can improve the security of a given generator set. However, given our current state of knowledge the generator set

$$\mathcal{S} := \left\{ \begin{pmatrix} t_0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} t_1 & 1 \\ 1 & 0 \end{pmatrix} \right\}$$

is secure for any t_0, t_1 such that $t_0 + t_1 \neq 1$, despite its closeness with the parameters of Tillich-Zémor hash function.

3.8 The end of the story ?

At first sight, the cryptanalysis of Zémor, LPS, Morgenstern and Tillich-Zémor hash function removes the confidence on the security of any Cayley function. However, the balance, representation and factorization problems still appear as potentially hard for most groups $\mathcal{G}$ and sets S. The attacks that we reviewed in this section provide us with some lessons to keep in mind when choosing parameters. In particular, they show the role of the subgroups and the danger of additional structure and symmetric parameters. They also emphasize a strong link with the Euclidean algorithm when $\mathcal{G}$ is $SL(2, .)$. However, the functions that were broken were all very special in a sense: Zémor and Tillich-Zémor use a set of generators with "small" coefficients, and LPS and Morgenstern use the set of all matrices with the same (small) reduced norm. Despite the increasing sophistication of lifting attacks, slight modifications of the original functions seem to resist known attacks. If the balance, representation and factorization problems were solved

for some parameters, the general case is still widely open. We now show that the factorization problem is also widely studied outside the cryptography community, with very limited success so far.

4 Progresses on Babai's conjecture

In the late eighties, Babai made the following conjecture: for every non-Abelian finite simple group $\mathcal{G}$ and every symmetric generating set $\mathcal{S}$ of $\mathcal{G}$, the diameter of the corresponding Cayley graph is smaller than $c_1(\log|\mathcal{G}|)^{c_2}$, where c_1, c_2 are absolute constants [2]. Clearly, solving the factorization problem for some group and generator set amounts to providing a constructive proof of Babai's conjecture for the same parameters. Babai's conjecture has been well-studied for 20 years, with the following results:

- For any group, there exist particular sets of generators for which it is true. In many cases, the factorization problem is solved for these particular sets.
- In the case of groups of Lie type of bounded rank (and in particular for special linear groups of bounded dimension), the conjecture is true for any generator set, but no factorization algorithm is known in general.
- For the cases $SL(2, \mathbb{Z}/p^k\mathbb{Z})$ and $PSL(2, \mathbb{Z}/p^k\mathbb{Z})$, p "small" discussed in Section 3.2, a factorization algorithm is known for any generator set.

We point out two important differences between the factorizations considered in this section and in the previous one. First, the factorizations here may involve negative powers of the generators, making the problem somewhat easier. Second, the instances considered in the previous section were chosen with the hope that the factorization problem would be difficult, whereas here they were specially chosen to make it "particularly easy". In our exposition of these results below, we provide the generators and sometimes a high-level sketch of the proofs that the resulting Cayley graphs have small diameters. However, we refer to the original papers for the often clever and beautiful ideas involved in the details of these proofs.

4.1 Symmetric groups

The alternating groups A_n are better studied through the slightly bigger corresponding symmetric groups S_n. Clearly, Babai's conjecture holds for alternating groups if and only if it holds for symmetric groups. Babai et al. [4] showed how to write any element of S_n as a product of $O(n\log(n))$ elements chosen among a set of 2 well-chosen elements and their inverses. We reproduce their demonstration when n is even.

The group S_n acts naturally on $\mathbb{Z}_{n-1} \cup \{\infty\}$. Let $\alpha_0 : x \mapsto 2x$ and $\alpha_1 : x \mapsto 2x+1$, both permutations fixing ∞. Let also γ_t be the transposition (t, ∞). Then any element of S_n can be written as a word of less than $2n(2\log n + 1)$ generators $\alpha_0, \alpha_1, \gamma_0$ (and their inverses). Indeed, any permutation can be written with less than $2n$ transpositions γ_t. Moreover, γ_t decomposes as $w_t^{-1}\gamma_0 w_t$ where w_t is any permutation fixing ∞ and sending 0 to t. Finally, w_t can be written with less than $\log(n)$ generators α_0 and α_1 using the binary decomposition of t.

Quisquater reduced the number of generators to 2 as follows. Let β_0 be the product of all 3-cycles $(x, 2x, 2x+1)$ where $2^j \leq x < 2^{j+1}$ for all even values j. Let β_1 be the product of all 3-cycles $(x, 2x, 2x+1)$ where $2^j \leq x < 2^{j+1}$ for all odd values j. Let also δ_t be the transposition $(1, t)$. Like before, any element of S_n can be written as a word of less than $2n(2\log n + 1)$ generators $\beta_0, \beta_1, \delta_1$ (and their inverses). However, we have $(\delta_1\beta_1)^3 = \delta_1$ and $(\delta_1\beta_1)^{-2} = \beta_1$, so

we can also write any elements of S_n as a word of less than $6n(2\log n + 1)$ generators $\beta_0, \delta_1\beta_1$ (and their inverses).

4.2 Special linear groups, dimension 2

Projective linear groups are better handled through the corresponding special linear groups. Clearly, Babai's conjecture is true for the first ones if and only if it is also true for the second ones.

For $SL(2, \mathbb{F}_{p^n})$, Babai et al. [4] provide a set of 3 generators that give diameter $O(\log(p^n))$. Their demonstration is as follows. Let

$$x(t) := \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \quad h(b) := \begin{pmatrix} b^{-1} & 0 \\ 0 & b \end{pmatrix} \quad r := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

for $b \neq 0, t \in \mathbb{F}_{p^n}$. Then

$$x(t_1 + t_2) = x(t_1)x(t_2) \text{ and } h(b)^{-1}x(t)h(b) = x(tb^2)$$

for all $b \neq 0, t_1, t_2 \in \mathbb{F}_{p^n}$, and if $ad - bc = 1$ we have

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = x(-c^{-1} + ac^{-1})r^{-1}x(-c)rx(-c^{-1} + dc^{-1}).$$

For the case p odd and $n = 1$, they take $\mathcal{S} = \{x(1), h(1/2)r\}$. For the case p odd and $n \geq 2$, they take $\mathcal{S} = \{x(1), h(1/2)r, h(\theta)\}$ where θ is a generator of $\mathbb{F}_{p^n}$ over $\mathbb{F}_p$. For the case $p = 2$, they take $\mathcal{S} = \{x(1), r, h(\theta)\}$. The proofs are straightforward.

Other generator sets lead to similar or better results. We have seen that the algorithm of Larsen [39] for $\mathcal{G} = SL(2, \mathbb{F}_p)$ provides factorizations of length $O(\log p \log\log p)$ in the matrices $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$. For $\mathcal{G} = SL(2, \mathbb{F}_{2^n})$ with n prime, the preimage algorithm of Petit and Quisquater [55] reduces the set of generators to two elements, the Tillich-Zémor generators. Moreover, the factorizations returned only involve positive powers of the generators. Interestingly, the matrices $x(t)$ also play an important role in this algorithm.

4.3 Special linear groups, dimension > 2

The literature contains a few interesting results for $\mathcal{G} = SL(m, \mathbb{F}_{p^n})$ when $m \geq 3$. The problem does not seem much harder than for $m = 2$ because $SL(2, \mathbb{F}_{p^n})$ is contained as a subgroup of $SL(m, \mathbb{F}_{p^n})$. Moreover, one can take benefit of the extra dimensions to shorten the factorizations as in Riley's algorithm below. Nevertheless, the factorization problem has only been solved for particular generators.

In the case $m \geq 12$, p odd, Kantor [36] has proved that the matrices

$$s_0 = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & 0 & \ddots & \\ & & & \ddots & 1 \\ (-1)^{m-1} & & & & 0 \end{pmatrix} \text{ and } s_1 = \begin{pmatrix} \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right) & & & & \\ & \left(\begin{smallmatrix} 0 & 1/2 \\ 2 & 0 \end{smallmatrix}\right) & & & \\ & & \left(\begin{smallmatrix} 0 & 1/2\theta \\ 2\theta & 0 \end{smallmatrix}\right) & & \\ & & & \left(\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right) & \\ & & & & \left(\begin{smallmatrix} -1 & 1 \\ 0 & 1 \end{smallmatrix}\right) \\ & & & & & I \end{pmatrix}$$

produce a graph with diameter $O(\log|\mathcal{G}|)$ when θ is a generator of $\mathbb{F}_{p^n}^*$. Interestingly, the matrix s_1 is an involution.

Let r_i be the matrix identity except in its entries (i,i) to $(i+1,i+1)$ where it equals $\left(\begin{smallmatrix} & 1 \\ -1 & \end{smallmatrix}\right)$. Let X_{ij} be the set of matrices equal to the identity except in the position (i,j), let D be the set of diagonal matrices and let N be the subgroup generated by D and the matrices r_i. The main steps of the proof are as follows. Any element of X_{34} and X_{56} is a product of respectively $O(\log p)$ and $O(n \log p)$ matrices s_0, s_0^{-1} and s_1. Any element of $< X_{12}, X_{21} >$ can be constructed with $O(n \log p)$ factors. Any element of H can be generated with $O(mn \log p)$ factors. Finally, any element of N can be generated with $O(m^2 n \log p)$ factors, and the result for $\mathcal{G}$ follows. Kantor provided similar results for $m \geq 10$, p odd, and for $m \geq 8$, $p = 2$.

When $n = 1$ and for any $m \geq 3$, Riley [60] has given an algorithm that writes any element as a word of length smaller than $Cm^3 \log p$ in the elements

$$s_0 = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & 0 & \ddots & \\ & & & \ddots & 1 \\ (-1)^{m-1} & & & & 0 \end{pmatrix} \text{ and } s_1 = \begin{pmatrix} 1 & 1 & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix},$$

where C is some absolute constant. For $1 \leq i, j \leq m$, let e_{ij} be the *elementary matrix* that has 1's along the diagonal, 1 in its (i,j) entry, and 0 elsewhere. Any e_{ij} can be written as a product of at most $10m$ generators s_0 and s_1, and the $m^2 - m$ matrices e_{ij} generate $SL(m, \mathbb{F}_p)$. An explicit factorization with respect to the e_{ij} can be recovered with the Euclidean algorithm, like for Zémor hash function in Section 3.4. In pathological cases, this factorization may contain large powers of e_{ij}. Riley found very nice short factorizations for these large powers. When the exponent is a Fibonacci number, a clever combination of e_{ij} matrices and their inverses provides us with the factorization needed. The general case is deduced from Zeckendorf's decomposition of integers as a sum of Fibonacci numbers [66].

Riley's result was improved by Kassabov and Riley [38] to words of length smaller than $O(m^2 \log p)$, which is essentially optimal. Let a *row matrix* be an upper diagonal matrix with ones in the diagonal differing from the identity in only one row. Let a *column matrix* be defined similarly. Kassabov and Riley have showed that any row and column matrix can be generated with at most $m \log p$ elements s_0 and s_1. Moreover, any matrix of $SL(m, \mathbb{F}_p)$ can be written with at most m row matrices, m column matrices and m elementary matrices. Kassabov and Riley also generalized these results to $SL(2, \mathbb{Z}/k\mathbb{Z})$, k integer.

4.4 Other groups

Similar results were obtained for all finite simple non Abelian groups. In particular, there exists a constant C such that any finite simple non-abelian group $\mathcal{G}$ has a set $\mathcal{S}$ of at most four generators such that every element of $\mathcal{G}$ can be written as a product of elements of $\mathcal{S} \cup \mathcal{S}^{-1}$ of length smaller than $C \log |\mathcal{G}|$ [5, 38]. The proof decomposes the group $\mathcal{G}$ into products of a restricted set of elementary subgroups (as for example in [50]) and then treats these elementary cases separately. The most interesting cases are the cases covered in Sections 4.1 and 4.3.

4.5 Non explicit results

All the results on Babai's conjecture that we have described so far are for particular generators making the factorization problem somewhat easier.

In a recent breakthrough, Helffgott [33] showed that Babai's conjecture is true for $SL(2, \mathbb{F}_p)$ with p prime and any generator sets (hence also for $PSL(2, \mathbb{F}_p)$). However, his proof does not

provide explicit factorizations. The arguments are purely combinatorics. Starting from a small set of generators, Helffgott proves that this set must grow "significantly" under multiplication and division by itself, unless it is already a "large" fraction of $SL(2, \mathbb{F}_p)$, from which all the elements can be easily constructed. In the proof, the growth of matrix sets is reduced to the growth of the sets containing their traces, and these sets are studied through the sum-product estimates of Bourgain-Katz-Tao [12]. Helffgott's results were extended to the groups $SL(2, \mathbb{F}_{p^n})$ and to directed graphs by Dinai [25], to the groups $SL(3, \mathbb{F}_p)$ by Helffgott himself [32], and to all groups of Lie type with bounded rank independently by Pyber and Szabó [57] and by Breuillard et al. [15]. For groups of Lie type with unbounded rank new ideas will be required [57], and unfortunately none of the previous results provides any explicit factorization algorithm. It is also worth noticing that under some wide conditions on the generator sets, the Cayley graphs of $SL(d, \mathbb{F}_p)$ do provide interesting families of expander graphs [11, 15, 57, 13].

4.6 The end of the story ?

The recent interest of the mathematics community for Babai's conjecture and the progresses made after Helfgott's big contribution [33] lead to some hope that the conjecture may be proved in a not too far horizon. However, the techniques that are currently used (involving tools from combinatorics and representation theory) have not provided us with explicit factorization algorithms. *Constructive* proofs of Babai's conjecture are known in some cases but only for particular sets of generators. A simple look at Sections 4.1, 4.2 and 4.3 of this paper suffices to see that these sets are very far from generic. After more than 20 years of active research on Babai's conjecture, a new breakthrough is probably needed in order to solve the factorization problem for arbitrary generator sets.

5 A new cryptographic challenge

We have seen in the previous sections that the balance, representation and factorization problems are potentially hard problems for generic parameters. However, cryptographic applications require parameters that are not only secure but that also lead to efficient implementations. Matrix groups over finite fields are appealing since the group operation can be implemented with a few additions and multiplications in the field. Moreover, they are among the most studied and best understood groups, giving more confidence on security.

Special linear groups and projective special linear groups are a bit more appealing than general linear groups. The reason is that solving the factorization problem in the quotients $GL(m, K)/PSL(m, K)$ or $GL(m, K)/SL(m, K)$ is essentially equivalent to solving a discrete logarithm problem in K^* or K^{*2}. Choosing K sufficiently large to make the discrete logarithm difficult would render Cayley hash functions too inefficient (a few multiplications per bit of message, more than discrete logarithm-based hash functions). The security for GL, SL and PSL is essentially equivalent for smaller fields, but using general linear groups would give the false feeling that the security is larger. The best choice for m seems to be $m = 2$. As mentioned in Section 4.3, taking $m > 2$ will not necessarily increase the security and might even decrease it a little bit. Besides, taking $m = 2$ is clearly better from an efficiency point of view.

The groups $SL(2, \mathbb{F}_{2^n})$ are more interesting than the groups $SL(2, \mathbb{F}_p)$: the arithmetic operations are much more efficient in $\mathbb{F}_{2^n}$ than in $\mathbb{F}_p$, especially in hardware. A few additional restrictions must be set on n. Clearly, n must be large enough such that birthday attacks (Section 3.1) are impossible. It must also be prime in order to avoid the subgroup attacks of Steinwandt et al. [61] (Section 3.2). Finally, it seems wise to require both $2^n + 1$ and $2^n - 1$ to be either primes or small multiples of primes in order to prevent other kinds of subgroup attacks.

The parameters $n \in \{127, 157, 223, 251, 383, 509\}$ seem satisfying for a security of roughly $n/2$ bits.

Having fixed a family of groups, we now turn to the generators. The parameters chosen by Tillich and Zémor for their function (Section 2.1) are particularly appealing from an efficiency point of view, but unfortunately they are vulnerable to the attacks of Section 3.6. According to Section 3.7, the generator set $\mathcal{S} := \left\{ \left(\begin{smallmatrix} X^3 & 1 \\ 1 & 0 \end{smallmatrix} \right), \left(\begin{smallmatrix} X+1 & 1 \\ 1 & 0 \end{smallmatrix} \right) \right\}$ seems secure for any t_0, t_1 such that $t_0 + t_1 \neq 1$. These generators have the advantage of requiring only one multiplication and a few additions per bit. To reduce even further the cost of the group operation to a few additions, we suggest taking $t_0 = X^3$ and $t_1 = X + 1$. This gives us the following challenge:

Challenge 1 *Solve the balance, representation or factorization problem for* $G := SL(2, \mathbb{F}_{2^n})$ *and* $\mathcal{S} := \left\{ \left(\begin{smallmatrix} X^3 & 1 \\ 1 & 0 \end{smallmatrix} \right), \left(\begin{smallmatrix} X+1 & 1 \\ 1 & 0 \end{smallmatrix} \right) \right\}$.

6 Conclusion

Cayley hash functions are very appealing to cryptography. They have a simple and elegant design, a nice mathematical structure and a natural parallelism. However, their main security properties rely on the hardness of mathematical problems that are non standard to cryptography. The recent cryptanalysis of all Cayley hash function proposals (Zémor, Tillich-Zémor, LPS, Morgenstern) has cast doubts on the hardness of these mathematical problems in the cryptography community.

In this paper, we have argued that these doubts are unjustified, or at least premature. Indeed, we have shown that

- The four Cayley hash functions that were broken had parameters that seem particularly weak *a posteriori*. The cryptanalysis techniques used against these functions cannot be easily applied to other parameters. In particular, small changes in the four functions make them immune against existing attacks.

- The mathematical problems supporting the security properties of Cayley hash functions have a rich history in mathematics, if not in cryptography. They originate at least to the work of Babai in the late eighties, and in particular to its conjecture on the diameter of the Cayley graphs of finite non-Abelian simple groups.

- The research on these problems has been very active and it has involved distinguished mathematicians like Babai, Bourgain, Gamburd, Green, Helfgott, Kantor, Lubotzky, Tao,... Nevertheless, very few instances have been solved today after 20 years.

The Rubik's cube is a notoriously hard mechanical puzzle... for humans. The factorization problem in non-Abelian groups is its natural mathematical generalization. Our survey demonstrates that this problem is potentially hard from a cryptographic point of view. It is also interesting in its own right, intersecting and connecting group theory, graph theory, number theory, combinatorics, the Euclidean algorithm,... Any new result on secure and unsecure Cayley hash function instances will be beneficial not only to cryptography but also to the numerous applications of Cayley graphs and expander graphs in mathematics and computer science. From a purely cryptographic point of view, the challenge is to find a set of parameters that leads not only to hard problems but also to reasonably efficient implementations. We hope that this paper will revive the interest for Cayley hash functions, and will be useful to those willing to study the hardness of the subjacent mathematical problems.

Acknowledgement *We express our gratitude to François Koeune and Sylvie Baudine for their help in improving this paper. We also thank Gilles Zémor and Kristin Lauter for pointing us important references, as well as for interesting and fruitful collaboration and discussions. The first author is supported by a postdoctoral grant of the Belgian National Science Foundation (FRS-FNRS). He is also grateful to the organizers of the* SHA3 ECRYPTII workshop*, to the* Institut Mathématiques de Bordeaux*, to* Microsoft Research, Seattle *and to the* Centre de Recherches Mathématiques de Montréal*, for giving him the opportunity to present his work.*

References

[1] `http://csrc.nist.gov/groups/ST/hash/documents/SHA-3_FR_Notice_Nov02_2007%20-%20more%20readable%20version.pdf`.

[2] L. Babai and Ákos Seress. On the diameter of permutation groups. *Eur. J. Comb.*, 13(4):231–243, 1992.

[3] L. Babai and T. P. Hayes. Near-independence of permutations and an almost sure polynomial bound on the diameter of the symmetric group. In *SODA*, pages 1057–1066. SIAM, 2005.

[4] L. Babai, G. Hetyei, W. M. Kantor, A. Lubotzky, and Á. Seress. On the diameter of finite groups. In *FOCS*, volume II, pages 857–865. IEEE, 1990.

[5] L. Babai, W. Kantor, and A. Lubotzky. Small-diameter Cayley graphs for finite simple groups. *European Journal of Combinatorics*, 10:507–552, 1989.

[6] M. Bellare and D. Micciancio. A new paradigm for collision-free hashing: Incrementality at reduced cost. In *EUROCRYPT*, pages 163–192, 1997.

[7] V. Berthé and H. Nakada. On continued fraction expansions in positive characteristic: Equivalence relations and some metric properties. *Expositiones Mathematicae*, 18:257–284, 2000.

[8] S. R. Blackburn. Orthogonal sequences of polynomials over arbitrary fields. *Journal of Number Theory*, 68(1):99 – 111, 1998.

[9] A. Boldyreva, D. Cash, M. Fischlin, and B. Warinschi. Foundations of non-malleable hash and one-way functions. Cryptology ePrint Archive, Report 2009/065, 2009. `http://eprint.iacr.org/`.

[10] J. Bosset. Contre les risques d'altération, un système de certification des informations. *Informatique*, 107, 1977.

[11] J. Bourgain and A. Gamburd. Uniform expansion bounds for cayley graphs of $sl_2(\mathbb{F}_p)$. *Annals of Mathematics. Second Series*, 167(2):625–642, 2008.

[12] J. Bourgain, N. Katz, and T. Tao. A sum-product estimate in finite fields, and applications. *Geometric and Functional Analysis*, 14:27, 2004.

[13] J. Bourgain and P. P. Varjù. Expansion in $sl_d(z/qz)$, q arbitrary. `http://arxiv4.library.cornell.edu/abs/1006.3365`, June 2010. Bourgain2010.

[14] G. Brassard, editor. *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*. Springer, 1990.

[15] E. Breuillard, B. Green, and T. Tao. Approximate subgroups of linear groups. arXiv:1005.1881v1, May 2010.

[16] E. Breuillard, B. Green, and T. Tao. Suzuki groups as expanders. `http://arxiv.org/abs/1005.0782v1`, May 2010. BREUILLARD2010.

[17] P. Camion. Can a fast signature scheme without secret key be secure? In *AAECC*, volume 228 of *Springer Verlag Lecture Notes in Computer Science*, pages 187–196, 1987.

[18] G. Carter. Some conditions on the linear complexity profiles of certain binary sequences. In *EUROCRYPT*, pages 691–695, 1989.

[19] D. Charles, E. Goren, and K. Lauter. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.

[20] P. Colmez. Le Rubik's cube, groupe de poche.

[21] S. Contini, A. K. Lenstra, and R. Steinfeld. VSH, an efficient and provable collision-resistant hash function. In S. Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 165–182. Springer, 2006.

[22] I. Damgård. A design principle for hash functions. In Brassard [14], pages 416–427.

[23] G. de Meulenaer, C. Petit, and J.-J. Quisquater. Hardware implementations of a variant of Zémor-Tillich hash function: Can a provably secure hash function be very efficient? Preprint, 2009.

[24] O. Dinai. Poly-log diameter bounds for some families of finite groups. *Proc. Amer. Math. Soc.*, 134:3137–3142, 2006.

[25] O. Dinai. *Expansion properties of finite simple groups*. PhD thesis, The Hebrew University, 2009.

[26] J. D. Dixon. The probability of generating the symmetric group. *Mathematische Zeitschrift*, 110 (3):199–205, 1969.

[27] S. Even and O. Goldreich. The minimum-length generator sequence problem is NP-hard. *J. Algorithms*, 2(3):311–313, 1981.

[28] A. Gamburd and M. Shahshahani. Uniform diameter bounds for some families of cayley graphs. *International mathematics research notices*, 71:3813–3824, 2004.

[29] D. Giry and P. Bulens. keylength.com.

[30] O. Goldreich. *Fundations of Cryptography, Volume II Basic Applications*. Cambridge University Press, 2004.

[31] M. Grassl, I. Ilic, S. Magliveras, and R. Steinwandt. Cryptanalysis of the Tillich-Zémor hash function. Cryptology ePrint Archive, Report 2009/376, 2009. `http://eprint.iacr.org/`.

[32] H. Helfgott. Growth and generation in $SL_3(Z/pZ)$. *Journal of the European Mathematical Society (JEMS)*, to appear, 2010.

[33] H. A. Helfgott. Growth and generation in $SL_2(Z/pZ)$, 2005.

[34] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–561, 2006.

[35] M. R. Jerrum. The complexity of finding minimum-length generator sequences. *Theor. Comput. Sci.*, 36(2-3):265–289, 1985.

[36] W. M. Kantor. Some large trivalent graphs having small diameters. *Discrete Applied Mathematics*, 37/38:353–357, 1992.

[37] W. M. Kantor and A. Lubotzky. The probability of generating a finite classical group. *Geometriae Dedicata*, 36:67–87, 1990.

[38] M. Kassabov and T. Riley. Diameters of cayley graphs of chevalley groups. *European Journal of Combinatorics*, 28(3):791–800, 2005.

[39] M. Larsen. Navigating the Cayley graph of $SL_2(\mathbb{F}_p)$. *International Mathematics Research Notices*, 27:1465–1471, 2003.

[40] A. Lasjaunias. A survey of diophantine approximationin fields of power series. *Monatshefte für Mathematik*, 130(3):211–229, 2000.

[41] A. Lauder. Continued fractions of laurent series with partial quotients from a given set. Acta Arithmetica XC.3, 1999.

[42] M. W. Liebeck and A. Shalev. The probability of generating a finite simple group. *Geometriae dedicata*, 56:103–113, 1995.

[43] A. Lubotzky. *Discrete groups, expanding graphs and invariant measures*. Birkhaüser Verlag, 1994.

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

[45] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In K. Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 54–72. Springer, 2008.

[46] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. The Deep Space Network Progress Report, DSN PR 42-44, January and February 1978. 114-116.

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

[48] J. P. Mesirov and M. M. Sweet. Continued fraction expansions of rational expressions with irreducible denominators in characteristic 2. *Journal of Number Theory*, 27:144–148, 1987.

[49] M. Morgenstern. Existence and explicit construction of $q + 1$ regular Ramanujan graphs for every prime power q. *Journal of Combinatorial Theory*, B 62:44–62, 1994.

[50] N. Nikolov. A product decomosition for the classical quasisimple groups. arXiv:math/0510173v1, October 2005.

[51] J. Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *EUROCRYPT*, pages 33–48, 1996.

[52] C. Petit. *On graph-based cryptographic hash functions.* PhD thesis, Université catholique de Louvain, 2009.

[53] C. Petit, K. Lauter, and J.-J. Quisquater. Full cryptanalysis of LPS and Morgenstern hash functions. In R. Ostrovsky, R. D. Prisco, and I. Visconti, editors, *SCN*, volume 5229 of *Lecture Notes in Computer Science*, pages 263–277. Springer, 2008.

[54] C. Petit, K. E. Lauter, and J.-J. Quisquater. Cayley hashes: A class of efficient graph-based hash functions. Available at `http://www.dice.ucl.ac.be/~petit/`, 2007.

[55] C. Petit and J.-J. Quisquater. Preimages for the Tillich-Zémor hash function. To appear in the proceedings of SAC2010 (in press), 2010.

[56] C. Petit, J.-J. Quisquater, J.-P. Tillich, and G. Zémor. Hard and easy components of collision search in the Zémor-Tillich hash function: New attacks and reduced variants with equivalent security. In M. Fischlin, editor, *CT-RSA*, volume 5473 of *Lecture Notes in Computer Science*, pages 182–194. Springer, 2009.

[57] L. Pyber and E. Szabó. Growth in finite simple groups of Lie type. arXiv:1001.4556v1, Jan 2010.

[58] J.-J. Quisquater and J.-P. Delescaille. How easy is collision search. new results and applications to DES. In Brassard [14], pages 408–413.

[59] O. Regev. Lattice-based cryptography. In C. Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 131–141. Springer, 2006.

[60] T. R. Riley. Navigating in the cayley graphs of $sl_n(\mathbb{Z})$ and $sl_n(\mathbb{F}_p)$. *Geometriae Dedicata*, 113/1:215–229, 2005.

[61] R. Steinwandt, M. Grassl, W. Geiselmann, and T. Beth. Weaknesses in the $SL_2(\mathbb{F}_{2^n})$ hashing scheme. In *Proceedings of Advances in Cryptology - CRYPTO 2000: 20th Annual International Cryptology Conference*, 2000.

[62] J.-P. Tillich and G. Zémor. Hashing with SL_2. In Y. Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 40–49. Springer, 1994.

[63] J.-P. Tillich and G. Zémor. Collisions for the LPS expander graph hash function. In N. P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 254–269. Springer, 2008.

[64] J.-P. Tillich and G. Zémor. Group-theoretic hash functions. In *Proceedings of the First French-Israeli Workshop on Algebraic Coding*, pages 90–110, London, UK, 1993. Springer-Verlag.

[65] G. Yuval. How to swindle Rabin. *Cryptologia*, 3:187–189, 1979.

[66] E. Zeckendorf. Représentation des nombres naturel par une somme de nombres fibonacci ou de nombres de lucas. *Bulletin de la Société Royale des Sciences à Liège*, 41:179–182, 1972.

[67] G. Zémor. Hash functions and graphs with large girths. In D. W. Davies, editor, *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 508–511. Springer, 1991.