Rubik's for Cryptographers

Christophe Petit and Jean-Jacques Quisquater

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 have survived thorough cryptanalysis, the most established ones being the integer factorization problem and the discrete logarithm problems on finite fields and elliptic curves. Other problems have been suggested, related, for example, to hyperelliptic curves, lattices [42], error-correcting codes [31], or multivariate polynomial equations [35] (the so-called postquantum cryptographic algorithms). They are currently less trusted than the three previous ones, but they might join or replace them in the future.

In this paper we discuss three alternative computational problems: namely the *balance, representation, and factorization problems in finite non-Abelian groups*. Interestingly, these problems can be seen as generalizations of the Rubik's Cube. The famous 3D puzzle is notoriously "hard" [12], but of course not in a cryptographic sense. Computer programs solve it instantaneously, and even human champions need less than ten seconds. Nevertheless, the "extensions" considered in this paper were proposed as the core computational problems underlying the security of *Cayley hash functions*, an elegant construction of a very important cryptographic primitive.

For the cryptographic applications to be secure, the balance, representation, and factorization problems must be computationally hard. These problems are of course easy for the Rubik's Cube. They are also easy in a few other particular cases, but they may still be hard in general. In fact, they are strongly connected to famous problems in group theory and can be seen as algorithmic versions of a twenty-year-old conjecture of Babai on the diameters of Cayley graphs. Although the conjecture is now proved for all parameters of interest in cryptography, many of the proofs are nonconstructive, hence useless, to "break" the functions.

In the last twenty years, the cryptography community (for Cayley hash functions) and the mathematics community (for Babai's conjecture) have been working independently on very similar problems. The goal of this paper is to bridge the results obtained by the two communities, with the cryptographic application in mind. In particular, we review known results coming from both sides, we provide some general attacks and design principles for the Cayley hash function construction, and finally we propose some parameters that can be considered as "safe" from our current knowledge of these problems.

Notation. In this paper, p will always be a prime and n a positive integer. We write $\mathbb{F}_q$ for the finite field with q elements. We identify the finite field $\mathbb{F}_{p^n}$ with $\mathbb{F}_p[X]/(q(X))$, where q is an irreducible polynomial over $\mathbb{F}_p$. If K is a finite field and m is a positive integer, we write $SL(m,K)$ for the special linear group of degree m over K, in other words, the group of m-by-m matrices over K with determinant 1. We write $PSL(m,K)$ for the projective special linear group of degree m over K, which is the quotient of $SL(m,K)$ by the set $\{\lambda I, \lambda \in K\}$. Finally, we write S_n for the group of permutations on n elements.

Outline. The remainder of this paper is organized as follows. In the first section we recall the Cayley

Christophe Petit is a research fellow of the Belgian Fund for Scientific Research (F.R.S.-FNRS) at Université catholique de Louvain (UCL). His email address is christophe.petit@uclouvain.be.

Jean-Jacques Quisquater is with the UCL Crypto Group. His email address is jjq@uclouvain.be.

DOI: http://dx.doi.org/10.1090/noti1001

hash function construction and its main advantages over other cryptographic hash functions. In the second section we define the balance, representation, and factorization problems and show their connection with the security of Cayley hash functions, Babai's conjecture, and the Rubik's Cube. In the third section we describe general methods to solve these problems in particular cases, and we provide particular parameters that appear as a cryptographic challenge today. We conclude the paper in the fourth and final section.

Cayley Hash Functions

In this section we first recall the definition and the main properties of cryptographic hash functions, a very important cryptographic primitive. We then recall the definition of Cayley hash functions, an interesting family of hash functions based on non-Abelian groups.

Cryptographic Hash Functions

Hash functions are a fundamental building block in cryptographic protocols, in particular in digital signature schemes and message authentication codes. A *hash function* is a function that takes as inputs bitstrings of arbitrary length and then returns bitstrings of fixed finite small length. Additionally, the 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 [32]:

- **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 $m' \in \{0,1\}^*$, $m' \neq m$, 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 $h = H(m)$.

In cryptographic protocols, the output of a hash function is often used as a small digest of its input aimed to represent this input uniquely. Since hash functions are not injective, this "uniqueness" is of course only "computational". The words "computationally hard" can be understood in two different ways. From a theoretical point of view, it means that no probabilistic algorithm that runs in polynomial time 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 [17]. On the other hand, from a practical point of view, "computationally hard" means that no big cluster of computers can perform a given task. A computational complexity of 2^{80} operations is currently considered out of reach [16].

Classical hash functions like the NIST's standard hash algorithms [34] mix pieces of the message again and again until the result looks sufficiently random. The design of these functions may somehow look like a sack of nodes that may discourage its study outside the cryptography community. In contrast, Cayley hash functions have a clear, simple, and elegant mathematical design.

A Construction Based on Cayley Graphs

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

$$h = s_{m_1} s_{m_2} \ldots s_{m_N}$$

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

The computation of a hash value amounts to a walk in the corresponding Cayley graph. To make the computation more efficient, the four Cayley hash functions proposed in the literature use the matrix groups $SL(2, K)$ or $PSL(2, K)$, where K is either $\mathbb{F}_p$ or $\mathbb{F}_{2^n}$ [48], [45], [9], [38]. Cayley hash functions are rather slow for most parameters, but in some contexts they perform better than the standard SHA-1 [11]. One big advantage of Cayley hash functions over classical hash functions is that their computation 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.

As we will see below, the overall security of Cayley hash functions (their collision, second preimage, and preimage resistance) may depend on the group and the generators used. The group structure also induces some properties that may be undesirable in some applications. For example, given the hash value of m and m', it is possible to compute the hash value of $m||m'$. However, additional design components can solve this problem [36].

Balance, Factorization, and Representation Problems

In this section we introduce the mathematical problems at the core of the security of Cayley hash functions. We then link them to famous problems in group theory and with the Rubik's Cube.

Security of Cayley Hash Functions

The collision, second preimage, and preimage properties of Cayley hash functions can easily be translated into group-theoretic problems.

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

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

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 "computationally 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 [15], [22]. The factorization problem was described by Lubotzky as a "noncommutative analog of the discrete logarithm problem" [30, 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, and it is preimage resistant if and only if the corresponding factorization problem is hard. Interestingly, the balance, representation, and factorization problems can be seen as constructive versions of old questions and some long-standing open problems in group theory.

Link with Babai's Conjecture

The first question (coming back to Dixon [14]) is the probability that a random set of elements in a finite non-Abelian group generates the whole group. This probability is arbitrarily close to 1 for sets of two elements if the groups are large enough [25], [29]. Given $\mathcal{G}$ and S, it is then natural to ask for the maximal length of all minimal representations of the elements of $\mathcal{G}$ as products of the elements of S. In other words, we are interested in the *diameter* of the corresponding Cayley graph.

This diameter can easily be lower bounded by $\log_{|S|} |\mathcal{G}|$ using the pigeon-hole principle, but it is not known whether the bound is tight in general. A large source of graphs with logarithmic diameter is provided by *expander graphs* [21]. Roughly speaking, an expander graph is a regular graph such that any set of its vertices has a comparatively large set of neighbors. An intense research effort in the last ten years has 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 [8]. Expander graphs are very important for computer science, with a wide range of applications.

In general, Cayley graphs are expected to have *polylogarithmic* rather than logarithmic diameters. More precisely, a conjecture made by Babai in the early 1990s states that the diameter of any undirected Cayley graph of a non-Abelian simple group is bounded by a polynomial function in the size of the group [3], [20]. Directed Cayley graphs can be included as well in the conjecture without strengthening it [2].

Interestingly, the factorization problem discussed in this paper can be seen as providing a *constructive* proof of Babai's conjecture. By definition, "small" factorizations exist in Cayley graphs with logarithmic or polylogarithmic diameter, but this does not imply that they can be efficiently computed with a polynomial time algorithm. In this sense, the cryptanalyst's task seems harder than just proving Babai's conjecture. On the other hand, cryptanalysts may not need a constructive proof for *all* parameter sets but only for a large fraction of parameters in some relevant subfamilies.

A "Toy" Example: The Rubik's Cube

We now express 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 six elementary rotations of the faces. The Rubik's group has order $|\mathcal{G}_R| = \frac{1}{12} 12!8!3^8 2^{12}$ and 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 semidirect group products [10]. Solving the Rubik's Cube amounts to solving the factorization problem for the group $\mathcal{G}_R$ and the set S containing the six rotations of the faces.

Rubik's for Cryptographers

After twenty years of research on Babai's conjecture, particularly intense after a breakthrough result of Helfgott in 2005 [20], Babai's conjecture has been proved for any generator set of any group of Lie type with a bounded rank [41], [7]. Helfgott and Seress [19] have also obtained a slightly weaker *quasipolynomial* bound for the diameter of permutation groups. However, most of these proofs are nonconstructive. Constructive proofs are available only for the four parameter sets that were proposed for Cayley hash functions, for permutation groups, and for a few "well-chosen" generator sets in other groups. In this section, we briefly survey these results and we conclude with a cryptographic challenge.

General Techniques

The mathematical structure of Cayley hash functions makes them more vulnerable to attacks that would exploit this structure. Two main attack techniques and design principles can be inferred from our experience.

A first important category of attacks is *subgroup attacks*. If $\mathcal{G}$ has a subgroup tower decomposition $\mathcal{G} = \mathcal{G}_0 \supset \mathcal{G}_1 \supset \mathcal{G}_2 \supset \cdots \supset \mathcal{G}_N = \{1\}$ and if $|G_i|/|G_{i+1}|$ is "small" for all i then, given any set $S = \{s_1, \ldots, s_k\}$, 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 $S^{(1)} = \{s_1^{(1)}, \ldots, s_{k'}^{(1)}\}$ 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 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. These attacks can be extended to solve the factorization problem as well, and the condition that all the quotients $|G_i|/|G_{i+1}|$ are "small" can sometimes be replaced by a weaker one.

A simple example of a subgroup attack is the "level-by-level" resolution method for the Rubik's Cube: 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 can be constructed against the Rubik's Cube. A similar idea was used by Dinai [13] for the group $SL(2, \mathbb{Z}/p^k\mathbb{Z})$ which has the subgroup tower $SL(2, \mathbb{Z}/p^k\mathbb{Z}) \supset SL(2, \mathbb{Z}/p^{k-1}\mathbb{Z}) \supset \cdots \supset SL(2, \mathbb{Z}/p\mathbb{Z}) \supset \{I\}$. If p is "small", then the factorization problem can be solved for any generator set of this group. Finally, two subgroup attacks against a Cayley hash function proposed by Tillich and Zémor [45] were provided in [44], [40]. In matrix groups, the orthogonal, triangular, or diagonal subgroups can typically be exploited.

The simplest subgroup attacks can be prevented by choosing the group $\mathcal{G}$ carefully. The minimal requirement is that the cardinalality of $\mathcal{G}$ has a "large" factor, but additional requirements may be needed depending on the group family.

A second very important category of attacks is that of *lifting attacks*. To describe their principle, let us suppose that $\mathcal{G}$ is $SL(2, \mathbb{F}_p)$. There is a natural homomorphism from $SL(2, \mathbb{Z})$ to $SL(2, \mathbb{F}_p)$ given by the "reduction modulo p". A lifting attack for $SL(2, \mathbb{F}_p)$ will "lift" the generators on $SL(2, \mathbb{Z})$ and then try to "lift" the element to be factored on the subgroup of $SL(2, \mathbb{Z})$ generated by the lifts of the generators. Factoring elements in $SL(2, \mathbb{Z})$ is usually easier (when a factorization exists) than in $SL(2, \mathbb{F}_p)$. This is mainly because factorizations are *unique* in infinite groups if the generators are properly chosen, and the elements composing a factorization can then often be recovered one by one. Once a factorization over $SL(2, \mathbb{Z})$ has been obtained, a "reduction modulo p" provides a factorization over $SL(2, \mathbb{F}_p)$. Therefore the hardest part of a lifting attack is typically the lifting part itself.

This part seems "hard" in general, but it is easier when the generators have a special structure. In the first Cayley hash function proposed by Zémor [48] with $\mathcal{G} = SL(2, \mathbb{F}_p)$, the generator set was chosen as $S = \{\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)\}$ to improve the efficiency of the function. However, these two matrices are so "small" that they generate the whole group $SL(2, \mathbb{Z}_+)$. Factorizations in this infinite group could be easily computed with the help of Euclid's algorithm [47]. Interestingly, all the lifting attacks against Cayley hash functions have used a connection between 2-by-2 matrices of particular forms and the Euclidean algorithm. In the cryptanalysis of LPS and Morgenstern hash functions [9], [38] using the groups $PSL(2, \mathbb{F}_p)$ and $PSL(2, \mathbb{F}_{2^n})$, the natural lifts of the generators did not generate a dense subgroup of their infinite counterparts.

However the generators (chosen in both cases to achieve an optimal output distribution) had natural symmetries that allowed one to reduce the lifting part to the resolution of a quadratic diophantine equation [46], [37], [36]. Finally, the cryptanalysis of the Tillich-Zémor hash function [45] consisted of lifting and factoring special elements of $SL(2, \mathbb{F}_{2^n})$ on $SL(2, \mathbb{F}_2[X])$ and then of combining these elements in $SL(2, \mathbb{F}_{2^n})$. The lifting part crucially needed the fact that the Tillich-Zémor generators had very "small" coefficients in $SL(2, \mathbb{F}_{2^n})$ [18], [39].

Lifting attacks are harder to prevent than subgroup attacks, since they have become more and more sophisticated over the years. However, the attacks have always required special properties of the generators (either "small" or "symmetric") to perform the lifting part itself. In all cases, simple modifications of the generators were proposed to counter the attacks, and these variants have remained unbroken so far.

Particular "Broken" Instances

While cryptographers have been trying to break Cayley hash functions, mathematicians have studied the diameter of Cayley graphs and tried to prove Babai's conjecture. The conjecture has been proved for almost all sets of generators in symmetric and alternating groups by Babai and Hayes [5]. The proof is constructive, hence permutation groups must be avoided in cryptographic applications (see also [23]). For the group $SL(2, \mathbb{F}_{p^n})$, Babai et al. [4] provided a set of 3 generators such that the diameter of the corresponding Cayley graph is $O(\log(p^n))$, together with a constructive algorithm. Kantor [24], Riley [43], and Kassabov and Riley [26] provided factorization algorithms with shorter and shorter lengths for well-chosen couples of elements in $SL(m, \mathbb{F}_p)$ and $SL(m, \mathbb{F}_{p^n})$, $m \geq 3$. More generally, there exists a constant C such that any finite simple non-Abelian group $\mathcal{G}$ has a set S of at most four generators such that every element of $\mathcal{G}$ can be written as a product of elements of $S \cup S^{-1}$ of length smaller than $C \log |\mathcal{G}|$ [1], [26].

A Cryptographic Challenge

Although Cayley hash functions can be broken for permutation groups and for a few specific parameters in other groups, determining their security remains an open problem in general. In particular, essentially nothing is known for "generic" parameters of special linear groups.

Challenge 1. *Solve the balance, representation, or factorization problem for* $G := SL(2, \mathbb{F}_{2^n})$ *and* $S := \left\{ \begin{pmatrix} t_1 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} t_2 & 1 \\ 1 & 0 \end{pmatrix} \right\}$, *where* $t_1 := X^3$ *and* $t_2 := X + 1$.

The Tillich-Zémor parameters were equivalent to a similar set with $t_1 := X$ and $t_2 := X + 1$ [47], [18]. Although our set also contains "small" matrices for efficiency reasons, the core ingredient in the cryptanalysis of the Tillich-Zémor hash function (an algorithm of Mesirov and Sweet related to the Euclidean algorithm [33]) does not seem to generalize to our parameters [18], [39] (in particular in the light of Lauder's results on generalizations of Mesirov-Sweet's algorithm [28]).

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 nonstandard 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 our opinion, these doubts are unjustified or at least premature. 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 in 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 has involved distinguished mathematicians such as Babai, Bourgain, Gamburd, Green, Helfgott, Kantor, Lubotzky, Tao,…. Nevertheless, with the exception of permutation groups, very few instances have been solved today after twenty 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. If it turns out to be "hard" enough, this problem could be very useful in cryptography. 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.

References

1. László Babai, William M. Kantor, and Alexander Lubotzky, Small-diameter Cayley graphs for finite simple groups, *European J. Combin.* **10** (1989), 507–552.
2. László Babai, On the Diameter of Eulerian Orientations of Graphs, *SODA*, ACM Press, 2006, pp. 822–831.
3. László Babai and Ákos Seress, On the diameter of permutation groups, *European J. Combin.* **13** (1992), no. 4, 231–243.
4. László Babai, Gábor Hetyei, William M. Kantor, Alexander Lubotzky, and Ákos Seress, On the diameter of finite groups, *FOCS*, vol. II, IEEE, 1990, pp. 857–865.
5. László Babai and Thomas P. Hayes, Near-independence of permutations and an almost sure polynomial bound on the diameter of the symmetric group, *SODA*, 2005, pp. 1057–1066.
6. Mihir Bellare and Daniele Micciancio, A new paradigm for collision-free hashing: Incrementality at reduced cost, *EUROCRYPT* (Walter Fumy, ed.), Lecture Notes in Computer Science, vol. 1233, Springer, 1997, pp. 163–192.
7. Emmanuel Breuillard, Ben Green, and Terence Tao, *Approximate subgroups of linear groups*, arXiv: 1005.1881v1, May 2010.
8. ———, *Suzuki groups as expanders*, http://arxiv.org/abs/1005.0782v1, May 2010.
9. Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren, Cryptographic hash functions from expander graphs, *J. Cryptology* **22** (2009), no. 1, 93–113.
10. Pierre Colmez, *Le Rubik's cube, groupe de poche*, http://www.math.ens.fr/culturemath/maths/articles/Colmez/rubiks-cube-groupe-de-poche.pdf.
11. Giacomo de Meulenaer, Christophe Petit, and Jean-Jacques Quisquater, *Hardware implementations of a variant of the Zémor-Tillich hash function: Can a provably secure hash function be very efficient?*, Cryptology ePrint Archive, Report 2009/229, 2009, http://http://eprint.iacr.org/.
12. Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Anna Lubiw, and Andrew Winslow, Algorithms for solving Rubik's Cubes, *ESA* (Camil Demetrescu and Magnús M. Halldórsson, eds.), Lecture Notes in Computer Science, vol. 6942, Springer, 2011, pp. 689–700.
13. Oren Dinai, Poly-log diameter bounds for some families of finite groups, *Proc. Amer. Math. Soc.* **134** (2006), 3137–3142.
14. John D. Dixon, The probability of generating the symmetric group, *Mathematische Zeitschrift* **110** (3) (1969), 199–205.
15. Shimon Even and Oded Goldreich, The minimum-length generator sequence problem is NP-hard, *J. Algorithms* **2** (1981), no. 3, 311–313.
16. Damien Giry and Philippe Bulens, http://www.keylength.com.
17. Oded Goldreich, *Foundations of Cryptography, Volume II: Basic Applications*, Cambridge University Press, 2004.
18. Markus Grassl, Ivana Ilic, Spyros S. Magliveras, and Rainer Steinwandt, Cryptanalysis of the Tillich-Zémor hash function, *J. Cryptology* **24** (2011), no. 1, 148–156.
19. Harald Helfgott and Akos Seress, *On the diameter of permutation groups*, http://arxiv.org/abs/1109.3550, 2011.
20. Harald Andrés Helfgott, Growth and generation in $SL_2(Z/pZ)$, *Ann. of Math.* **(2) 167** (2) (2008), 601–623.
21. Shlomo Hoory, Nathan Linial, and Avi Wigderson, Expander graphs and their applications, *Bull. Amer. Math. Soc. (N.S.)* **43** (2006), 439–561.
22. Mark R. Jerrum, The complexity of finding minimum-length generator sequences, *Theor. Comput. Sci.* **36** (1985), no. 2-3, 265–289.
23. Arkadius Kalka, Mina Teicher and Boaz Tsaban, *Short expressions of permutations as products and cryptanalysis of the Algebraic Eraser*, Advances in Applied Mathematics **49**, (2012), 57–76.
24. William M. Kantor, Some large trivalent graphs having small diameters, *Discrete Appl. Math.* **37/38** (1992), 353–357.
25. William M. Kantor and Alexander Lubotzky, The probability of generating a finite classical group, *Geom. Dedicata* **36** (1990), 67–87.
26. Martin Kassabov and Tim R. Riley, Diameters of Cayley graphs of Chevalley groups, *Eur. J. Comb.* **28** (2007), no. 3, 791–800.
27. Michael Larsen, *Navigating the Cayley graph of $SL_2(\mathbb{F}_p)$*, International Mathematics Research Notices, IMRN **27** (2003), 1465–1471.
28. Alan Lauder, Continued fractions of Laurent series with partial quotients from a given set, *Acta Arithmetica XC* **3** (1999), 252–271.
29. Martin W. Liebeck and *Aner Shalev*, The probability of generating a finite simple group, *Geom. Dedicata* **56** (1995), 103–113.
30. Alexander Lubotzky, *Discrete Groups, Expanding Graphs and Invariant Measures*, Birkhäuser Verlag, 1994.
31. 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.
32. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Inc., Boca Raton, FL, 1996.
33. J. P. Mesirov and M. M. Sweet, Continued fraction expansions of rational expressions with irreducible denominators in characteristic 2, *J. Number Theory* **27** (1987), 144–148.
34. National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-3, 2009.
35. Jacques Patarin, Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms, *EUROCRYPT* (Ueli M. Maurer, ed.), Lecture Notes in Computer Science, vol. 1070, Springer, 1996, pp. 33–48.
36. Christophe Petit, *On graph-based cryptographic hash functions*, Ph.D. thesis, Université catholique de Louvain, 2009. http://perso.uclouvain.be/christophe.petit/files/thesis.pdf.
37. Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater, Full cryptanalysis of LPS and Morgenstern hash functions, *SCN* (Rafail Ostrovsky,

Roberto De Prisco, and Ivan Visconti, eds.), Lecture Notes in Computer Science, vol. 5229, Springer, 2008, pp. 263–277.

38. ———, *Cayley hashes: A class of efficient graph-based hash functions*, 2007, Available at `http://perso.uclouvain.be/christophe.petit/index.html`

39. Christophe Petit and Jean-Jacques Quisquater, Preimages for the Tillich-Zémor hash function, *Selected Areas in Cryptography* (Alex Biryukov, Guang Gong, and Douglas R. Stinson, eds.), Lecture Notes in Computer Science, vol. 6544, Springer, 2010, pp. 282–301.

40. Christophe Petit, Jean-Jacques Quisquater, Jean-Pierre Tillich, and Gilles Zémor, Hard and easy components of collision search in the Zémor-Tillich hash function: New attacks and reduced variants with equivalent security, *CT-RSA* (Marc Fischlin, ed.), Lecture Notes in Computer Science, vol. 5473, Springer, 2009, pp. 182–194.

41. László Pyber and Endre Szabó, *Growth in finite simple groups of Lie type*, arXiv:1001.4556v1, Jan 2010.

42. Oded Regev, Lattice-based cryptography, *CRYPTO* (Cynthia Dwork, ed.), Lecture Notes in Computer Science, vol. 4117, Springer, 2006, pp. 131–141.

43. T. R. Riley, Navigating in the Cayley graphs of $SL_N(\mathbb{Z})$ and $SL_N(\mathbb{F}_p)$, *Geom. Dedicata* **113/1** (2005), 215–229.

44. Rainer Steinwandt, Markus Grassl, Willi Geiselmann, and Thomas Beth, Weaknesses in the $SL_2(\mathbb{F}_{2^n})$ hashing scheme, *CRYPTO* (Mihir Bellare, ed.), Lecture Notes in Computer Science, vol. 1880, Springer, 2000, pp. 287–299.

45. Jean-Pierre Tillich and Gilles Zémor, Hashing with SL_2, *CRYPTO* (Yvo Desmedt, ed.), Lecture Notes in Computer Science, vol. 839, Springer, 1994, pp. 40–49.

46. ———, Collisions for the LPS expander graph hash function, *EUROCRYPT* (Nigel P. Smart, ed.), Lecture Notes in Computer Science, vol. 4965, Springer, 2008, pp. 254–269.

47. Jean-Pierre Tillich and Gilles Zémor, Group-theoretic hash functions, *Proceedings of the First French-Israeli Workshop on Algebraic Coding (London, UK)*, Springer-Verlag, 1993, pp. 90–110.

48. Gilles Zémor, Hash functions and graphs with large girths, *EUROCRYPT* (Donald W. Davies, ed.), Lecture Notes in Computer Science, vol. 547, Springer, 1991, pp. 508–511.