Using shifted conjugacy in braid-based cryptography

Patrick Dehornoy

Contemporary Mathematics

Using shifted conjugacy in braid-based cryptography

Patrick DEHORNOY

Abstract. Conjugacy is not the only possible primitive for designing braid-based protocols. To illustrate this principle, we describe a Fiat–Shamir-style authentication protocol that be can be implemented using any binary operation that satisfies the left self-distributive law. Conjugation is an example of such an operation, but there are other examples, in particular the shifted conjugation on Artin's braid group B_∞, and the finite Laver tables. In both cases, the underlying structures have a high combinatorial complexity, and they lead to difficult problems.

Most of the braid-based cryptographic schemes proposed so far [**1, 18, 3**] rely on the supposed complexity of the conjugation operation in Artin's braid groups. In this note, we would like to stress the fact that conjugation is by far not the only possible primitive operation for designing braid-based protocols.

To illustrate this general idea on a concrete example, we shall discuss an authentication scheme directly reminiscent of the Fiat–Shamir scheme, and a variant of some scheme considered in [**20**] in the case of braids. We show that such a scheme can be implemented naturally in every algebraic system that involves a binary operation that satisfies the algebraic law $x(yz) = (xy)(xz)$, called (left) self-distributivity. Conjugation on any group is an example of such an operation, but there are other examples, in particular the operation that we call *shifted conjugation* on Artin's braid group B_∞. There are reasons to think that sfifted conjugation is (much) more complicated than standard conjugation, and it could provide a promising alternative primitive for braid-based cryptography.

We also mention the Laver tables, which provide other examples of self-distributive operations, this time on a finite underlying domain of size 2^n. Again, these combinatorially very complex structures could provide a valuable platform.

1. A Fiat–Shamir-like authentication scheme

Here we start with the general principle of the Fiat–Shamir authentication scheme, and show that, under rather natural hypotheses, it can be implemented in any algebraic system involving a self-distributive binary operation.

1.1. The general principle. Let us start with an arbitrary set S, and try to construct an authentication scheme using the elements of S. To this end, we assume that a function F_s of S into itself is attached to each element of S and that there exist efficiently sampleable distributions on S such that, provided s and p are chosen according to them, the probability that s can be retrieved from the pair

1991 *Mathematics Subject Classification.* 94A60, 94A62, 68P25, 20F36.

Key words and phrases. cryptography; self-distributive operation; braid group; Fiat–Shamir scheme; Laver table; central duplication.

$(p, F_s(p))$ in feasible running time is negligible. Under such hypotheses, we can use s as a private key, and $(p, F_s(p))$ as a public key.

A natural idea for designing an authentication scheme is to let the prover appeal to a second, auxiliary (random) key r, and use $F_r(s)$ as a disguised version of s. What we need for a Fiat–Shamir-like authentication scheme is a commitment of the verifier guaranteeing that r is fixed, *and* an equality witnessing that $F_r(s)$ is connected in some way to s, *via* the commitment of the prover. As the elements p and $F_s(p)$ are public, it is natural to use $F_r(p)$ and/or $F_r(F_s(p))$ as the commitment(s) of the prover. Indeed, the assumption that x cannot be retrieved from $(y, F_x(y))$, which is already needed for $(p, F_s(p))$, automatically guarantees that r cannot be retrieved from the commitments of the prover.

Then what we need is some equality connecting $x = F_r(p)$, $y = F_r(F_s(p))$, and s—in a way that heavily involves s, *i.e.*, in such a way that the probability for another $\widetilde{s}$ to give rise to the same equality is negligible. A simple, but very particular, solution is to require that F_s and F_r commute: in this case, the connection between x and y is just $y = F_s(x)$. This situation is essentially that considered in [**21, 18**], and it is not suitable in the current framework as the verifier would have to know the secret s.

A more general and flexible solution is to require that $F_r(F_s(p))$ be connected to $F_r(p)$ and s by some relation of the form $F_r(F_s(p)) = G_{r,s}(F_r(p))$ for some new function $G_{r,s}$. A not so special case is when $G_{r,s}$ is itself of the form $F_{g(r,s)}$ where g is some mapping of $S \times S$ into S: considering such a case is natural, because it avoids introducing a new family of functions and it enables one to work with the functions $(F_s)_{s\in S}$ solely. For the same reason, it is natural to consider the case when $g(r, s)$ is defined in terms of the F-functions, typically $g(r, s) = F_r(s)$. This leads to requiring that the functions F_s satisfy the condition

$$F_r(F_s(p)) = F_{F_r(s)}(F_r(p)), \tag{1.1}$$

and to use this equality for proving authentication.

1.2. An authentication scheme. The previous analysis leads to considering the following authentication scheme.

We assume that S is a set and $(F_s)_{s\in S}$ is a family of functions of S to itself that satisfies Condition (1.1). Then the public keys are a pair (p, p') of elements of S satisfying $p' = F_s(p)$, while s is Alice's private key. The authentication procedure consists in repeating k times the following three exchanges:

A chooses r in S, and sends the *commitments* $x = F_r(p)$ and $x' = F_r(p')$;
B chooses a random bit c and sends it to A;
For $c = 0$, A sends $y = r$, and B checks $x = F_y(p)$ and $x' = F_y(p')$;
For $c = 1$, A sends $y = F_r(s)$, and B checks $x' = F_y(x)$.

The correctness of the scheme directly follows from Condition (1.1). Its security relies on the following assumptions:

(*i*) It is impossible to retrieve s from the pair $(p, F_s(p))$, and, more generally, it is impossible to find $\widetilde{s}$ satisfying $F_s(p) = F_{\widetilde{s}}(p)$; similar assertions hold for the pairs $(p, F_r(p))$, $(p', F_r(p'))$, and $(F_r(p), F_r(p'))$;

(*ii*) It is impossible to deduce s from $F_r(s)$ when r is unknown.

1.3. Self-distributive operations. Specifying an S-indexed family of mappings of a set S into itself amounts to specifying a binary operation on S, namely the operation $*$ defined by $x*y = F_x(y)$. Conversely, $(F_s)_{s\in S}$ is the family of all left translations for $(S,*)$. Now, in terms of the operation $*$, Condition (1.1) becomes

$$r*(s*p) = (r*s)*(r*p), \tag{1.2}$$

i.e., it asserts that the operation $*$ satisfies the *left self-distributivity* law, usually denoted (LD) [**6**].

Definition 1.1. A set equipped with a binary operation satisfying (1.2) is called an it LD-system.

Translating the previous authentication scheme into the language of LD-systems yields the following version.

Assume that $(S,*)$ is an LD-system. The public keys are a pair (p,p') of elements of S satisfying $p' = s*p$, while s is Alice's private key. The authentication procedure consists in repeating k times the following three exchanges:

A chooses r in S, and sends the commitments $x = r*p$ and $x' = r*p'$;
B chooses a random bit c and sends it to A;
For $c=0$, A sends $y=r$, and B checks $x = y*p$ and $x' = y*p'$;
For $c=1$, A sends $y = r*s$, and B checks $x' = y*x$.

---

2. LD-systems

The algebraic platforms eligible for implementing the scheme of Section 1 are LD-systems, and we are led to reviewing the existing examples of such algebraic systems.

2.1. Classical examples. A trivial example of an LD-system is given by an arbitrary set S equipped with the operation $x*y = y$, or, more generally,

$$x*y = f(y),$$

where f is any map of S into itself. Such examples are clearly not relevant for the scheme of Section 1, as the secret s plays no role in the computation.

The most classical example of an LD-system is provided by a group G equipped with the conjugacy operation

$$x*y = xyx^{-1}.$$

When G is a non-abelian group for which the conjugacy problem is sufficiently difficult, G is relevant for the scheme of Section 1, and, more generally, for the various schemes based on the Conjugacy Search Problem such as those of [**21, 18**] or [**1**]. Typical platform groups that have been much discussed in this context are Artin's braid groups B_n; in particular, the specific scheme considered in Section 1 is, in the case of the group B_n, (a variant of a scheme) proposed by H. Sibert in his PhD thesis [**20**].

2.2. The shifted conjugacy of braids. Now, and this is the point we wish to emphasize here, examples of LD-system of a very different flavour exist.

Those LD-systems are connected with *free* LD-systems, *i.e.*, LD-systems that satisfy no other relations than those resulting from self-distributivity itself. It is easy to understand that a group equipped with conjugacy, even a free group, is not a free LD-system: indeed, the conjugacy operation always satisfies (among others) the idempotency law $x * x = x$, and the latter is not a consequence of (LD), as shows the existence of non-idempotent LD-system such as the integers equipped with $x * y = y + 1$.

Actually, free LD-systems are quite complicated objects, and we refer to [**6**], which contains an extensive description. For our purpose, it will be enough to know that, for some deep reasons that need not be explained here, there exists a simple self-distributive operation on Artin's braid group B_∞ that includes many copies of the free LD-system with one generator. Let us first recall the definition [**2, 6**]:

DEFINITION 2.1 (braid group). For $n \geqslant 2$, Artin's *braid group* B_n is defined to be the group with presentation

$$\langle \sigma_1, ..., \sigma_{n-1} \,;\, \sigma_i\sigma_j = \sigma_j\sigma_i \text{ for } |i-j| \geqslant 2,\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ for } |i-j| = 1\rangle. \tag{2.1}$$

For each n, the identity mapping on $\{\sigma_1, ..., \sigma_{n-1}\}$ induces an embedding of B_n into B_{n+1}, so that the groups B_n naturally arrange into an inductive system of groups, and the limit is denoted by B_∞: this is just the group generated by an infinite family $\sigma_1, \sigma_2, ...$ subject to the relations (2.1).

LEMMA 2.2. *Let* d *be the shift mapping of the sequence* $(\sigma_1, \sigma_2, ...)$*, i.e., the function mapping* σ_i *to* σ_{i+1} *for each* i*. Then* d *induces an injective morphism of* B_∞ *into itself.*

PROOF (SKETCH). As the relations of (2.1) are invariant under shifting the indices, d induces a well-defined endomorphism of B_∞. That this endomorphism is injective follows from the interpretation of the elements of B_∞ in terms of braid diagrams [**2, 6**]: the geometric operation of deleting the leftmost strand is then well-defined, and it enables one to deduce $x = y$ from $\mathrm{d}x = \mathrm{d}y$. □

The main notion is then the following.

DEFINITION 2.3 (shifted conjugacy). For x, y in B_∞, we put

$$x * y = x \cdot \mathrm{d}y \cdot \sigma_1 \cdot \mathrm{d}x^{-1}. \tag{2.2}$$

The above operation is a skew version of conjugation: y appears in the middle, and it is surrounded by x and x^{-1}; the difference with ordinary conjugation lies in the introduction of the shift d, and of the generator σ_1. The reader can check the equalities

$$1 * 1 = \sigma_1, \qquad 1 * \sigma_1 = \sigma_2\sigma_1, \qquad \sigma_1 * 1 = \sigma_1^2\sigma_2^{-1}, \qquad \sigma_1 * \sigma_1 = \sigma_2\sigma_1,$$

which show that shifted conjugation is quite different from conjugation.

PROPOSITION 2.4. [**4, 6**] *The system* $(B_\infty, *)$ *is an LD-system. Moreover, every braid generates under* $*$ *a free sub-LD-system.*

Checking that the operation defined in (2.2) satisfies the LD law is an easy verification. In the context of groups, the property that every element generates a

free subgroup is torsion-freeness. Thus Proposition 2.4 expresses that $(B_\infty, *)$ is in some sense a torsion-free LD-system.

Understanding why the weird definition of shifted conjugacy has to appear requires a rather delicate analysis which is the main subject of the book [**6**]. It can be observed that, once the definition (2.2) is used, braids inevitably appear. Indeed, if we assume that G is a group, that f is an endomorphism of G, and that a is a fixed element of G, then defining

$$x * y = x\, f(y)\, a\, f(x)^{-1}$$

yields a left self-distributive operation (if and) only if the subgroup of G generated by the elements $f^n(a)$ is a homomorphic image of Artin's braid group B_∞, *i.e.*, up to an isomorphism, it is B_∞ or a quotient of the latter group.

2.3. Discussion. Our intuition is that the LD-system $(B_\infty, *)$, *i.e.*, braids equipped with shifted conjugacy, might be a promising platform for implementing the scheme of Section 1—or, more generally, for implementing any scheme based on a left self-distributive operation. This intuition ought to be confirmed by an experimental evidence, which at this early stage is not yet available. Here we content ourselves with a few remarks about the respective properties of conjugacy and shifted conjugacy in B_∞.

First, note that in general using free structures does not seem a very good idea in cryptography, as by definition the free structures are those in which the least possible number of equalities are satisfied, a not very good framework for hiding things. That is why, for instance, a free LD-system would probably not be the optimal platform for implementing the scheme of Section 1. However, the LD-system $(B_\infty, *)$ is far from being free, and it is even conjectured that it contains no free LD-system with two generators. For instance, the equality $\sigma_1 * \sigma_1 = \sigma_2 * \sigma_2$ $(= \sigma_2\sigma_1)$ shows that the sub-LD-system generated by σ_1 and σ_2 is not free. No presentation of $(B_\infty, *)$ as an LD-system is known.

Practically, using shifted conjugacy of braids as suggested here relies on the difficulty of the following problem, which is analogous to the Conjugacy Search Problem:

> **Shifted Conjugacy Seach Problem**: Assuming that s, p are braids in B_∞ and $p' = s * p$ holds, find a braid $\widetilde{s}$ satisfying $p' = \widetilde{s} * p$.

Contrary to the Conjugacy Seach Problem, no solution to the Shifted Conjugacy Search Problem is known so far. It is not even known whether the simple Shifted Conjugacy Problem is decidable, *i.e.*, whether one can effectively decide for two braids p, p' the existence of s satisfying $p' = s * p$. It is likely that shifted conjugacy is quite different from ordinary conjugacy, and that none of the many specific results established for the latter [**14, 11, 15**] extends to shifted conjugacy. In particular, we see no simple strategy for constructing the "shifted super summit set" of a braid p, defined as the family of all shifted conjugates of p with minimal canonical length—which is the key point in all solutions to the Conjugacy Problem known so far.

However, it is fair to mention that the Shifted Conjugacy Search Problem, which should not be threatened by specific attacks against the Conjugacy Search Problem [**16, 19**], remains, as the latter, an instance of the general Decomposition Problem and, as such, it is not a priori immune against length-based attacks [**17, 12, 13**].

To emphasize the difference between ordinary and shifted conjugations, we point

PROPOSITION 2.5 ([**5**], Corollary 1.8). *The mapping $f : s \mapsto s * 1$ is injective.*

In the case of ordinary conjugacy, every conjugate of 1 is 1, so the above injective function f is replaced with the constant function with value 1. By the way, very little is known about f. In particular, we raise

QUESTION 2.6. *Starting with a braid p, find s satisfying $s * 1 = p$ (when it exists).*

Once more, nothing is known. This might suggest to use f as a possible one-way function on braids.

3. The Laver tables and other algebraic systems

To conclude, we mention that braids are not the only possible platform for implementing self-distributive operations—and that the self-distributivity law is not even the only algebraic law eligible for the approach sketched in Section 1.

3.1. The Laver tables. Instead of resorting to an infinite LD-system like B_∞ equipped with shifted conjugacy, one could instead use finite LD-systems. Such algebraic systems are far from being completely understood, but there exists an infinite sequence of so-called Laver tables that plays a fundamental role among LD-systems—similar to the role of the cyclic groups $\mathbb{Z}/p\mathbb{Z}$ among finite abelian groups—and, at the same time, has a high combinatorial complexity.

We refer to Chapter X of [**6**] for details. For our current overview, it is enough to mention that, for each nonnegative integer n, there exists a unique LD-system A_n such that the underlying set is the 2^n elements interval $\{0, 1, ..., 2^n - 1\}$ and one has $p * 0 = p + 1$ for $0 \leqslant p \leqslant 2^n - 2$ and $2^n - 1 * 0 = 0$. The value of $p * q$ in A_n can be easily computed using a double induction on q increasing from 0 to $2^n - 1$ and for p decreasing from $2^n - 1$ to 0, using the rule

$$p * (q + 1) = (p * q) * (p * 0),$$

and observing that $p * q$ has to be always strictly larger than p. Table 2 displays the first four Laver tables.

A_0	0
0	0

A_1	0	1
0	1	1
1	0	1

A_2	0	1	2	3
0	1	3	1	3
1	2	3	2	3
2	3	3	3	3
3	0	1	2	3

A_3	0	1	2	3	4	5	6	7
0	1	3	5	7	1	3	5	7
1	2	3	6	7	2	3	6	7
2	3	7	3	7	3	7	3	7
3	4	5	6	7	4	5	6	7
4	5	7	5	7	5	7	5	7
5	6	7	6	7	6	7	6	7
6	7	7	7	7	7	7	7	7
7	0	1	2	3	4	5	6	7

TABLE 1. The Laver tables A_n with $0 \leqslant n \leqslant 3$

Several general phenomena can be observed on these particular examples. First, for each n, the table A_n with 2^n elements is the projection modulo 2^n of the table A_{n+1} with 2^{n+1} elements. In other words, if we use a length n binary representation for the elements of A_n, only the dominant bit of each value has to be computed in order to determine A_{n+1} from A_n. Next, every row in the table A_n is periodic, with a period that is a power of 2. More precisely, for each p, the row of p in A_n consists of 2^k values

$$r_0 = p + 1 < r_1 < ... < r_{2^k-1} = 2^n - 1$$

repeated 2^{n-k} times. One can show that, if $(r_0, ..., r_{2^k-1})$ is the periodic pattern in the row of p in A_n, with $r_0 = p+1$ and $r_{2^k-1} = 2^n - 1$ and if t denotes the smallest integer for which one has $p * t \geqslant 2^n$ in A_{n+1}, then

- (i) either $t = 2^k$ holds, the period of p doubles from 2^k to 2^{k+1} between A_n and A_{n+1}, and the periodic pattern in A_{n+1} is $(r_0, ..., r_{2^k-1}, r_0 + 2^n, ..., r_{2^k-1} + 2^n)$,

- (ii) or $0 \leqslant t < 2^k$ holds, the period of p remains 2^k in A_{n+1}, and the periodic pattern in A_{n+1} is $(r_0, ..., r_{t-1}, r_t + 2^n, ..., r_{2^k-1} + 2^n)$.

In each case, the only piece of information needed to construct the row of p in A_{n+1} from the row of p in A_n is the value of t, which is called the *threshold* of p in A_n, and, therefore, the list of thresholds suffices to construct A_{n+1} from A_n (*cf.* Table 2). Note that, as A_n is the projection of A_{n+1}, we can consider that we work in the inverse limit A_∞ of the tables A_n, *i.e.*, we are constructing an LD-operation on 2-adic numbers.

A_0	1
	–

A_1	1	2
	0	1

A_2	1	2	3	4
	1	0	0	2

A_3	1	2	3	4	5	6	7	8
	2	2	1	0	0	0	0	4

TABLE 2. Threshold table for A_n with $1 \leqslant n \leqslant 3$

The reason for mentioning the Laver tables here is that their combinatorial properties seem to be very complicated. In particular, predicting the values in the first half of the sequence of thresholds is extremely difficult (the values in the second half are always 0, ..., 0, 2^n): this is witnessed by the results of [**8, 9, 10**] which show that fast growing functions are necessarily involved here.

3.2. Central duplication. As a final remark, we come back to the Fiat–Shamir-like authentication scheme of Section 1. We noted that its security requires two conditions, namely one that is directly connected with the difficulty of what can be called the $*$-Search Problem, and the additional requirement that communicating $F_r(s)$, *i.e.*, $r * s$, gives no practical information about s when r remains unknown. Using the latter condition to forge an attack seems unclear, but, at least for aesthetic reasons, we might like to avoid it. This can be done, at the expense of changing the algebraic law.

Indeed, instead of communicating $F_r(s)$ in case $c = 1$ of the authentication scheme, Alice could communicate $F_s(r)$. In this case, the supposed difficulty of the $*$-Search Problem guarantees that $F_s(r)$ gives no information about s. Now, when the scheme is modified in this way, the equality checked by the verifier has to be modified as well. If we keep the same principle, we are led to replace Condition (1.1) with

$$F_r(F_s(p)) = F_{F_s(r)}(F_r(p)). \tag{3.1}$$

When Condition (3.1) is translated into the language of binary operations, we obtain a new algebraic law, namely

$$r * (s * p) = (s * r) * (r * p), \tag{3.2}$$

instead of left self-distributivity. Nothing specific is known about this law so far, but it should be possible to use the general method explained for a similar law in [**7**] to construct concrete examples of algebras that satisfy it.

4. Conclusion

We discussed various non-classical algebraic operations that could possibly be used as cryptographical primitives, typically for a Fiat–Shamir-like authentication scheme. The most promising example seems to be the shifted conjugacy operation on braids. At the least, the existence of such an operation shows that conjugacy is not the only possible primitive for braid-based cryptography, and that further investigation in this direction is needed.

References

[1] I. Anshel, M. Anshel, B. Fisher, & D. Goldfeld, *New key agreement protocols in braid group cryptography, CT–RSA 2001 (San Francisco, CA)*, Springer Lect. Notes Comp. Sci. **2020** (2001) 1–15.

[2] J. Birman, *Braids, Links, and Mapping Class Groups*, Annals of Math. Studies vol. 82, Princeton Univ. Press (1975).

[3] J.C. Cha, K.H. Ko, S.J. Lee, J.W. Han, J.H. Cheon, *An efficient implementation of braid groups, AsiaCrypt 2001*, Springer Lect. Notes in Comp. Sci. **2048** (2001) 144–156.

[4] P. Dehornoy, *Braid groups and left distributive operations*, Trans. Amer. Math. Soc. **345-1** (1994) 115–151.

[5] P. Dehornoy, *Strange questions about braids*, J. Knot Th. and its Ramifications **8-5** (1999) 589–620.

[6] P. Dehornoy, *Braids and Self-Distributivity*, Progress in Math. vol. 192, Birkhäuser (2000).

[7] P. Dehornoy, *Study of an identity*, Alg. Universalis **48** (2002) 223–248.

[8] R. Dougherty, *Critical points in an algebra of elementary embeddings*, Ann. P. Appl. Logic **65** (1993) 211–241.

[9] R. Dougherty & T. Jech, *Finite left-distributive algebras and embedding algebras*, Advances in Math. **130** (1997) 201–241.

[10] A. Drápal, *Persistence of left distributive algebras*, J. Pure Appl. Algebra **105** (1995) 137–165.

[11] N. Franco, J. González-Meneses, *Conjugacy problem for braid groups and Garside groups*, J. of Algebra **266-1** (2003) 112–132.

[12] D. Garber, S. Kaplan, M. Teicher, B. Tsaban, & U. Vishne, *Length-based conjugacy search in the braid group*, arXiv math.GT/0209267

[13] D. Garber, S. Kaplan, M. Teicher, B. Tsaban, & U. Vishne, *Probabilistic solutions of equations in the braid group*, Adv. Applied Math. **35** (2005) 323–334.

[14] F.A. Garside, *The braid group and other groups*, Quart. J. Math. Oxford **20-78** (1969) 235–254.

[15] V. Gebhardt, *A new approach to the conjugacy problem in Garside groups*, J. of Algebra **292–1** (2005) 282–302.

[16] D. Hofheinz & R. Steinwandt, *A practical attack on some braid group based cryptographic primitives*, PKC 2003; Springer Lect. Notes in Comp. Sci.; 2567; 2002; 187–198.

[17] J. Hughes & A. Tannenbaum, *Length-based attacks for certain group based encryption rewriting systems*, Workshop SECI02 Sécurité de la communication sur internet, September 2002, Tunis, http://www.storagetek.com/hughes/SECI02.pdf.

[18] K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang, & C. Park, *New public-key cryptosystem using braid groups, Crypto 2000*, Springer Lect. Notes Comp. Sci. **1880** (2000) 166–184.

[19] A.G. Myasnikov, V. Shpilrain & A. Ushakov, *A practical attack on some braid group based cryptographic protocols, Crypto 2005*, Springer Lect. Notes Comp. Sc. **3621** (2005) 86–96.

[20] H. Sibert, *Algorithmique des groupes de tresses*, Thèse de doctorat, Université de Caen (2003).

[21] V.M. Sidelnikov, M.A. Cherepnev, & V.Y. Yashcenko, *Systems of open distribution of keys on the basis of noncommutative semigroups*, Ross. Acad. Nauk Dokl. **332-5** (1993) English translation: Russian Acad. Sci. Dokl. Math. 48-2 (1994) 384–386.

Laboratoire de Mathématiques Nicolas Oresme UMR 6139, Université de Caen, 14032 Caen, France

E-mail address: dehornoy math.unicaen.fr

URL: //www.math.unicaen.fr/∼dehornoy