INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team CARAMEL

Cryptology, Arithmetic: Hardware and Software

Nancy - Grand Est

Theme : Algorithms, Certification, and Cryptography

Table of contents

1. Team

Research Scientists

Pierrick Gaudry [Team leader, Senior Researcher since October 1st, CNRS, HdR]
Jérémie Detrey [Junior Researcher, INRIA]
Emmanuel Thomé [Junior Researcher, INRIA]
Paul Zimmermann [Senior Researcher, INRIA, HdR]

Faculty Member

Marion Videau [Associate Professor, Université Henri Poincaré; on secondment to ANSSI until January 2011]

Technical Staff

Lionel Muller [ADT grant until August 2011]

PhD Students

Gaëtan Bisson [MESR grant, INPL and Technische Universiteit Eindhoven; defense planned in 2011]
Romain Cosset [INRIA/DGA grant; defense planned in 2011]
Nicolas Estibals [Contrat doctoral, Université Henri Poincaré; defense planned in 2012]
Alexander Kruppa [CNRS grant; defended on January 28th]
Damien Robert [MESR grant, Université Henri Poincaré; defended on July 21st]

Post-Doctoral Fellows

Sylvain Chevillard [Until November 14th]
Antonio Vera [Until January 31st]

Administrative Assistant

Emmanuelle Deschamps [part time]

Others

Răzvan Bărbulescu [Master project student, since September]
Julie Feltin [Internship, June 1st–July 15th]
Thomas Prest [Internship, June 1st–July 31st]
Richard Brent [Professor, University of Canberra, external collaborator]

2. Overall Objectives

2.1. Introduction

A general keyword that could encompass most of our research objectives is *arithmetic*. Indeed, in the CARAMEL team, the goal is to push forward the possibilities to compute efficiently with objects having an arithmetic nature. This includes integers, real and complex numbers, polynomials, finite fields, and, last but not least, algebraic curves.

Our main application domains are public-key cryptography and computer algebra systems. Concerning cryptography, we concentrate on the study of the primitives based on the factorization problem or on the discrete-logarithm problem in finite fields or (Jacobians of) algebraic curves. Both the constructive and destructive sides are of interest to CARAMEL. For applications in computer algebra systems, we are mostly interested in arithmetic building blocks for integers, floating-point numbers, polynomials, and finite fields. Also some higher level functionalities like factoring and discrete-logarithm computation are usually desired in computer algebra systems.

Since we develop our expertise at various levels, from most low-level software or hardware implementation of basic building blocks to complicated high-level algorithms like integer factorization or point counting, we have remarked that it is often too simple-minded to separate them: we believe that the interactions between low-level and high-level algorithms are of utmost importance for arithmetic applications, yielding important improvements that would not be possible with a vision restricted to low- or high-level algorithms.

We emphasize three main directions in the CARAMEL team:

- Integer factorization and discrete-logarithm computation in finite fields.

  We are in particular interested in the number field sieve algorithm (NFS) that is the best known algorithm for factoring large RSA-like integers, and for solving discrete logarithms in prime finite fields. A sibling algorithm, the function field sieve (FFS), is the best known algorithm for computing discrete logarithms in finite fields of small characteristic.

  In all these cases, we plan to improve on existing algorithms, with a view towards practical considerations and setting new records.

- Algebraic curves and cryptography.

  Our two main research interests on this topic lie in genus-2 cryptography and in the arithmetic of pairings, mostly on the constructive side in both cases. For genus-2 curves, a key algorithmic tool that we develop is the computation of explicit isogenies; this allows improvements for cryptography-related computations such as point counting in large characteristic, complex-multiplication construction and computation of the ring of endomorphisms.

  For pairings, our principal concern is the optimization of pairing computations, in particular in hardware, or in constrained environments. We plan to develop automatic tools to help in choosing the most suitable (hyper-)elliptic curve and generating efficient hardware for a given security level and set of constraints.

- Arithmetic.

  Integer, finite-field and polynomial arithmetics are ubiquitous to our research. We consider them not only as tools for other algorithms, but as a research theme *per se*. We are interested in algorithmic advances, in particular for large input sizes where asymptotically fast algorithms become of practical interest. We also keep an important implementation activity, both in hardware and in software.

2.2. Highlights

The highlights for year 2010 in the CARAMEL team are:

- the factorization of RSA-768, which happened in fact in December 12th, 2009, but was publicly announced on January 6, 2010;
- the publication of the book *Modern Computer Arithmetic*, by Richard Brent and Paul Zimmermann.

3. Scientific Foundations

3.1. Scientific Foundations

One of the main topics for our project is public-key cryptography. After 20 years of hegemony, the classical public-key algorithms (whose security is based on integer factorization or discrete logarithm in finite fields) are currently being overtaken by elliptic curves. The fundamental reason for this is that the best-known algorithms for factoring integers or for computing discrete logarithms in finite fields have a subexponential complexity, whereas the best known attack for elliptic-curve discrete logarithms has exponential complexity. As a consequence, for a given security level 2^n, the key sizes must grow linearly with n for elliptic curves, whereas they grow like n^3 for RSA-like systems. As a consequence, several governmental agencies, like the NSA or the BSI, now recommend to use elliptic-curve cryptosystems for new products that are not bound to RSA for backward compatibility.

Besides RSA and elliptic curves, there are several alternatives currently under study. There is a recent trend to promote alternate solutions that do not rely on number theory, with the objective of building systems that would resist a quantum computer (in contrast, integer factorization and discrete logarithms in finite fields and elliptic curves have a polynomial-time quantum solution). Among them, we find systems based on hard problems in lattices (NTRU is the most famous), those based on coding theory (McEliece system and improved versions), and those based on the difficulty to solve multivariate polynomial equations (HFE, for instance). None of them has yet reached the same level of popularity as RSA or elliptic curves for various reasons, including the presence of unsatisfactory features (like a huge public key), or the non-maturity (system still alternating between being fixed one day and broken the next day).

Returning to number theory, an alternative to RSA and elliptic curves is to use other curves and in particular genus-2 curves. These so-called hyperelliptic cryptosystems have been proposed in 1989 [29], soon after the elliptic ones, but their deployment is by far more difficult. The first problem was the group law. For elliptic curves, the elements of the group are just the points of the curve. In a hyperelliptic cryptosystem, the elements of the group are points on a 2-dimensional variety associated to the genus-2 curve, called the Jacobian variety. Although there exist polynomial-time methods to represent and compute with them, it took some time before getting a group law that could compete with the elliptic one in terms of speed. Another question that is still not yet fully answered is the computation of the group order, which is important for assessing the security of the associated cryptosystem. This amounts to counting the points of the curve that are defined over the base field or over an extension, and therefore this general question is called point-counting. In the past ten years there have been major improvements on the topic, but there are still cases for which no practical solution is known.

Another recent discovery in public-key cryptography is the fact that having an efficient bilinear map that is hard to invert (in a sense that can be made precise) can lead to powerful cryptographic primitives. The only examples we know of such bilinear maps are associated with algebraic curves, and in particular elliptic curves: this is the so-called Weil pairing (or its variant, the Tate pairing). Initially considered as a threat for elliptic-curve cryptography, they have proven to be quite useful from a constructive point of view, and since the beginning of the decade, hundreds of articles have been published, proposing efficient protocols based on pairings. A long-lasting open question, namely the construction of a practical identity-based encryption scheme, has been solved this way. The first standardization of pairing-based cryptography has recently occurred (see ISO/IEC 14888-3 or IEEE P1363.3), and a large deployment is to be expected in the next years.

Despite the raise of elliptic curve cryptography and the variety of more or less mature other alternatives, classical systems (based on factoring or discrete logarithm in finite fields) are still going to be widely used in the next decade, at least, due to resilience: it takes a long time to adopt new standards, and then an even longer time to renew all the software and hardware that is widely deployed.

This context of public-key cryptography motivates us to work on integer factorization, for which we have acquired expertise, both in factoring moderate-sized numbers, using the ECM (Elliptic Curve Method) algorithm, and in factoring large RSA-like numbers, using the number field sieve algorithm. The goal is to follow the transition from RSA to other systems and continuously assess its security to adjust key sizes. We also want to work on the discrete-logarithm problem in finite fields. This second task is not only necessary for assessing the security of classical public-key algorithms, but is also crucial for the security of pairing-based cryptography.

We also plan to investigate and promote the use of pairing-based and genus-2 cryptosystems. For pairings, this is mostly a question of how efficient can such a system be in software, in hardware, and using all the tools from fast implementation to the search for adequate curves. For genus 2, as said earlier, constructing an efficient cryptosystem requires some more fundamental questions to be solved, namely the point-counting problem.

We summarize in the following table the aspects of public-key cryptography that we address in the CARAMEL team.

public-key primitive	cryptanalysis	design	implementation
RSA	X	–	–
Finite Field DLog	X	–	–
Elliptic Curve DLog	–	–	Soft
Genus 2 DLog	–	X	Soft
Pairings	X	X	Soft/Hard

Another general application for the project is computer algebra systems (CAS), that rely in many places on efficient arithmetic. Nowadays, the objective of a CAS is not only to have more and more features that the user might wish, but also to compute the results fast enough, since in many cases, the CAS are used interactively, and a human is waiting for the computation to complete. To tackle this question, more and more CAS use external libraries, that have been written with speed and reliability as first concern. For instance, most of today's CAS use the GMP library for their computations with big integers. Many of them will also use some external Basic Linear Algebra Subprograms (BLAS) implementation for their needs in numerical linear algebra.

During a typical CAS session, the libraries are called with objects whose sizes vary a lot; therefore being fast on all sizes is important. This encompasses small-sized data, like elements of the finite fields used in cryptographic applications, and larger structures, for which asymptotically fast algorithms are to be used. For instance, the user might want to study an elliptic curve over the rationals, and as a consequence, check its behaviour when reduced modulo many small primes; and then he can search for large torsion points over an extension field, which will involve computing with high-degree polynomials with large integer coefficients.

Writing efficient software for arithmetic as it is used typically in CAS requires the knowledge of many algorithms with their range of applicability, good programming skills in order to spend time only where it should be spent, and finally good knowledge of the target hardware. Indeed, it makes little sense to disregard the specifics of the possible hardware platforms intended, even more so since in the past years, we have seen a paradigm shift in terms of available hardware: so far, it used to be reasonable to consider that an end-user running a CAS would have access to a single-CPU processor. Nowadays, even a basic laptop computer has a multi-core processor and a powerful graphics card, and a workstation with a reconfigurable coprocessor is no longer science-fiction.

In this context, one of our goals is to investigate and take advantage of these influences and interactions between various available computing resources in order to design better algorithms for basic arithmetic objects. Of course, this is not disconnected from the others goals, since they all rely more or less on integer or polynomial arithmetic.

4. Application Domains

4.1. Cryptology

The first application domain for our research is cryptology. This includes cryptography (constructive side) and cryptanalysis (breaking systems). For the cryptanalysis part, although it has practical implications, we do not expect any transfer in the classical sense of the term: it is more directed to governmental agencies and the end-users who build their trust, based on the cryptanalysis effort.

4.1.1. Cryptography

Our cryptographic contributions are related to multiple facets of the large realm of curve-based cryptology. While it is quite clear that a satisfying range of algorithms exists in order to provide cryptographers with elliptic curves having a suitably hard discrete logarithm (as found in cryptographic standards for instance), one must bear in mind that refinements of the requirements and extensions to curves of higher genus raise several interesting problems. Our work contributes to expanding the cryptographer's capabilities in these areas.

In the context of genus-2 curves, our work aims at two goals. First, improvements on the group law on selected curves yield better speed for the associated cryptosystems. The cryptographic primitives, and then the whole suite of cryptographic protocols built upon such curves would be accelerated. The second goal is the expansion of the set of curves that can be built given a set of desired properties. Using point counting algorithms for arbitrary curves, a curve offering a 128-bit security level, together with nice properties for fast arithmetic, has been computed by CARAMEL[24]. Another natural target for construction of curves for cryptography is also the suitability of curves for pairings. We expect to be able to compute such curves.

Implementations of curve-based cryptography, both in hardware and software, are a necessary step on the way to assessing cryptographic speed. We plan to provide such implementations. In particular, on the hardware side, one of our goals is the design of a complete cryptographic coprocessor, including all the primitives for curve-based and pairing-based cryptography, providing optimized and configurable efficiency vs area trade-off.

4.1.2. Cryptanalysis

Our research on cryptanalysis is important for the cryptographic industry: by detecting weak instances, and setting new records we contribute to the definition of recommended families of systems together with their key sizes. The user's confidence in a cryptographic primitive is also related to how well the underlying problem is studied by researchers.

In particular, our involvement in computations with "NFS-like" algorithms encompasses of course the task of assessing the computational limits for integer factorization and discrete-logarithm computations. The impact of the former is quite clear as it concerns the RSA algorithm; record-sized computations attract broad interest and determine updates on key-length recommendations. The latter are particularly important for pairing-based cryptography, since, in this context, one naturally encounters discrete-logarithm problems in extension fields of large degree.

4.1.3. Standardization

4.1.3.1. Floating-point arithmetic

The IEEE 754 standard for floating-point arithmetic was revised in 2008. The main new features are some new formats for decimal computations, and the recommendation of correctly rounded transcendental functions. The new decimal formats should not have an impact on our work, since we either use integer-only arithmetic, or arbitrary-precision binary floating-point arithmetic through the GNU MPFR library.

A new standard (P1788) is currently under construction for interval arithmetic. We are not officially involved in this standard, but we follow the discussions, to check in particular that the proposed standard will also cover arbitrary precision (interval) arithmetic.

4.1.3.2. Curve-based cryptography

Elliptic-curve cryptography has been standardized for almost 10 years now, in the IEEE P1363 standard. This standard provides key agreement, signature and encryption schemes, based on integer factorization, discrete logarithm in finite fields and in elliptic curves. There is another standardization effort, called SECG, which is mostly lead by the Certicom company, with the goal to maintain interoperability between different implementations. In particular, the SECG documents give explicit elliptic curves that can be used for cryptography. Similarly, some elliptic curves have been standardized by the US government; the latest version comes from the NSA Suite B that includes only elliptic curves defined over prime fields.

In the long term, those standards are a natural place to promote genus-2 curve cryptography, and by the time we consider that the curves we propose are mature enough, we will look for an industrial partner to help us pushing towards their standardization.

4.1.3.3. Pairing-based cryptography

Despite their very recent discovery, identity-based cryptosystems—and more generally pairing-based cryptosystems—have already spawned several international standardization efforts.

The first standard, part of ISO/IEC 14888-3, was published in 2006. However, it almost exclusively focuses on protocols and therefore is of little interest to us. On the other hand, the IEEE P1363.3 standard, which is still in preparation, is planned to offer more details as to the considered curves and pairings on which the protocols are based.

Although we are not officially involved in the elaboration of this standard, we have already participated in the review process of its first draft.

4.1.4. Computer algebra systems

Some of our software libraries are being used by computer algebra systems. Most of those libraries are free software, with a license that allows proprietary systems to link them. This gives us a maximal visibility, with a large number of users.

4.1.4.1. Magma

Magma is a very large computational algebra package. It provides a mathematically rigorous environment for computing with algebraic, number-theoretic, combinatoric, and geometric objects. It is developed in Sydney, by the team around John Cannon. It is non-commercial (in the sense that its goal is not to make profit), but is not freely distributed and is not open-source.

Several members of the team have visited Sydney — a few years ago — to contribute to the development of Magma, by implementing their algorithms or helping in integrating their software. Our link to Magma exists also via the libraries it uses: it currently links GNU MPFR and MPC for its floating-point calculations, and links GMP-ECM as part of its factorization suite.

4.1.4.2. Pari-GP

Pari/GP is a computational number theory system that is composed of a C library and an interpreter on top of it. It is developed in Bordeaux, where Karim Belabas from the LFANT project-team is the main maintainer. Its license is GPL. Although we do not directly contribute to this package, we have good contact with the developers and in the future, GNU MPFR and MPC could be included.

4.1.4.3. Sage

Sage is a fairly large scale and open-source computer algebra system written in Python. Sage aggregates a large amount of existing free software, aiming at the goal of selecting the fastest free software package for each given task. The motto of Sage is that instead of "reinventing the wheel" all the time, Sage is "building the car". To date, Sage links GNU MPFR, GMP-ECM, and MPC as optional package since 2010 (this was the result of a huge work done by Philippe Théveny in the MPtools ODL which finished in 2009). Plans exist to link GF2X and CADO-NFS into Sage.

5. Software

5.1. Introduction

A major part of the research done in the CARAMEL team is published within software. On the one hand, this enables everyone to check that the algorithms we develop are really efficient in practice; on the other hand, this gives other researchers — and us of course — basic software components on which they — and we — can build other applications.

5.2. GNU MPFR

Participants: Sylvain Chevillard, Paul Zimmermann [contact].

GNU MPFR is one of the main pieces of software developed by the CARAMEL team. Since end 2006, with the departure of Vincent Lefèvre to ENS Lyon, it has become a joint project between CARAMEL and the ARÉNAIRE project-team (INRIA Grenoble - Rhône-Alpes). GNU MPFR is a library for computing with arbitrary precision floating-point numbers, together with well-defined semantics, and is distributed under the LGPL license. All arithmetic operations are performed according to a rounding mode provided by the user, and all results are guaranteed correct to the last bit, according to the given rounding mode.

Several software systems use GNU MPFR, for example: the GCC and GFORTRAN compilers; the SAGE computer algebra system; the KDE calculator Abakus by Michael Pyne; CGAL (Computational Geometry Algorithms Library) developed by the Geometrica project-team (INRIA Sophia Antipolis - Méditerranée); Gappa, by Guillaume Melquiond; Sollya, by Sylvain Chevillard, Mioara Joldeş and Christoph Lauter; Genius Math Tool and the GEL language, by Jiri Lebl; Giac/Xcas, a free computer algebra system, by Bernard Parisse; the iRRAM exact arithmetic implementation from Norbert Müller (University of Trier, Germany); the Magma computational algebra system; and the Wcalc calculator by Kyle Wheeler.

The main developments in 2010 were the release of version 3.0.0 (the "boudin aux pommes" release) in June, and the publication of an article with Kaveh Ghazi, one of the GCC developers, in *Computing in Science and Engineering* [7], which explains why and how to use arbitrary precision floating-point arithmetic. The main changes in GNU MPFR 3.0.0 are the following: GNU MPFR is now distributed under the LGPL v3+ license (instead of v2.1+ previously), a new rounding mode "away" has been added, a few new functions have been added, and trigonometric functions with large precisions are much faster.

In 2010, tools have been added by Sylvain Chevillard into the development repository of GNU MPFR, that allow one to automatically generate code for the evaluation of hypergeometric functions (such as, e.g., Bessel functions, erf, Airy functions). Tools have also been added for the automatic tuning of functions when the complexity depends on both the precision and the point of evaluation.

Last but not least, we mention a nice application of GNU MPFR, made outside the CARAMEL team. Denis Roegel (LORIA, Nancy, France) has done a huge work of reconstructing historical mathematical tables, using the GNU MPFR library to compute correct entries. His work named LOCOMAT (The LORIA Collection of Mathematical Tables) is available from http://www.loria.fr/~roegel/locomat.html.

5.3. MPC

Participant: Paul Zimmermann [contact].

MPC is a floating-point library for complex numbers, which is developed on top of the GNU MPFR library, and distributed under the LGPL license. It is co-written with Andreas Enge (LFANT project-team, INRIA Bordeaux - Sud-Ouest). A complex floating-point number is represented by $x + iy$, where x and y are real floating-point numbers, represented using the GNU MPFR library. The MPC library provides correct rounding on both the real part x and the imaginary part y of any result. MPC is used in particular in the TRIP celestial mechanics system developed at IMCCE (*Institut de Mécanique Céleste et de Calcul des Éphémérides*), and by the Magma computational number theory system.

A new version, MPC 0.8.1 (Dianthus deltoides), was released in December 2009, and MPC 0.8.2 was released in May 2010, with a major speedup in the integer power function `mpc_pow_ui`. Since version 4.5 of GCC, released in May 2010, GCC requires MPC to compute constant complex expressions at compile-time (constant folding), like it requires GNU MPFR since GCC 4.3. Also, thanks to some work done by Philippe Théveny during the MPtools project, MPC is now an optional Sage package.

5.4. GMP-ECM

Participants: Romain Cosset, Julie Feltin, Pierrick Gaudry, Alexander Kruppa, Paul Zimmermann [contact].

GMP-ECM is a program to factor integers using the Elliptic Curve Method. Its efficiency comes both from the use of the GMP library, and from the implementation of state-of-the-art algorithms. GMP-ECM contains a library (LIBECM) in addition to the binary program (ECM). The binary program is distributed under GPL, while the library is distributed under LGPL, to allow its integration into other non-GPL software. The Magma computational number theory software and the SAGE computer algebra system both use LIBECM.

In 2010, GMP-ECM 6.3 has been released. Also, during an internship of 6 weeks, Julie Feltin did implement some portable C-code for stage 1 of ECM which is independent from the GMP library, and can thus be easily ported to a GPU (Graphical Processing Unit). The multiple-precision numbers are stored into double-precision floating-point registers instead of integer registers, storing 52 bits per register instead of 64 bits. This code is available in the GMP-ECM svn repository.

5.5. Finite fields

Participants: Pierrick Gaudry, Emmanuel Thomé [contact].

$\texttt{mp}\mathbb{F}_q$ is (yet another) library for computing in finite fields. The purpose of $\texttt{mp}\mathbb{F}_q$ is not to provide a software layer for accessing finite fields determined at runtime within a computer algebra system like Magma, but rather to give a very efficient, optimized code for computing in finite fields precisely known at *compile time*. $\texttt{mp}\mathbb{F}_q$ is not restricted to a finite field in particular, and can adapt to finite fields of any characteristic and any extension degree. However, one of the targets being the use in cryptology, $\texttt{mp}\mathbb{F}_q$ somehow focuses on prime fields and on fields of characteristic two.

$\texttt{mp}\mathbb{F}_q$'s ability to generate specialized code for desired finite fields differentiates this library from its competitors. The performance achieved is far superior. For example, $\texttt{mp}\mathbb{F}_q$ can be readily used to assess the throughput of an efficient software implementation of a given cryptosystem. Such an evaluation is the purpose of the "EBats" benchmarking tool[1]. $\texttt{mp}\mathbb{F}_q$ entered this trend in 2007, establishing reference marks for fast elliptic curve cryptography: the authors improved over the fastest examples of key-sharing software in genus 1 and 2, both over binary fields and prime fields. These timings are now comparison references for other implementations [32].

The library's purpose being the *generation* of code rather than its execution, the working core of $\texttt{mp}\mathbb{F}_q$ consists of roughly 18,000 lines of Perl code, which generate most of the `C` code. $\texttt{mp}\mathbb{F}_q$ is distributed at http://mpfq.gforge.inria.fr/.

The $\texttt{mp}\mathbb{F}_q$ library has undergone no change in 2010, but has been used in several new research articles as an implementation reference [30], or also as a back-end for fast arithmetic [27].

5.6. gf2x

Participants: Richard Brent, Pierrick Gaudry, Emmanuel Thomé [contact], Paul Zimmermann.

GF2X is a software library for polynomial multiplication over the binary field, developed together with Richard Brent (Australian National University, Canberra, Australia). There are implementations of various algorithms corresponding to different degrees of the input polynomials. In the case of polynomials that fit into one or two machine-words, the schoolbook algorithm has been improved and implemented using SSE instructions for maximum speed. For small degrees, we switch to Karatsuba's algorithm and then to Toom-Cook's algorithm. These have been implemented using the most recent improvements. Finally, for very large degrees one has to switch to Fourier-transform based algorithms, namely Schönhage's or Cantor's algorithm. In order to choose between these two asymptotically fast algorithms, we have implemented and compared them. The GF2X package is distributed and maintained. It is available from http://gforge.inria.fr/projects/gf2x/. The software library NTL, as of version 5.5, can be configured to use GF2X as an auxiliary package for best performance. Documentation on the link between NTL and GF2X can be obtained from http://www.shoup.net/ntl/doc/tour-gf2x.html. Work on GF2X was part of the ANC associate team (see below).

[1] http://www.ecrypt.eu.org/ebats/

In November 2010, version 1.0 of GF2X has been released under the GNU General Public License. The main improvement to GF2X is the support of the new PCLMULQDQ on modern Intel microprocessors.

5.7. CADO-NFS

Participants: Jérémie Detrey, Pierrick Gaudry, Alexander Kruppa, Lionel Muller, Thomas Prest, Emmanuel Thomé [contact], Antonio Vera, Paul Zimmermann.

CADO-NFS is a program to factor integers using the Number Field Sieve algorithm (NFS), developped in the context of the ANR-CADO project (November 2006 to January 2010).

NFS is a complex algorithm which contains a large number of sub-algorithms. The implementation of all of them is now complete, but still leaves some places to be improved. Compared to existing implementations, the CADO-NFS implementation is already a reasonable player. Several factorizations have been completed using our implementations.

Since 2009, the source repository of CADO-NFS is publicly available for download. On December 10, 2010, the 1.0 version of CADO-NFS has been released. Several improvements to the program have been obtained, in practically all areas of the program. In particular, the polynomial selection code described by Thorsten Kleinjung at the CADO workshop in 2008 has been implemented in CADO-NFS; it is not yet used in production, but extensive experiments, in particular on RSA-768, have shown that it yields quite good polynomials in very short time. Also, Thomas Prest has implemented polynomial selection code for two non-linear polynomials. Some tools written for the factorization of RSA-768 have been integrated into CADO-NFS by Lionel Muller. Overall, CADO-NFS keeps improving its competitivity over alternative code bases.

The largest factorizations performed by CADO-NFS in 2010 are a 217-digit integer with Special NFS (1000 days of computation on one core) and a 166-digit integer with General NFS (700 days).

5.8. Hardware implementations of Shabal

Participants: Jérémie Detrey [contact], Pierrick Gaudry.

In collaboration with Karim Khalfallah (SGDSN/ANSSI), Jérémie Detrey and Pierrick Gaudry have developed FPGA implementations of the SHA-3 hash function candidate Shabal [28]. Along with a reference implementation, two lightweight architectures targeted at modern Xilinx FPGAs were designed. On top of being among the smallest implementations of the literature, they also achieve the best throughput-per-area ratio among all SHA-3 candidates [10].

The 3,000 lines of VHDL describing these three implementations were publicly released under the GNU LGPL and are available at http://hwshabal.gforge.inria.fr/.

5.9. AVIsogenies

Participants: Gaëtan Bisson [contact], Romain Cosset, Damien Robert.

AVISOGENIES (Abelian Varieties and Isogenies) is a Magma package for working with abelian varieties, with a particular emphasis on explicit isogeny computation; it has been publicly released under the LGPLv2+ license in 2010.

Its prominent feature is the computation of (ℓ, ℓ)-isogenies between Jacobian varieties of genus-2 hyperelliptic curves over finite fields of characteristic coprime to 2ℓ; practical runs have involved values of ℓ in the hundreds.

Internally, it implements many routines for handling points of dimension-2 abelian varieties $\mathcal{A}$ represented by theta functions; more specifically, it can:

1. construct a field extension where geometric points of maximal isotropic subgroups of $\mathcal{A}[\ell]$ are defined;
2. compute a symplectic basis of $\mathcal{A}[\ell]$ over that extension;
3. list all rational maximal isotropic subgroups;
4. convert the basis of each such subgroup $\mathcal{H}$ from Mumford to theta coordinates of level 2;
5. enumerate other points of $\mathcal{H}$ via differential additions;
6. apply the level-changing formula to recover level-2 theta constants for $\mathcal{A}/\mathcal{H}$;
7. deduce the absolute Igusa invariants of the isogenous variety $\mathcal{A}/\mathcal{H}$.

These routines are put together in a wrapper function that, on input an abelian variety and a prime ℓ, returns the list of rationally (ℓ,ℓ)-isogenous varieties. Previously, only the case $\ell = 2$ (known as Richelot isogenies) was available in software packages.

In addition, there are procedures for exploring and drawing isogeny graphs, and for computing various complex-multiplication-related structures, such as Shimura's gothic C group.

The package can be obtained at http://avisogenies.gforge.inria.fr/.

6. New Results

6.1. Integer factorization

During his two-month internship in June and July, Thomas Prest developed with Paul Zimmermann a new algorithm for the polynomial selection in the Number Field Sieve (NFS). This algorithm produces two non-linear polynomials, extending Montgomery's "two quadratics" method. For degree 3, it gives two skewed polynomials with resultant $O(N^{5/4})$, which improves on Williams $O(N^{4/3})$ recent result. The paper has been submitted to a special issue of the Journal of Symbolic Computation in honour of Joachim von zur Gathen [21].

The completion of the record RSA-768 factorization has led to three publications. A paper published in Crypto 2010 gives a general presentation of the record and the technology involved [13]. On the specific aspects of linear algebra on computer grids, a paper has been published in the Grid 2010 conference describing how the block Wiedemann algorithm has been successfully used on a shared resource [14]. Finally, an article describing how the different resources have been used for sieving has been accepted for publication in Cluster Computing [8].

6.2. Implementation of cryptographic pairings

First, the paper submitted in 2009 at IEEE Transactions on Computers by Jean-Luc Beuchat, Jérémie Detrey, Nicolas Estibals, Eiji Okamoto, and Francisco Rodríguez-Henríquez has been accepted [3]. This work presents coprocessors for computing the Tate pairing over supersingular curves in characteristics 2 and 3 based on a fast and parallel implementation of a Karatsuba multiplier for performing the arithmetic over the base field. Although they do not scale to the recommended security level of 128 bits, these coprocessors hold the current speed record for pairing computation, at 17 μs for a 109-bit security pairing.

Given that this last approach does not scale to the 128-bit security level mainly because of the increasing size of the base field, Nicolas Estibals suggested to use fields of moderately composite extension degree. Then it is possible to handle elements in smaller subfields and have an efficient arithmetic representation of the field of definition. Even though the Weil descent applies on such curves, the Gaudry–Hess–Smart attack is shown not being effective for the selected parameters. Thanks to those methods, a compact FPGA accelerator for pairing computation at the 128-bit security level has been designed and presented at the Pairing 2010 conference [12].

Finally, in collaboration with Diego F. Aranha (UNICAMP, Brazil) and Jean-Luc Beuchat (University of Tsukuba, Japan), Jérémie Detrey and Nicolas Estibals presented a new pairing algorithm over a supersingular genus-2 hyperelliptic curve in characteristic two. Based on the optimal pairing technique [33] applied to the action of the Verschiebung (*i.e.*, as in the η_T pairing case [26]), this method shortens the pairing computation algorithm by 33% with respect to previous works. This approach was validated by software and hardware (FPGA) implementations, yielding timing results very close to pairings over elliptic curves. A paper describing this work was submitted to EUROCRYPT 2011 [19].

6.3. Low-resource hardware implementation of Shabal

On July 24, 2009, the NIST (National Institute of Standards and Technology) announced the list of the fourteen hash function candidates accepted to the second round of the SHA-3 competition [31]. Participating in the vast effort undertaken by the research community to assess the security and performances of those candidates, Jérémie Detrey, Pierrick Gaudry, and Karim Khalfallah (SGDSN/ANSSI) designed a low-cost implementation of Shabal [28] on modern Xilinx FPGAs. This design, benefiting from the embedded shift-registers available in the FPGA, achieves the best throughput-per-area ratio among all SHA-3 candidates [10]. (On December 9, the five SHA-3 candidate algorithms for the third and final round were announced by NIST: BLAKE, Grøstl, JH, Keccak and Skein.)

6.4. Enumeration of short vectors on FPGA

The security of several public-key cryptosystems relies of the hardness of solving the Shortest and Closest Vector Problems (SVP and CVP, respectively) in high-dimensional lattices. Using the blockwise Korkine–Zolotarev algorithm for basis reduction, these problems can be broken down to many "small" instances of SVP in lattices of lower dimension (typically, 40 to 80), which can then be solved using the Kannan–Fincke–Pohst (KFP) algorithm for enumerating short vectors.

In this work, Jérémie Detrey, along with Guillaume Hanrot, Xavier Pujol, and Damien Stehlé (Arénaire project-team, LIP, ENS Lyon), studied how this KFP algorithm could be ported to FPGAs so as to benefit from the fine-grained parallelism inherent to these architectures [11]. In order to do so, the enumeration algorithm has to be tailored to allow for an efficient usage of the FPGA resources. Among other things, this entails resorting to fixed-point instead of floating-point arithmetic, while ensuring correctness of the result via a careful error analysis.

6.5. Multiple-precision arithmetic

Sylvain Chevillard worked on the development of multiple-precision floating-point code for the GNU MPFR library. His work was twofold. First, he implemented two algorithms for the evaluation of the Airy Ai function with correct rounding in arbitrary precision. The first algorithm is a naive evaluation of the series with a step-by-step computation of the coefficients; neglecting logarithmic factors, its theoretical bit-complexity is $\mathcal{O}(p\,M(p))$ where p is the working precision and $M(p)$ is the cost of a multiplication between two numbers of p bits. The second algorithm is based on a baby step/giant step strategy; its theoretical complexity is $\mathcal{O}(p^{1/2}\,M(p))$. Though the second algorithm is asymptotically better than the first one, the practice shows that the first algorithm is competitive for low precisions. Sylvain Chevillard developed benchmark algorithms to experimentally study which of both algorithms is the fastest. This allows one to define thresholds where one should switch between one algorithm and the other, depending on the desired precision and the point of evaluation. He also designed a tuning program that automatically finds suitable thresholds for the specific architecture on which it is run. Sylvain Chevillard also compared his implementation with the implementation provided by Maple, Mathematica and Sage: the experimental results show that they all use the naive algorithm. The implementation provided by Sylvain Chevillard in GNU MPFR is the fastest both theoretically and experimentally.

The second part of this work is an attempt to automate parts of the development of functions in the GNU MPFR library. Many usual mathematical functions $f(x)$ have a Taylor series $\sum_{n=0}^{\infty} a_n\, x^n$ where (a_n) satisfies a finite linear recurrence with polynomial coefficients: $\forall n, p_d(n)\, a_{n+d} = \sum_{i=0}^{d-1} p_i(n)\, a_i$. In order to evaluate the series, one needs to accurately evaluate the first coefficients $a_0, \cdots, a_{d-1}$ and then evaluate the other coefficients by means of the recurrence, up to a sufficient order and using a suitable precision. Often there exist efficient formulas for evaluating the constants $a_0, \cdots, a_d$. Sylvain Chevillard designed and implemented an algorithm that automatically generates code for the evaluation of such constant formulas in arbitrary precision and with a rigorous error bound [20]. This implementation is available in the development repository of the Sollya software tool. Sylvain Chevillard also developed a tool that generates a large part of the code for evaluating $f(x)$ with correct rounding in arbitrary precision, in the case when the recurrence is of the form $p_d(n)\, a_{n+d} = p_0(n)\, a_n$. This tool generalizes what has been done for the Airy Ai function.

Following a question from Steven Galbraith, Richard Brent and P. Zimmermann designed a new subquadratic algorithm for computing the Jacobi symbol. This algorithm works by the least significant bits, and is thus easier to implement and prove correct, since no "fixup step" is needed. The algorithm was published in the proceedings of the ANTS-IX conference [9], and an implementation in GMP is available from the authors' web site.

With Guillaume Melquiond (Proval project-team, INRIA Saclay), P. Zimmermann worked on the numerical approximation of the Masser-Gramain constant, following some work of Gramain and Weber in 1985. Preliminary results tend to disprove a conjecture of Gramain, and would enable one to determine three decimal digits of that constant after the decimal point (only one was known before). This work will be completed in 2011.

6.6. Proving the complexity of computing endomorphism rings

Subsequent to joint work with Andrew V. Sutherland on computing endomorphism rings of ordinary elliptic curves $\mathcal{E}$ over finite fields [4], Gaëtan Bisson has been working on rigorously proving a subexponential running time bound for such computations. The main ingredient remains the action of the class group $\mathrm{cl}\,\mathcal{O}$ on the isogeny graph of $\mathcal{E}$, where $\mathcal{O}$ is the imaginary quadratic order isomorphic to the endomorphism ring of $\mathcal{E}$, but the rigorous proof involves several new constructions:

- a more generic "order lattice ascending" procedure;
- a modified Hafner–McCurley method for finding short relations in class groups;
- showing that the structure of $\mathrm{cl}\,\mathcal{O}$ determines $\mathcal{O}$ in most cases;
- describing a simple fall-back method for the (rare) other cases.

The proof then rests on two assumptions: the extended Riemann hypothesis (ERH), and a conjectural bound on the diameter of a minimal generating set of relations for $\mathrm{cl}\,\mathcal{O}$. The latter needs to be further studied in order to determine whether it is independent from the ERH. These results are expected to be made public in the next few months.

6.7. Pollard-rho type algorithm for generic groups

In studying the above-mentioned bound, Gaëtan Bisson has again been collaborating with Andrew V. Sutherland on a novel, Pollard-rho type algorithm for finding relations in generic groups. On input a sequence S of elements of a generic group G satisfying $\#S > 2\log_2 \#G$, and a target element $z \in G$, the algorithm finds a subsequence of S that adds up to z. For random subsequences S, its runtime can be proven to be $\sqrt{\#G}$ up to logarithmic factors, and it requires virtually no memory. Applications notably include searching for an isogeny of degree polynomial in $\log q$ between two ordinary elliptic curves with the same endomorphism ring defined over the finite field with q elements, with only a polynomial space complexity. These results are expected to be made public in the next few months.

6.8. Computation of Isogenies between abelian varieties

David Lubicz and Damien Robert have written an article [15] about the explicit computation of isogenies using theta coordinates. The algorithm takes for input an abelian variety A and a basis of a maximal isotropic subgroup $K \subset A[\ell]$ written in theta coordinates of level n, and outputs the isogeny $A \to A/K$ where A/K is described by theta coordinates of level $n\ell$.

The theta functions of level n are coordinates on abelian varieties. For arithmetic reasons, we usually use $n = 2$ or 4, however for computing ℓ-isogenies with the method of Lubicz and Robert we need to use theta functions of level $n\ell$. Romain Cosset and Damien Robert found formulæ to change the level of theta functions from n to $n\ell$. Combined with the above method, this gives the first formulæ to compute ℓ-isogenies between abelian varieties. An article is being written about these results. The formulæ for changing level can also be found in [2].

In the particular case of Jacobian of hyperelliptic curves, Romain Cosset gave explicit methods to transfer points between the classical representation with Mumford's coordinates and the theta functions. This is a generalisation of the work of Van Wamelen.

6.9. Pairing using theta functions

David Lubicz and Damien Robert have described an algorithm [25] to compute the Weil and Tate pairings in theta coordinates. This algorithm has the advantage that it is available on all abelian varieties. Moreover over the Kummer surfaces of hyperelliptic curves of genus 2, it makes use of the fast formula for the arithmetic of theta functions provided by Gaudry.

6.10. Point counting in genus 2

In 2009, Pierrick Gaudry and Éric Schost have run a large-scale point-counting computation in order to construct a genus-2 curve over a prime field suitable for cryptographical use. In 2010, they wrote an article describing in detail many improvements that were designed for this computation; they also proved some phenomenons that were experimentally observed. The article [24] has been submitted.

6.11. National Initiatives

6.11.1. ANR CADO (Crible algébrique, Distribution, Optimisation)

Participants: Pierrick Gaudry, Alexander Kruppa, Lionel Muller, Thomas Prest, Emmanuel Thomé, Antonio Vera, Paul Zimmermann.

The ANR project CADO, from "programme blanc" has finished at the end of January 2010. Its purpose was to study the number field sieve factoring algorithm.

The most visible results are:

- a complete implementation of the NFS algorithm: CADO-NFS (see the software section);
- the PhD thesis of A. Kruppa, defended in January 2010 [1];
- the participation to the RSA-768 record computation (completed in December 2009).

6.11.2. ANR RAPIDE (Design and analysis of stream ciphers dedicated to constrained environments)

Participant: Marion Videau.

The project from "programme Sécurité Et Informatique 2006" involves the team together with the SECRET (former CODES) project-team, the XLIM lab from the University of Limoges and the CITI lab from INSA-Lyon. It has been running since January 2007 and ended in December 2010.

The research project consists in the study and analysis, both from theoretical and practical points of view, of existing stream ciphers and new designs based on non-linear feedback shift registers.

Despite the departure of Marion Videau (on secondment to the cryptographic lab of the Agence Nationale de la Sécurité des Systèmes d'Information), the coordination tasks are held by her from the team side.

6.11.3. ANR DEMOTIS (Collaborative Analysis, Evaluation and Modelling of Health Information Technology)

Participant: Marion Videau.

The project from "programme ARPEGE" involves three INRIA project-teams as a single partner (SMIS, SECRET and CARAMEL) together with colleagues from CECOJI (CNRS) and the company Sopinspace. It has been running from January 2009 and will continue until the end of 2011.

The project experiments new methods for the multidisciplinary design of large information systems that have to take into account legal, social and technical constraints. Its main field of application is personal health information systems.

6.11.4. ANR CHIC (Courbes Hyperelliptiques, Isogénies, Comptage)

Participants: Pierrick Gaudry, Emmanuel Thomé, Gaëtan Bisson, Romain Cosset, Damien Robert.

The team has obtained a financial support from the ANR ("programme blanc") for a project, common with colleagues from IRMAR (Rennes) and IML (Marseille). The ANR CHIC grant covers the period 09/2009 to 08/2012. The purpose of this ANR project is the study of several aspects of curves in genus 2, with a very strong focus on the computation of explicit isogenies between Jacobians.

This ANR project has been an important source of motivation for both permanent researchers and PhD students, giving notably PhD students the opportunity to meet interested colleagues on a regular basis. In particular the article by Lubicz and Robert [15] is central to the topics of the ANR CHIC project.

6.12. European Initiatives

6.12.1. PHC application with EPFL

The team applied for a PHC Germaine de Staël grant in collaboration with the LACAL team from EPFL (Lausanne, Switzerland), over the period 2011-2012. The proposal has been accepted. This collaboration will be focused on integer factorization and discrete logarithms. We plan to organise several themed workshop on these topics.

6.13. International Initiatives

6.13.1. Collaboration with ANU

Participants: Shi Bai, Richard Brent, Judy-anne Osborn, Srinivas Subramanya, Paul Zimmermann.

In the context of the "associate team" ANC (Algorithms, Numbers, Computers), which started in 2008 and ended in 2010 (http://www.loria.fr/~zimmerma/anc.html), between the CARAMEL team and the team of Richard Brent at the Australian National University (ANU), several visits were organized in 2010: P. Zimmermann visited ANU for two weeks in May, where he did attend the workshop on maximal determinant matrices; Shi Bai visited LORIA for one month in July, he attended the ANTS-IX conference and worked with P. Zimmermann on polynomial selection for NFS; Srinivas Subramanya visited LORIA for one month in September, where he worked on efficient modular arithmetic on GPU.

The ANC associate team was formally evaluated in November 2010 by two external reviewers and the INRIA COST-RI team.

One of the main results of the associate team is the book "Modern Computer Arithmetic", whose paper version is published by Cambridge University Press, and whose electronic version will remain freely available for download [16].

7. Dissemination

7.1. Animation of the scientific community

7.1.1. Caramel seminar

Twenty-two speakers were invited to our seminar in 2010: Pascal Molin, Fabien Laguillaumie, Osmanbey Uzunkol, Wouter Castryck, Luca De Feo, Romain Cosset, Peter Schwabe, Mioara Joldeş, Jean-François Biasse, Francesco Sica, Paul Zimmermann, Xavier Goaoc, Fabrice Rouillier, Julie Feltin, Peter Montgomery, Thomas Prest, Louise Huot, Marcelo Kaihara, Mehdi Tibouchi, Vanessa Vitse, Guillaume Batog, and Sylvain Collange.

7.1.2. Conference organization

Pierrick Gaudry and Emmanuel Thomé, together with Anne-Lise Charbonnier from the "comité colloques" of INRIA Nancy - Grand Est, have organised the ANTS-IX conference[2] [18], at LORIA in July 2010. The conference has been a success, with 140 participants. All the members of the team have contributed to making ANTS-IX a major event for researchers in algorithmic number theory.

In 2011, the team will also organise the ECC 2011 workshop. A significant amount of funding has already been secured for this event.

7.2. Committees memberships

- Jérémie Detrey was a member of the "Comité de Sélection" for an Associate Professor position in computer science (section 27) at ENSI Caen.
- Pierrick Gaudry was referee for the PhD thesis of Tony Ezome (Toulouse III) and Jean-François Biasse (École polytechnique). He was in the defense committee of the PhD thesis of Damien Robert (LORIA). He was in the program committee of the SCC 2010 conference (London), of the ECC 2010 workshop (Redmond) and of the Eurocrypt 2011 conference (Tallinn, Estonia). He served in the INRIA hiring committee for CR1 at Rocquencourt. He is member of the "Équipe de direction" of the LORIA.
- Marion Videau is member of the program committee of the FSE 2011 conference (Fast Software Encryption), is member of the scientific committee of the CCA seminar (Codage, Cryptologie, Algorithmes) and of the "journées C2".
- Paul Zimmermann was head of the hiring committee for junior researchers (CR2 and CR1) at INRIA Nancy - Grand Est; he is also head of the "Comité Colloques" of INRIA Nancy - Grand Est; he was member of the PhD thesis jury of Alexander Kruppa and Willemien Ekkelkamp.

7.3. Vulgarization

P. Gaudry gave a one-hour vulgarisation talk about integer factorization during the ceremony of the "Olympiades de mathématiques" at LORIA.

Together with Richard Brent, P. Zimmermann was invited in 2009 to write an article for the Notices of the American Mathematical Society. This article, describing their hunt for primitive trinomials over $\mathrm{GF}(2)$, was finalized in 2010 [5].

Together with 9 colleagues, P. Zimmermann has written a book in French about the Sage computer algebra system [17]. This book of 315 pages was published online in July under a Creative Commons license, has been downloaded more than 2000 times during the first week, and since that time about 300 times per week. Discussions are in progress with commercial editors to publish a paper version. P. Zimmermann also gave a 1-hour talk on floating-point computations in a mathematics course at "lycée Jeanne d'Arc" in Nancy to students of "seconde" level (about 15 years old).

[2] http://www.ants9.org/

7.4. Invited Conferences

P. Gaudry gave two one-hour talks at the workshop "Counting points: theory, algorithms and practice" in the CRM center of Montreal and at the workshop "Workshop on computational number theory and arithmetic geometry" in Leuven.

P. Zimmermann gave a talk on Sage at the Plume one-day workshop "Les alternatives libres aux outils propriétaires de maths" (Paris) in February, a talk on the factorization of RSA-768 at the ANSSI in May, an invited talk on the factorization of RSA-768 at the Workshop on Tools for Cryptanalysis (Royal Holloway, UK) in June, an invited talk on the GNU MPFR library at the Third International Congress on Mathematical Software (Kobe, Japan) in September, an invited talk on floating-point computations at the "Leçons de Mathématiques d'Aujourd'hui" at the University of Bordeaux 1 in October, and an invited talk on mathematics and cryptographic at a colloquium on mathematics and society organized by the "Académie Lorraine des Sciences" in November (Nancy).

7.5. Teaching

- Jérémie Detrey gave a twelve-hour introductive course on cryptography at ÉSIAL (*École Supérieure d'Informatique et Applications de Lorraine*, Nancy).
- Jérémie Detrey gave a two-hour lecture in *licence professionnelle* at IUT Charlemagne (Nancy) on the topic of security.
- Pierrick Gaudry gave 30 hours of Master 1 courses at Université Henri Poincaré on the topic of cryptology.
- Emmanuel Thomé gave a fifteen-hour course on algorithmic number theory at École Normale Supérieure (Paris).
- Emmanuel Thomé gave a six-hour course at MPRI (Master Parisien de Recherche en Informatique).
- Emmanuel Thomé has been a jury member for the Agrégation Externe de Mathématiques examination.

8. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] A. KRUPPA. *Améliorations de la multiplication et de la factorisation d'entier*, Université Henri Poincaré - Nancy I, January 2010, http://hal.inria.fr/tel-00477005/en.

[2] D. ROBERT. *Fonctions thêta et applications à la cryptographie*, Université Henri Poincaré - Nancy I, July 2010, http://hal.inria.fr/tel-00528942/en.

Articles in International Peer-Reviewed Journal

[3] J.-L. BEUCHAT, J. DETREY, N. ESTIBALS, E. OKAMOTO, F. RODRÍGUEZ-HENRÍQUEZ. *Fast architectures for the η_T pairing over small-characteristic supersingular elliptic curves*, in "IEEE Transactions on Computers", July 2010 [*DOI :* 10.1109/TC.2010.163], http://hal.inria.fr/inria-00424016/en.

[4] G. BISSON, A. V. SUTHERLAND. *Computing the endomorphism ring of an ordinary elliptic curve over a finite field*, in "Journal of Number Theory", 2010 [*DOI :* 10.1016/J.JNT.2009.11.003], http://hal.inria.fr/inria-00383155/en.

[5] R. BRENT, P. ZIMMERMANN. *The Great Trinomial Hunt*, in "Notices of the AMS", 2010, http://hal.inria.fr/inria-00443797/en.

[6] R. COSSET. *Factorization with genus 2 curves*, in "Mathematics of Computation / Mathematics of Computation of the American Mathematical Society", 2010, vol. 79, p. 1191-1208 [*DOI :* 10.1090/S0025-5718-09-02295-9], http://hal.inria.fr/inria-00384128/en.

[7] K. GHAZI, V. LEFÈVRE, P. THÉVENY, P. ZIMMERMANN. *Why and How to Use Arbitrary Precision*, in "Computing in Science and Engineering", 2010, vol. 12, n° 3, p. 62-65 [*DOI :* 10.1109/MCSE.2010.73], http://hal.inria.fr/inria-00543927/en.

[8] T. KLEINJUNG, J. BOS, A. LENSTRA, D. A. OSVIK, K. AOKI, S. CONTINI, J. FRANKE, E. THOMÉ, P. JERMINI, M. THIÉMARD, P. LEYLAND, P. MONTGOMERY, A. TIMOFEEV, H. STOCKINGER. *A Heterogeneous Computing Environment to Solve the 768-bit RSA Challenge*, in "Cluster Computing", 2010, http://hal.inria.fr/inria-00535765/en.

International Peer-Reviewed Conference/Proceedings

[9] R. BRENT, P. ZIMMERMANN. *An O(M(n) log n) algorithm for the Jacobi symbol*, in "9th Algorithmic Number Theory Symposium - ANTS IX", France Nancy, G. HANROT, F. MORAIN, E. THOMÉ (editors), Lecture Notes in Computer Science, Springer Verlag, July 2010, vol. 6197, p. 83-95, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-14518-6_10], http://hal.inria.fr/inria-00447968/en.

[10] J. DETREY, P. GAUDRY, K. KHALFALLAH. *A low-area yet performant FPGA implementation of Shabal*, in "17th International Workshop on Selected Areas in Cryptography, SAC 2010", Canada Waterloo, A. BIRYUKOV, G. GONG, D. STINSON (editors), Lecture Notes in Computer Science, 2010, http://hal.inria.fr/inria-00498705/en.

[11] J. DETREY, G. HANROT, X. PUJOL, D. STEHLÉ. *Accelerating lattice reduction with FPGAs*, in "First International Conference on Cryptology and Information Security in Latin America (LATINCRYPT'10)", Mexico Puebla, M. ABDALLA, P. S. L. M. BARRETO (editors), Lecture Notes in Computer Science, August 2010, vol. 6212, p. 124-143 [*DOI :* 10.1007/978-3-642-14712-8_8], http://hal.inria.fr/inria-00539929/en.

[12] N. ESTIBALS. *Compact hardware for computing the Tate pairing over 128-bit-security supersingular curves*, in "Pairing 2010 – 4th International Conference on Pairing-Based Cryptography", Japan Yamanaka Hot Spring, M. JOYE, A. MIYAJI, A. OTSUKA (editors), Lecture Notes in Computer Science, December 2010, vol. 6487, p. 397-416 [*DOI :* 10.1007/978-3-642-17455-1], http://hal.inria.fr/inria-00539926/en.

[13] T. KLEINJUNG, K. AOKI, J. FRANKE, A. LENSTRA, E. THOMÉ, J. BOS, P. GAUDRY, A. KRUPPA, P. MONTGOMERY, D. A. OSVIK, H. TE RIELE, A. TIMOFEEV, P. ZIMMERMANN. *Factorization of a 768-bit RSA modulus*, in "CRYPTO 2010", United States Santa Barbara, T. RABIN (editor), Lecture Notes in Computer Science, Springer Verlag, 2010, vol. 6223, p. 333-350, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-14623-7_18], http://hal.inria.fr/inria-00444693/en.

[14] T. KLEINJUNG, L. NUSSBAUM, E. THOMÉ. *Using a grid platform for solving large sparse linear systems over GF(2)*, in "11th ACM/IEEE International Conference on Grid Computing (Grid 2010)", Belgium Brussels, October 2010, http://hal.inria.fr/inria-00502899/en.

[15] D. LUBICZ, D. ROBERT. *Efficient pairing computation with theta functions*, in "ANTS IX - Algorithmic Number Theory 2010", France Nancy, G. HANROT, F. MORAIN, E. THOMÉ (editors), Lecture Notes in Computer Science, Springer-Verlag, 2010, vol. 6197, p. 251-269, The original publication is available at www.springerlink.com [*DOI :* 10.1007/978-3-642-14518-6_21], http://hal.inria.fr/hal-00528944/en.

Scientific Books (or Scientific Book chapters)

[16] R. BRENT, P. ZIMMERMANN. *Modern Computer Arithmetic*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2010, vol. 18, http://hal.inria.fr/inria-00424347/en.

[17] A. CASAMAYOU, G. CONNAN, T. DUMONT, L. FOUSSE, F. MALTEY, M. MEULIEN, M. MEZZAROBBA, C. PERNET, N. THIÉRY, P. ZIMMERMANN. *Calcul mathématique avec Sage*, none (electronic version only), 2010, http://hal.inria.fr/inria-00540485/en.

Books or Proceedings Editing

[18] G. HANROT, F. MORAIN, E. THOMÉ (editors). *Algorithmic Number Theory. 9th. International Symposium, ANTS-IX. Nancy, France, July 2010. Proceedings*, Lecture Notes in Computer Science, Springer-Verlag, July 2010, vol. 6197 [*DOI :* 10.1007/978-3-642-14518-6], http://hal.inria.fr/inria-00544503/en.

Research Reports

[19] D. ARANHA, J.-L. BEUCHAT, J. DETREY, N. ESTIBALS. *Optimal Eta pairing on supersingular genus-2 binary hyperelliptic curves*, IACR, November 2010, http://hal.inria.fr/inria-00540002/en.

[20] S. CHEVILLARD. *Evaluating a constant expression in multiple precision with a guaranteed error bound*, INRIA, November 2010, n^{o} RR-7443, http://hal.inria.fr/inria-00537935/en.

[21] T. PREST, P. ZIMMERMANN. *Non-Linear Polynomial Selection for the Number Field Sieve*, INRIA, 2010, http://hal.inria.fr/inria-00540483/en.

[22] A. VERA. *A Note on Integer Factorization Using Lattices*, CNRS/INRIA/Nancy Université, March 2010, http://hal.inria.fr/inria-00467590/en.

Other Publications

[23] S. CHEVILLARD, J. HARRISON, M. JOLDES, C. LAUTER. *Efficient and accurate computation of upper bounds of approximation errors*, 35 pages, http://hal.inria.fr/ensl-00445343/en.

[24] P. GAUDRY, É. SCHOST. *Genus 2 point counting over prime fields*, 2010, Preprint, http://hal.inria.fr/inria-00542650/en.

[25] D. LUBICZ, D. ROBERT. *Computing isogenies between Abelian Varieties*, 2010, 47 pages, http://hal.inria.fr/hal-00446062/en.

References in notes

[26] P. S. L. M. BARRETO, S. D. GALBRAITH, C. Ó ÉIGEARTAIGH, M. SCOTT. *Efficient pairing computation on supersingular Abelian varieties*, in "Designs, Codes and Cryptography", 2007, vol. 42, p. 239–271.

[27] D. J. BERNSTEIN, P. BIRKNER, T. LANGE, C. PETERS. *ECM using Edwards curves*, June 2010, http://cr.yp.to/papers.html#eecm.

[28] E. BRESSON, A. CANTEAUT, B. CHEVALLIER-MAMES, C. CLAVIER, T. FUHR, A. GOUGET, T. ICART, J.-F. MISARSKY, M. NAYA-PLASENCIA, P. PAILLIER, T. PORNIN, J.-R. REINHARD, C. THUILLET, M. VIDEAU. *Shabal, a Submission to NIST's Cryptographic Hash Algorithm Competition*, 2008, Preprint, http://www.shabal.com/.

[29] N. KOBLITZ. *Hyperelliptic cryptosystems*, in "J. Cryptology", 1989, vol. 1, p. 139–150.

[30] P. LONGA, C. H. GEBOTYS. *Efficient Techniques for High-Speed Elliptic Curve Cryptography*, in "CHES", S. MANGARD, F.-X. STANDAERT (editors), Lecture Notes in Computer Science, Springer, 2010, vol. 6225, p. 80-94.

[31] A. REGENSCHEID, R. PERLNER, S.-J. CHANG, J. KELSEY, M. NANDI, S. PAULU. *Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition*, National Institute of Standards and Technology, September 2009, n^{o} NISTIR 7620, http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/documents/sha3_NISTIR7620.pdf.

[32] M. SCOTT. *New record breaking implementations of ECC on quadratic extensions using endomorphisms*, September 2008, Invited talk at the ECC 2008 Conference. Utrecht, the Netherlands, Sep. 22-24, 2008.

[33] F. VERCAUTEREN. *Optimal Pairings*, in "IEEE Transactions on Information Theory", January 2010, vol. 56, n^{o} 1, p. 455–461.