Project proposal

Title	Cryptology, arithmetic: algebraic methods for better algorithms
Acronym	**CARAMBA**
Scientific leader	Emmanuel Thomé
Inria theme	Algorithmics, programming, software and architecture
Inria sub-theme	Algorithmics, Computer Algebra and Cryptology
Inria center	Nancy – Grand Est
Keywords	Cryptography, Integer factorization, Discrete logarithm, Elliptic and hyperelliptic curve cryptography, Number theory, Polynomial system solving, Computer arithmetic

CARAMBA will be a joint team with CNRS and University of Lorraine, and will be part of LORIA (Laboratoire Lorrain de Recherche en Informatique et Applications). Within LORIA, CARAMBA will be part of Department 1: “Algorithms, Computation, Image & Geometry”.

Contents

1 Team members

- **Head of project-team**:
  - Emmanuel Thomé (DR2 Inria).
- **Vice-head of project-team**:
  - Pierrick Gaudry (DR2 CNRS).
- **Staff members**:
  - Jérémie Detrey (CR1 Inria);
  - Pierre-Jean Spaenlehauer (CR1 Inria);
  - Paul Zimmermann (DR1 Inria).
- **Associate members**:
  - Luc Sanselme (Lycée Henri Poincaré);
  - Marion Videau (Univ. Lorraine, on secondment to Quarkslab since 01/2015).
- **Post-doctoral fellows and PhD students**:
  - As of 2015/09/01, 2 post-doctoral fellows (Coxon, Massierer);
  - As of 2015/09/01, 4 PhD students (Abelard, Covanov, Grémy, Labrande).

Within this document, references such as [1] correspond to publications listed in footnotes, while references such as [BBKZ15] correspond to publications involving one or several team members at the time of publication, listed in §14.

2 Overall objectives

Our research addresses the broad application domain of cryptography and cryptanalysis from the algorithmic perspective. We study all the algorithmic aspects, from the top-level mathematical background down to the optimized high-performance software implementations. Several kinds of mathematical objects are commonly encountered in our research. Some basic ones are truly ubiquitous: integers, finite fields, polynomials, real and complex numbers. We also work with more structured objects such as number fields, algebraic curves, or polynomial systems. In all cases, our work is geared towards making computations with these objects effective and fast.

The mathematical objects we deal with are of utmost importance for the applications to cryptology, as they are the background of the most widely developed cryptographic primitives, such as the RSA cryptosystem or the Diffie–Hellman key exchange. The two facets of cryptology—cryptography and cryptanalysis—are central to our research. The key challenges are the assessment of the security of proposed cryptographic primitives, through the study of the cornerstone problems, which are the integer factorization and discrete logarithm problems, as well as the optimization work in order to enable cryptographic implementations that are both efficient *and* secure.

Among the research themes we set forth in this research proposal, two are guided by the most important mathematical objects used in today's cryptography, and two others are rather guided by the technological background we use to address these problems.

- Extended NFS family. A common algorithmic framework, called the Number Field Sieve (NFS), addresses both the integer factorization problem as well as the discrete logarithm problem over finite fields. We have numerous algorithmic contributions in this context, and develop software to illustrate them.

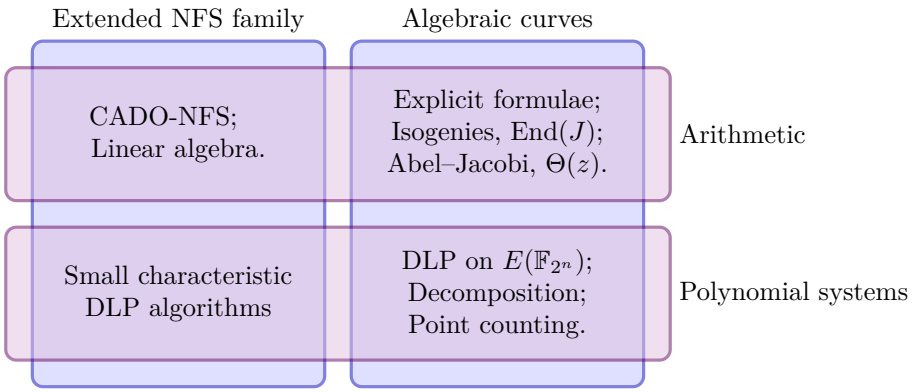


Figure 1: Proposed visual representation of the thematic organization of CARAMBA.

We plan to improve on the existing state of the art in this domain by researching new algorithms, by optimizing the software performance, and by demonstrating the reach of our software with highly visible computations.

- Algebraic curves and their Jacobians. We develop algorithms and software for computing essential properties of algebraic curves for cryptology, eventually enabling their widespread cryptographic use.

  One of the challenges we address here is point counting. In a wider perspective, we also study the link between abelian varieties over finite fields and principally polarized abelian varieties over fields of characteristic zero, together with their endomorphism ring. In particular, we work in the direction of making this link an effective one. We are also investigating various approaches for attacking the discrete logarithm problem in Jacobians of algebraic curves.

- Arithmetic. Our work relies crucially on efficient arithmetic, be it for small or large sizes. We work on improving algorithms and implementations, for computations that are relevant to our application areas.

- Polynomial systems. It is rather natural with algebraic curves, and occurs also in NFS-related contexts, that many important challenges can be represented via polynomial systems, which have structural specificities. We intend to develop algorithms and tools that, when possible, take advantage of these specificities.

As represented by Figure 1, the first two challenges above interact with the latter two, which are also research topics in their own right. Both algorithmic and software improvements are the necessary ingredients for success. The different axes of our research form thus a coherent set of research directions, where we apply a common methodology.

The CARAMBA project-team is a follow-up to the CARAMEL project-team. Beyond the change of scientific leader, several important research results (some of which are a direct consequence of our work) shaped the scientific domain somewhat differently from what it was in 2010 when CARAMEL started. New algorithms [BGJT14] have evicted some mathematical objects from the cryptographer's portfolio. We notably observed the introduction of polynomial systems as important tools for various computations related to finite fields as well as algebraic curves in cryptology. This introduction was simultaneous to the hiring of a new junior researcher (P.-J. Spaenlehauer) who has strong expertise in this area.

We consider that the impact of our research on cryptology in general owes a lot to the publication of concrete practical results. We are strongly committed to making our algorithms available as software implementations. We thus have several long-term software development projects that are, and will remain, parts of our research activity.

3 Scientific grounds

Public-key cryptography is our main application target. We are interested in the study of the cryptographic primitives that serve as a basis for the most widespread protocols.

Since the early days of public-key cryptography, and through the practices and international standards that have been established for several decades, the most widespread cryptographic primitives have been the RSA cryptosystem, as well as the Diffie–Hellman key exchange using multiplicative groups of finite fields. The level of security provided by these cryptographic primitives is related to the hardness of the underlying mathematical problems, which are integer factorization and the discrete logarithm problem. The complexity of attacking them is known to be subexponential in the public key size, and more precisely written as $L_N(1/3, c)$ for factoring an integer N, where the L notation stands for

$$L_N(\alpha, c) = \exp\left(c(1 + o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha}\right).$$

This complexity is achieved with the Number Field Sieve (NFS) algorithm and its many derivatives. This means that as the desired security level s grows, the matching public key size grows roughly like s^3. As to how these complexity estimates translate into concrete assessments and recommendations, the hard facts are definitely the computational records that are set periodically by academics, and used as key ingredients by governmental agencies emitting recommendations for the industry [1, 2].

Software for NFS is obviously the entry point to computational records. Few complete NFS implementations exist, and their improvement is of crucial importance for better assessment of the hardness of the key cryptographic primitives considered. Here, "improvement" may be understood in many ways: better algorithms (outperforming the NFS algorithm as a whole is certainly a tremendous improvement, but replacing one of its numerous substeps is one, too), better implementations, better parallelization, or better adaptation to suitable hardware. The numerous sub-algorithms of NFS strongly depend on arithmetic efficiency. This concerns various mathematical objects, from integers and polynomials to ideals in number fields, lattices, or linear algebra.

Since the early 1990's, no new algorithm improved on the complexity of NFS. As it is used in practice, the algorithm has complexity $L_N(1/3, (64/9)^{1/3})$ for factoring general integers or for computing discrete logarithms in prime fields of similar size (the so-called "multiple polynomial" variants have better complexity by a very thin margin, but this has not yet yielded to a practical improvement). Given the wide use of the underlying hard problems, progress in this area is of utmost importance. In 2013, several new algorithms have modified the complexity of the discrete logarithm problem in small characteristic fields, which is a closely related problem, reaching a heuristic quasi-polynomial time algorithm [BGJT14, 3, 4, 5]. A stream of computational records have been obtained since 2013 using these algorithms, using in particular techniques from polynomial system solving, or from Galois theory. These new algorithms, together with these practical realizations, have had a very strong impact of course on the use of small-characteristic fields for cryptography (now clearly unsuitable), as well as on pairings on elliptic curves over small-characteristic finite fields (which are also no longer considered safe to use).

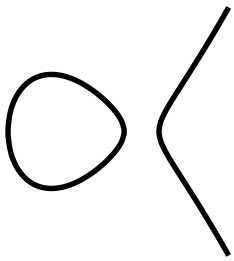
[1] National Institute of Standards and Technology. *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, `http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf`, 2011.

[2] Agence nationale de la sécurité des systèmes d'information. *Référentiel général de sécurité, annexe B1*, `http://www.ssi.gouv.fr/fr/reglementation-ssi/referentiel-general-de-securite/`. Version 2.03, 2014.

[3] A. Joux, *A New Index Calculus Algorithm with Complexity $L(1/4+o(1))$ in Small Characteristic*. In T. Lange, K. Lauter, and P. Lisoněk (eds.), Selected Areas in Cryptography – SAC 2013, vol. 8282 of *Lecture Notes in Comput. Sci.*, 355–379. Springer–Verlag, 2014.

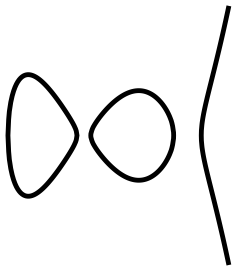
[4] F. Göloglu, R. Granger, G. McGuire, and J. Zumbrägel, *On the Function Field Sieve and the Impact of Higher Splitting Probabilities*. In R. Canetti and J. A. Garay (eds.), CRYPTO 2013, vol. 8043 of *Lecture Notes in Comput. Sci.*, 109–128. Springer–Verlag, 2013. , Part II.

[5] R. Granger, T. Kleinjung, and J. Zumbrägel, *On the Powers of 2*, 2014. Available at `http://eprint.iacr.org/2014/300`. Cryptology ePrint Archive report.



A genus-1 curve

$y^2 = x^3 + ax + b.$



A genus-2 curve

$y^2 = x^5 + a_4x^4 + \cdots + a_0.$

While it is relatively easy to set public key sizes for RSA or Diffie–Hellman that are "just above" the reach of academic computing power with NFS, the sensible cryptographic choice is to aim at security parameters that are of course well above this feasibility limit, in particular because assessing this limit precisely is in fact a very difficult problem. In line with the security levels offered by symmetric primitives such as AES-128, public key sizes should be chosen so that with current algorithmic knowledge, an attacker would need at least 2^{128} elementary operations to solve the underlying hard problem. Such security parameters would call for RSA key sizes above 3,000 bits, which is seldom seen, except in contexts where computing power is plentiful anyway.

Since the mid-1980's, elliptic curves, and more generally Jacobians of algebraic curves, have been proposed as alternative mathematical settings for building cryptographic primitives. The discrete logarithm problem in these groups is formidably hard, and in comparison to the situation with the traditional primitives mentioned above, the cryptanalysis algorithms are such that the appropriate public-key size grows only linearly with the desired security level: a 256-bit public key, using algebraic curves, is well suited to match the hardness of AES-128. This asset makes algebraic curves more attractive for the future of public-key cryptography.

Challenges related to algebraic curves in cryptology are rather various, and call for expertise in several areas. Suggesting curves to be used in the cryptographic context requires to solve the point counting problem. This may be done by variants of the Schoof–Elkies–Atkin algorithm and its generalizations (which, in genus 2, require arithmetic modulo multivariate systems of equations), or alternatively the use of the complex multiplication method, a rich theory that opens the way to several problems in computational number theory.

The long-awaited transition from the legacy primitives to primitives based on curves is ready to happen, only circumstantially slowed down presently by the need to agree on a new set of elliptic curves (not because of any attack, but because of skepticism over how the currently widespread ones have been generated). The Internet Research Task Force has completed in 2015 a standardization proposal [6]. In this context, the recommended curves are not of the complex multiplication family, and enjoy instead properties that allow fast implementation, and avoid a few implementation difficulties. Those are also naturally chosen to be immune to the few known attacks on the discrete logarithm problem for curves. No curve of genus 2 has made its way to the standardization process so far, however one candidate exists for the 128-bit security level [GS11].

The discrete logarithm problem on curves is very hard. Some results were obtained however for curves over extension fields, using techniques such as the Weil descent, or the point decomposition problem. In this context, the algorithmic setup connects to polynomial system solving, fast arithmetic, and linear algebra.

Another possible route for transitioning away from RSA and finite field-based cryptography is suggested, namely the switch to the "post-quantum" cryptographic primitives. Public-key cryptographic primitives that rely on mathematical problems related to Euclidean lattices or

[6] A. Langley, M. Hamburg, and S. Turner. *Elliptic Curves for Security*. IRTF draft, available at `http://tools.ietf.org/html/draft-irtf-cfrg-curves`, version 11, 2015.

coding theory have an advantage: they would resist the potential advent of a quantum computer. Research on these topics is quite active, and there is no doubt that when the efficiency challenges that are currently impeding their deployment are overcome, the standardization of some post-quantum cryptographic primitives will be a worthwhile addition to the general cryptographic portfolio. The NSA has recently devoted an intriguing position text to this topic [7] (for a glimpse of some of the reactions within the academic community, the reference [8] is useful). Post-quantum cryptography, as a research topic, is complementary to the topics we address most, which are NFS and algebraic curves. We are absolutely confident that, at the very least for the next decade, primitives based on integer factoring, finite fields, and algebraic curves will continue to hold the lion's share in the cryptographic landscape. We also expect that before the advent of standardized and widely developed post-quantum cryptographic primitives, the primitives based on algebraic curves will become dominant (despite the apparent restraint from the NSA on this move).

We acknowledge that the focus on cryptographic primitives is part of a larger picture. Cryptographic primitives are part of cryptographic protocols, which eventually become part of cryptographic software. All these steps constitute research topics in their own right, and need to be scrutinized (as part of independent research efforts) in order to be considered as dependable building blocks. This being said, the interplay of the different aspects, from primitives to protocols, sometimes spawns very interesting and fruitful collaborations. A very good example of this is the LogJam attack [ABD+15].

4 The extended family of the Number Field Sieve

4.1 State of the art

The Number Field Sieve (NFS) has been the leading algorithm for factoring integers for more than 20 years, and its variants have been used to set records for discrete logarithms in finite fields. It is reasonable to understand NFS as a framework that can be used to solve various sorts of problems. Factoring integers and computing discrete logarithms are the most prominent for the cryptographic observer, but the same framework can also be applied to the computation of class groups. The state of the art with the NFS is built from numerous improvements of its inner steps. For further reference, we mention the most important of these steps.

- The polynomial selection step chooses two polynomials from which the algebraic structures used in the NFS are defined. There is an immense degree of freedom in the choice of the polynomial pairs, and some are better than others, sometimes much better. The goal of the polynomial selection step is to find an exceptionally good pair, which will minimize the time required by the subsequent steps of the NFS.

- The relation collection step depends on the algebraic structures defined by the previous step. Among a gigantic search space, values for which some crucial conditions are met lead to relations. The relation collection step gathers a very large number of these. It is composed of inner steps, commonly referred to as sieving and cofactorization. For factoring integers, the relation collection step is the most computationally expensive step.

- The linear algebra step solves a large and sparse linear system. Depending on which problem is to be solved exactly (integer factoring or discrete logarithm), the system may be homogeneous or inhomogeneous, and the definition field may either be GF(2) or GF(p) for a large prime p. The first part of the linear algebra step is a preprocessing step commonly referred to as filtering, whose aim is to reduce the linear system size. The second step uses black box linear algebra algorithms such as the block Wiedemann algorithm.

[7] National Security Agency, *Cryptography Today*, 2015/08/19. `https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml`.

[8] N. Koblitz and A. J. Menezes, *A Riddle Wrapped in an Enigma*, 2015. Available at `http://eprint.iacr.org/2015/1018`. Cryptology ePrint Archive report.

- The last step depends on the context. In the factoring context, it is called the square root step, and in the discrete logarithm context it is replaced by the individual logarithm computation. In both cases, this step represents a negligible amount of time with respect to the overall computation.

In terms of algorithmic improvements, the recent research activity on the NFS family has been rather intense. In particular, for the discrete logarithm problem over finite fields of small characteristic (defined as $\mathrm{GF}(p^n)$ with $p < L_{p^n}(1/3, c)$), several new algorithms have been discovered [BGJT14, 3, 4, 5], and their practical reach has been demonstrated by actual experiments [9, 10]. In particular, the Function Field Sieve variant of NFS, for which the latest record set was [BBD+14], has been superseded by these new algorithms. One striking novelty of the algorithms that have been proposed in 2013 is the introduction of polynomial systems as a crucial means to accelerate the computation. Those improvements have not had, however, any impact so far on the integer factorization problem, or on the discrete logarithm problem in large characteristic (for $p > L_{p^n}(2/3, c)$). As for the so-called medium characteristic case which is crucial for pairing-based cryptography, some recent progress has also been obtained [BP14, BGGM15, 11].

Improvements on the sub-steps of NFS have also been obtained recently, the most important being the new polynomial selection algorithm from [BBKZ15]. On the practical side, existing algorithms are making their way to practical use, such as batch cofactorization [12, 13, 14], or multiple matrix products [15]. Both ideas have been mentioned in [16].

The algorithmic contributions of the CARAMBA members (including, but not limited to the ones mentioned above [BGJT14, BBD+14, BP14, BGGM15, BBKZ15]) to NFS would hardly have been possible without access to a dependable software implementation. To this end, members of the CARAMBA team have been developing the CADO-NFS software suite since 2007. CADO-NFS is now the most widely visible open source implementation of NFS, and is a crucial platform for developing prototype implementations for new ideas for the many sub-algorithms of NFS. CADO-NFS is free software (LGPL) and follows an open development model, with publicly accessible development repository and regular software releases. Competing free software implementations exist, such as `msieve`, developed by J. Papadopoulos. In Lausanne, T. Kleinjung develops his own code base, which is unfortunately not public.

Software such as CADO-NFS is meant to be used, in particular to demonstrate potential weaknesses of cryptographic implementations, or to set records. These two goals are not exactly identical: sadly, deployed cryptography is too often using security levels that are severely outdated, and attacks could require much less than record-level computations. For example, CADO-NFS

[9] R. Granger, T. Kleinjung, and J. Zumbrägel, *Breaking '128-bit Secure' Supersingular Binary Curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$)*. In J. A. Garay and R. Gennaro (eds.), CRYPTO 2014, vol. 8617 of *Lecture Notes in Comput. Sci.*, 126–145. Springer–Verlag, 2014. , Part II.

[10] R. Granger, T. Kleinjung, and J. Zumbrägel, *Discrete Logarithms in GF(2^9234)*, Jan. 2014. Email to the `NMBRTHRY` mailing-list.

[11] A. Guillevic, *Computing Individual Discrete Logarithms Faster in GF(p^n) with the NFS-DL Algorithm*. In T. Iwata and J. H. Cheon (eds.), ASIACRYPT 2015. *Lecture Notes in Comput. Sci.* Springer–Verlag, 2015. To appear.

[12] D. J. Bernstein, *How to find small factors of integers*, 2002. Available at `http://cr.yp.to/papers.html#sf`. Preprint.

[13] D. J. Bernstein, *How to find smooth parts of integers*, 2004. Available at `http://cr.yp.to/papers.html#smoothparts`. Preprint.

[14] J. Franke, T. Kleinjung, F. Morain, and T. Wirth, *Proving the Primality of Very Large Numbers with fastECPP*. In D. Buell (ed.), ANTS-VI, vol. 3076 of *Lecture Notes in Comput. Sci.*, 194–207. Springer–Verlag, 2004.

[15] T. Kleinjung, *Filtering and the matrix step for NFS*, Workshop on Computational Number Theory, 2008. Available at `http://event.cwi.nl/wcnt2011/slides/kleinjung.pdf`.

[16] T. Kleinjung, J. W. Bos, and A. K. Lenstra, *Mersenne Factorization Factory*. In P. Sarkar and P. Iwata (eds.), ASIACRYPT 2014 (1), vol. 8873 of *Lecture Notes in Comput. Sci.*, 358–377. Springer–Verlag, 2014.

has been used recently in [17, ABD+15] to show how such misconfigurations could reveal true practical weaknesses.

As to assessing the feasibility limit of integer factoring, the RSA-768 factoring record that has been set in 2010 still holds. Recent evolution of the polynomial selection step has been a rightful reason to defer the launch of the factorization of the next natural target, RSA-896, for which the latest runtime projections are in the whereabouts of 200,000,000 CPU hours.

4.2 Detailed research objectives

Participating permanent members: All.
Ongoing or recently completed projects of non-permanent staff: Grémy (higher dimensional sieving, →2017); Bouvier (NFS filtering, →2015); Jeljeli (NFS linear algebra, →2015); Coxon (NFS nonlinear polynomial selection, post-doc →2015).

4.2.1 Short-term objectives

The CADO-NFS software is a very valuable base for research on improving NFS. The merits of algorithmic improvements, or of improvements directly related to the implementation itself, are best understood when experimented in the context of a well-established and optimised software implementation. In CARAMBA, we intend to pursue the work on NFS, which entails in particular making it ready to tackle larger challenges. Several of the important computational steps of NFS that are currently identified as stumbling blocks will be worked on: polynomial selection, strategies for cofactorization (separation into large primes), use of auxiliary hardware for cofactorization (GPUs or various kinds of coprocessors) and also the improvement of linear algebra efficiency for DLP computation. This entails both algorithmic and software development.

Since about a decade, factoring records have been a clear demonstration of the importance of the so-called cofactorization sub-step, which is part of the relation collection step. The efficiency of implementations of this step is conditioned by very different hardware characteristics than the rest of the sieving step. Some preparation work will be necessary before we can try alternative approaches to the cofactorization sub-step. Preliminary work has been already reported in this direction, using GPUs, FPGAs, or with software-based solutions using asymptotically fast techniques based on remainder trees [12, 13, 14]. Using such techniques, it is possible to save on the cofactorization step by handling a large number of cofactorization attempts simultaneously. In order to increase the benefit of this approach, we will make cofactorization a collective operation involving several computers.

Following the CATREL ANR project (see §13), as well as [ABD+15], we will also work on the computation of discrete logarithms. A short-term goal is to solve a kilobit-size discrete logarithm challenge for what is called an "SNFS prime". This is a much easier computation than what a general prime of the same size would entail. The currently limiting factor is the performance of the linear algebra step.

In the context of a collaboration with University of Kaiserslautern, we will work on the connection of the NFS framework with problems that are more firmly in the number theoretic realm: computation of class groups of number fields, for example. One motivation for going in that direction is the connection of the principal ideal problem with the cryptanalysis of schemes using so-called ideal lattices.

4.2.2 Medium-term and long-term objectives

We will work on providing better runtime predictions for NFS. Existing work on this topic has unfortunately not been able to meet the challenge with the desired accuracy. It is remarkable that the state of the art for runtime predictions for NFS is made of mere extrapolations based

[17] B. Beurdouche, K. Bhargavan, A. Delignat-Lavaud, C. Fournet, M. Kohlweiss, A. Pironti, P.-Y. Strub, and J. K. Zinzindohoue, *A Messy State of the Union: taming the Composite State Machines of TLS*, 2015 IEEE Symposium on Security and Privacy, 535–552. IEEE, 2015.

on the asymptotic formula $L_N(1/3, (64/9)^{1/3})$, which embeds a $(1 + o(1))$ term in the exponent, thus making these extrapolations rather artificial. Instead, we need to properly model the input, the output, and the running time of all the NFS steps in a satisfactory way, and versatile enough to account for various parameter changes. The results obtained will have to be backed with convincing experiments. This work will be of great value to assess the hardness of factoring or computing discrete logarithms with the Number Field Sieve, and will also help choose optimal parameters for NFS.

Linear algebra for NFS is not the dominant step of integer factorization (but for discrete logarithm problems, it is). This step has however to run on more expensive hardware, featuring high-performance interconnects for example. For this reason, the acknowledged room for improvement in this step will be worked on. The core operations call for the design of adapted sparse BLAS-like subroutines, adapted to CPU and cache hierarchies, and arranged in an optimal way to obtain best performance. While some design ideas can be shared with similar projects from the numerical world, the different nature of the data handled (real numbers versus finite-field elements) clearly hinders direct reuse.

The ultimate success of a cryptanalytic enterprise is the dismissal of its target from the cryptographic landscape. In this vein, we aim at a situation where RSA is only a relic from early-days cryptography, and is completely replaced by better cryptosystems, in particular those based on algebraic curves. A particular landmark in this direction would be the factorization of RSA-1024.

It is important to stress that in a post-RSA cryptographic world, the knowledge acquired on NFS and its variants would still be very valuable. NFS expertise would remain essential for assessing the security of cryptographic pairings, for instance. Also, the NFS approach is sufficiently generic, and reaches towards related problems such as the principal ideal problem.

4.2.3 Milestones

- Factorization of RSA-896; RSA-1024 is a longer-term goal in sight.
- Discrete logarithm record modulo a 1024-bit SNFS prime.
- Updated knowledge on the NFS algorithm will be the topic of a reference book to be written.

5 Algebraic Curves in Cryptology

5.1 State of the art

The challenges associated to algebraic curves in cryptology are diverse, because of the variety of mathematical objects to be considered. These challenges are also connected to each other. On the cryptographic side, efficiency matters. Much effort has been put towards providing secure instances of curves that enjoy fast arithmetic, especially so for curves of genus 1 (elliptic curves) or Jacobians of curves of genus 2. Beyond the pure implementation speed challenge, the curve choice in itself leads to interesting research topics.

Choosing a curve for the appealing characteristics of its defining equation regarding implementation speed is one strategy. This requires to address the challenging problem of point counting. While this challenge is efficiently overcome with so-called p-adic algorithms when the characteristic of the base field is tiny, the case of large characteristic, which is preferred for cryptographic uses, is harder. Polynomial-time algorithms exist, but they are by no means trivial in genus 2 and have ramifications into several areas of computer algebra.

Alternatively, the construction of curves with Jacobian cardinality known in advance, through the "CM" (complex multiplication) construction, avoids this issue but cannot compete in terms of implementation performance. Furthermore, CM curves tend to raise skepticism because of hypothetical cryptographic weaknesses. Other strategies exist, such as [18]; again, the design

[18] S. D. Galbraith, X. Lin, and M. Scott, *Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves*, J. Cryptology **24**(3) (2011), 446–469.

choices that allow for fast arithmetic in this case are not desirable for a global standard.

As of 2015, the most widely used set of elliptic curves, the so-called NIST curves, are in the process of being replaced by a new set of candidate elliptic curves for future standardization. This standardization effort is coordinated by the IRTF (Internet Research Task Force).

On the cryptanalytic side, the discrete logarithm problem on (Jacobians of) curves has resisted all attempts for many years. Among the currently active topics, the decomposition algorithms raise interesting problems related to polynomial system solving, as do attempts to solve the discrete logarithm problem on curves defined over binary fields. In particular, while it is generally accepted that the so-called Koblitz curves (base field extensions of curves defined over GF(2)) are likely to be a weak class among the various curve choices, no concrete attack supports this claim fully.

The mathematical theory around curves and abelian varieties in general is quite rich. Important mathematical objects in relationship to abelian varieties are their endomorphism rings and the isogenies between abelian varieties. While the concrete cryptographic application is often interested in the situation over finite fields, the understanding of the broader picture includes the situation over number fields or the complex numbers. Several research directions arise from this study. On the constructive side, the complex multiplication theory is a way to generate curves with prescribed properties over finite fields. Those are not ideal from an implementation speed perspective, though. A second aspect is the explicit computation of isogenies, and the art of walking the so-called isogeny graph of a given abelian variety. This graph is important because the discrete logarithm problem on connected abelian varieties is of roughly equal difficulty, and being able to efficiently transport it leads to a potential attack if a weak instance exists in the graph.

Software dealing with many of the algebraic and analytic structures connected to elliptic curves and abelian varieties includes first and foremost the Magma computer algebra software, seconded by Sage. The Pari/GP software and library also provide some functionality in this area. Advanced algorithms are also found in external packages, such as the Echidna and AVIsogenies packages.

Work by the CARAMBA members covers many aspects of the realm of algebraic curves in cryptology: fast arithmetic [Gau07], algorithms for the discrete logarithm problem [DT08, Gau09, EGT11], algorithms for constructing curves [GS11, ET14], as well as software.

5.2 Detailed research objectives

Participating permanent members: Gaudry, Spaenlehauer, Thomé.
Ongoing or recently completed projects of non-permanent staff: Abelard (Higher genus point counting, →2017); Labrande (Theta functions,→ 2016); Massierer (Quasi-polynomial DLP for curves, post-doc → 2015).

5.2.1 Short-term objectives

Jacobians of algebraic curves, and more generally abelian varieties, come with a rich structure. Important mathematical operations such as the Abel–Jacobi map and its converse (which connect to the analytic representation), or the computation of theta functions, are unfortunately not done in quasi-linear time with current algorithms, while we expect that this should be the case. Likewise, acceptably fast computation of endomorphism rings or explicit isogenies, which are the subject of current research, deserve improvements both in theoretical and practical terms. Being experts both on algebraic curves and also on arithmetic topics, we plan to improve this situation. Together with work already done by team members, this is intended to form a coherent set of advanced tools for curves of genus 2, to be merged with existing software, e.g., in Magma or Sage.

The security of pairing-based cryptography in general has recently faced serious attacks that are the consequence of our work. While the small characteristic case has now clearly lost most of its cryptographic relevance since [BGJT14], developments around NFS might in fact endanger some other well-known classes of so-called "pairing-friendly" curves [BGGM15]. We are in position to investigate further in this direction.

5.2.2 Long-term objectives

The point counting problem for genus 2 curves, although solvable in polynomial time, is hard. For potential cryptographic use, the cardinality of the Jacobian as well as its quadratic twist must both have a large prime factor. The current point counting situation with generalizations of the Schoof–Elkies–Atkin algorithm is that it is feasible to try sufficiently many trial curves to find one that meets these criteria, for the 128-bit security level [GS11]. We will work on the polynomial system arithmetic that occupies a large part of these computations, and see how we can improve on point counting for the 192-bit security level.

Although less directly interesting for cryptography, we plan to investigate point-counting for genus 3 curves, again with generalizations of the Schoof–Elkies–Atkin algorithm. As far as we know, the theoretical polynomial-time algorithm has yet to be turned into something practical. We expect, again, that a careful study of the polynomial systems involved in this setting is key to a fast implementation.

The discrete logarithm problem on abelian varieties is an extremely hard problem. We have contributed to identifying several weak instances, while more cases are the subject of very active research to which we take part. Those are for example the so-called decomposition approach for the discrete logarithm problem over extension fields, where polynomial systems undoubtedly play a central role through the point decomposition problem.

Last, an open problem as of now is the potential extension of the recent quasi-polynomial finite field discrete logarithm algorithm to favorable instances of algebraic curves.

5.2.3 Milestones

- Compute the cardinality of a general genus 2 Jacobian over a 256-bit prime field, and find a general cryptographically secure genus 2 Jacobian over a 192-bit prime field.
- Set a first point counting record for genus 3 Jacobian over a prime field, using a Schoof-like algorithm.
- Provide reference software implementation for quasi-linear computation of theta functions and the Abel–Jacobi map.

6 Computer arithmetic

6.1 State of the art

Computer arithmetic is part of the common background of all team members, and is naturally ubiquitous in the two previous application domains mentioned, which are the Number Field Sieve family and the cryptographic applications of abelian varieties. However involved the mathematical objects considered may be, dealing with them first requires to master more basic objects: integers, finite fields, polynomials, and real and complex floating-point numbers. Libraries such as GNU MP, GNU MPFR, GNU MPC do an excellent job for these, both for small and large sizes (we rarely, if ever, focus on small-precision floating-point data, which explains our lack of mention of libraries relevant to it).

Over the past 15 to 20 years, asymptotically fast algorithms have taken over the realm of arbitrary precision computer arithmetic. Software libraries such as GNU MP have matured to include asymptotically fast algorithms for multiplication as well as many related operations. Furthermore, competing or complementary projects such as MPIR, Flint or Arb (developed by F. Johansson from the LFANT project-team in Bordeaux) provide alternative implementations. More specialized algorithms and implementations are also found in GMP-ECM, or in several projects focused towards computing digits of π. These various implementations often include hardware-specific improvements and/or deal with out-of-core computations.

Some operations however are less covered by mainstream computer arithmetic libraries. For example, our library GF2X has little to no competitors for arithmetic with binary polynomials.

Likewise, algorithms performing operations like Lagrange reconstruction, or more generally various kinds of operations related to product and remainder trees are generally programmed *ad hoc*, with no leading implementation available, and in particular nothing addressing the concern of parallel or out-of-core computations.

The latest striking algorithmic improvement on the complexity of integer multiplication by Fürer in 2007 [19] has been followed by more recent work [20]. It is generally understood that the range where this algorithm improves over the existing ones is truly remote. While recent research articles bring a somewhat more optimistic view, with practicality being perhaps nearer than once thought, it is still the case that no practical implementation of Fürer's algorithm or of its variants currently exists.

The term "arithmetic" may also be understood in a broader sense. All the mathematical structures that appear in the context of the study of number fields or algebraic curves are extremely important for our research perspectives. For dealing with these mathematical objects, higher level software packages such as Magma, Sage, or Pari/GP are the places where implementations of the most advanced algorithms are found.

6.2 Detailed research objectives

Participating permanent members: Detrey, Gaudry, Thomé, Zimmermann.
Ongoing or recently completed projects of non-permanent staff: Covanov (Fast asymptotic arithmetic, →2017).

Most of our involvement in subjects related to computer arithmetic is to be understood in connection to our applications to the Number Field Sieve and to abelian varieties.

We will work on the topic of optimal bilinear complexity, following the line of work we started in [BDEZ12]. The goal here will be to improve on existing methods for exhaustively searching for formulae computing a given bilinear application, such as polynomial or matrix multiplication, by exploiting inherent symmetries of the problem at hand. Beyond the interest in the fundamental problems attached to these questions, improved algorithms in this area can lead to efficiency gains in arithmetic on Jacobians of curves, for example.

Motivated both by an application for the computation of class polynomials in genus 2 [ET14], as well as by the extension of the cofactorization sub-step of NFS to a collective operation involving several nodes (mentioned in §4.2.1), we will provide a parallel application handling the arithmetic of variants of remainder trees on a computer cluster, when memory footprint is a stumbling block.

We will continue to work on the arithmetic libraries in which we have crucial involvement, such as GNU MPFR, GNU MPC, GF2X, MPFQ, and also GMP-ECM.

In connection with the computation of the Abel–Jacobi map and its converse (mentioned in §5.2.1), we will work on accurate rounding guarantees for these operations.

The absence of a reference implementation for the latest variations of Fürer's multiplication algorithm is a gap that ought to be filled. We are likely to work on this.

7 Polynomial systems

7.1 State of the art

Systems of polynomial equations have been part of the cryptographic landscape for quite some time, with applications to the cryptanalysis of block and stream ciphers, as well as multivariate cryptographic primitives. Concerning the cryptographic primitives based on number theory, relying on hard problems such as the discrete logarithm problem on curves or finite fields, the three most important connections to polynomial systems are the Weil descent, the point decomposition

[19] M. Fürer, *Faster integer multiplication.* In U. Feige (ed.), STOC '07, 57–66. ACM, 2007.

[20] D. Harvey, J. van der Hoeven, and G. Lecerf, *Even faster integer multiplication*, 2014. Available at `http://arxiv.org/abs/1407.3360`. Preprint.

problem [21, Gau09], and the intermediate $L(1/4)$ algorithm for discrete logarithms over fields of small characteristic, which was discovered shortly prior to the quasi-polynomial algorithm for this problem.

Polynomial systems arising from the situations above are usually not generic, in the sense that they have some distinct structural properties, such as symmetries, or bilinearity for example. Such properties lead to faster solving by the F_4 and F_5 algorithms of Faugère, even if implementations do not knowingly exploit them. However, during the last decades, several results have shown that identifying and exploiting these structures can lead to dedicated Gröbner bases algorithms that can achieve large speedups compared to generic implementations [FSS11, FSS14].

Polynomial systems are also at the heart of a series of papers [22, 23] claiming an improvement for the elliptic curve discrete logarithm problem over binary fields. These works are unfortunately relying on the so-called "first-fall-degree assumption", which is currently being disputed.

The best known Gröbner basis implementations are Faugère's FGb [24] and Magma [25]. Both are closed source, but the open-source specialized linear algebra software GBLA[1] for Gröbner bases and the open-source implementation of F_4 written by Coladon, Joux, and Vitse[2] have been recently released and provide efficient routines for critical parts of the F_4 and F_5 algorithms.

7.2 Detailed research objectives

Participating permanent members: Gaudry, Spaenlehauer, Thomé.
Ongoing or recently completed projects of non-permanent staff: Abelard (Higher genus point counting, →2017); Massierer (Quasi-polynomial DLP for curves, post-doc →2015).

Solving polynomial systems is well done by existing software, and duplicating this effort is not relevant. However we will develop testbed open-source software for ideas relevant to the specific polynomial systems that arise in the context of our applications. The TinyGB software, that we describe further in Section 9.3.2, is our platform to test new ideas.

In this context, we will pursue the study of the specificities of the polynomial systems that are strongly linked to our targeted applications, and for which we have significant expertise [FSS11, FSS14]. We also want to see these recent results provide practical benefits compared to existing software, in particular for systems relevant for cryptanalysis, such as in the context of small characteristic finite field DLP, or for systems arising in constructive aspects of cryptology such as point counting on low genus curves.

The current situation with respect to the ECDLP in small characteristic is unclear. Subexponential, or even polynomial-time algorithms have been claimed, but under assumptions that are not really convincing. Practical experiments become practically unfeasible well before it is possible to observe a complexity behaviour; and a more theoretical approach is also very hard. For the moment, our position on this particularly hot topic is to follow carefully what is done, but we did not yet contribute to it. This could of course change in the future, either by proposing new potential attacks (hopefully under more reliable assumptions than the present ones), or in the direction of analyzing the ones that have been already proposed.

[21] I. A. Semaev, *Summation polynomials and the discrete logarithm problem on elliptic curves*, 2004. Available at `http://eprint.iacr.org/2004/031`. Cryptology ePrint Archive report.

[22] C. Petit and J.-J. Quisquater, *On Polynomial Systems Arising from a Weil Descent*. In X. Wang and K. Sako (eds.), ASIACRYPT 2012, vol. 7658 of *Lecture Notes in Comput. Sci.*, 451–466. Springer–Verlag, 2012.

[23] I. A. Semaev, *New algorithm for the discrete logarithm problem on elliptic curves*, 2015. Available at `http://eprint.iacr.org/2015/310`. Cryptology ePrint Archive report.

[24] J.-C. Faugère, *FGb: A Library for Computing Gröbner Bases*. In K. Fukuda, J. van der Hoeven, M. Joswig, and N. Takayama (eds.), Mathematical Software - ICMS 2010, vol. 6327 of *Lecture Notes in Comput. Sci.*, 84–87. Springer–Verlag, 2010.

[25] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24**(3-4) (1997), 235–265. Computational algebra and number theory (London, 1993).

[1] `http://hpac.imag.fr/gbla/`
[2] `https://www-fourier.ujf-grenoble.fr/~viva/f4/html/index.html`

8 Technological and societal impact

8.1 Better awareness and avoidance of cryptanalytic threats

Our study of the Number Field Sieve family of algorithms aims at showing how the threats underlying various supposedly hard problems are real. Our record computations, as well as new algorithms, contribute to having a scientifically accurate assessment of the feasibility limit for these problems, given academic computing resources. The data we provide in this way is a primary ingredient for government agencies whose purpose includes guidance for the choice of appropriate cryptographic primitives. For example the French ANSSI[3], German BSI, or the NIST[4] in the United States base their recommendations on such computational achievements.

The software we make available to achieve these cryptanalytic computations also allows us to give cost estimates for potential attacks to cryptographic systems that are taking the security/efficiency/legacy compatibility trade-offs too lightly. Attacks such as LogJam [ABD+15] are understood as being serious concerns thanks to our convincing proof-of-concepts. In the LogJam context, this impact has led to rapid worldwide security advisories and software updates that eventually defeat some potential intelligence threats and improve confidentiality of communications.

8.2 Promotion of better cryptography

We also promote the switch to algebraic curves as cryptographic primitives. Those offer nice speed and excellent security, while primitives based on elementary number theory (integer factorization, discrete logarithm in finite fields), which underpin e.g., RSA, are gradually forced to adopt unwieldy key sizes so as to comply with the desired security guarantees of modern cryptography. Our contributions to the ultimate goal of having algebraic curves eventually take over the cryptographic landscape lie in our fast arithmetic contributions, our contributions to the point counting problem, and more generally our expertise on the diverse surrounding mathematical objects, or on the special cases where the discrete logarithm problem is not hard enough and should be avoided.

We also promote cryptographically sound electronic voting, for which we develop the Belenios prototype software, (licensed under the AGPL). It depends on research made in collaboration with the PESTO (ex-CASSIS) team, and provides stronger guarantees than current state of the art.

8.3 Key software tools

The vast majority of our work is eventually realized as software, listed in detail in §9. We can roughly categorize it in two groups. Some of our software covers truly fundamental objects, such as the GNU MPFR, GNU MPC, GF2X, or MPFQ packages. To their respective extent, these software packages are meant to be included or used in broader projects. For this reason, it is important that the license chosen for this software allows proper reuse, and we favor licenses such as the LGPL, which is not restrictive. We can measure the impact of this software by the way it is used in e.g., the GNU Compiler Collection (GCC), in Victor Shoup's Number Theory Library (NTL), or in the Sage computer algebra system. The availability of these software packages in most Linux distributions is also a good measure for the impact of our work.

We also develop more specialized software. Our flagship software package is CADO-NFS, and we also develop some others with various levels of maturity, such as GMP-ECM, CMH, or Belenios, aiming at quite diverse targets. Within the lifespan of the CARAMBA project, we expect more software packages of this kind to be developed, specialized towards tasks relevant to our research targets: important mathematical structures attached to genus 2 curves, generation of cryptographically secure curves, or tools for attacking cryptographically hard problems. Such software both illustrates our algorithms, and provides a base on which further research work can be established. Because of the very nature of these specialized software packages as research

[3] In [2], the minimal recommended RSA key size is 2048 bits for an usage up to 2030. See also Annex B, in particular Section B.1 "Records de calculs cryptographiques".

[4] The work [KAF+10] is one of the only two academic works cited by NIST in the report [1].

topics in their own right, needing both to borrow material from other projects, and being possible source of inspiring material for others, it is again important that these be developed in a free and open-source development model.

9 Software

Software is truly central to our research. We develop several software packages. Some can serve as companions to our publications, so as to provide reproducible research results, and some are research objects in their own right. We are committed to providing software that goes beyond the simple proof of concept, especially so when we believe it can be useful to some users (either within or outside the academic world). Our flagship software aims therefore at being well-developed and reasonably portable. Software development is clearly an important part of our activity.

9.1 Flagship software

This section lists our most visible software packages. Among these packages, the research activity at the moment is intense concerning CADO-NFS (§9.1.1). In contrast, GNU MPFR, GNU MPC, and GMP-ECM (§9.1.2 and §9.1.3) are less active research topics than they once were.

9.1.1 CADO-NFS

CADO-NFS is a complete and original implementation of the Number Field Sieve, licensed under the LGPL. Work on CADO-NFS started in 2007, and the cumulative development effort since then, involving permanent researchers, PhD students, and engineers, has been quite substantial. CADO-NFS represents today more than 200,000 lines of (mainly) C code, and a history of more than 10,000 modifications. CADO-NFS targets the factorization of medium-size numbers, as well as larger targets, nearing records. CADO-NFS can also be used to compute discrete logarithms in finite fields. CADO-NFS is commonly used by scholars worldwide as a way to provide reference timings for the Number Field Sieve, or just to factor numbers arising in their research.

Continued development on CADO-NFS is an inherent companion to the research work we plan to achieve on the extended family of NFS, detailed in §4. Short-term goals include in particular more integration work related to the finite field discrete logarithm problem. This includes linear algebra on GPUs, following the PhD thesis of H. Jeljeli in 2015. In the longer term, we ambition to extend the reach of CADO-NFS towards further factoring and discrete logarithm records, and provide improved assessments for the hardness of factoring 1024-bit numbers, or computing discrete logarithms modulo prime numbers of 768 to 1024 bits, the relevance of which has been exemplified by [ABD+15].

9.1.2 GNU MPFR and GNU MPC

Starting in Nancy in 1998, a research project on implementing arithmetic for arbitrary precision floating-point numbers with correct rounding has led to the development of what has become the MPFR software library. MPFR extends the IEEE-754 standard in two ways: beyond double precision and beyond basic arithmetic operations. MPFR has now evolved into a mature project that implements all arithmetic operations of Annex F of the ISO C99 standard, as well as some special functions. An extensive test suite ensures code quality. GNU MPFR is currently developed jointly between the ARIC and CARAMBA Inria teams (mostly V. Lefèvre and P. Zimmermann). Since 2002, the MPC library, dealing with arbitrary precision floating-point complex numbers, was developed jointly with the Inria LFANT project in Bordeaux. Now named GNU MPFR and GNU MPC and licensed under the GNU LGPL, both libraries are pervasive in many software projects, as being for example dependencies of the GNU Compiler Collection. Only a little more than basic maintenance work is planned with GNU MPFR and GNU MPC over the next years.

9.1.3 GMP-ECM

GMP-ECM is an integer factoring tool developed by P. Zimmermann and others since 1998. Because it implements the Elliptic Curve Method for factoring integers, GMP-ECM is good at finding moderate-size factors. Many records of this kind have been obtained with GMP-ECM over the years. GMP-ECM includes several original algorithms (using decomposable hyperelliptic curve Jacobians for example, or original stage 2 algorithms), as well as optimizations both for computation on small data (assembly optimizations) and large data (out-of-core computation of large Fourier transforms). GMP-ECM is distributed under the GNU GPL for the executable and under the LGPL for the library. It is a package in many Linux distributions, and is furthermore used by Magma and Sage. Like MPFR and MPC, GMP-ECM is now a mature tool, no special investment is planned in the future, except of course basic support for users.

9.2 Utility software

9.2.1 MPFQ

MPFQ is a code generation library for finite fields. It produces optimized code for computations that occur in cryptography, namely with moderate-size finite fields. The most important benefit is drawn at compile-time, exploiting the idea that in cryptography or cryptanalysis, the very same finite fields are used over and over again. The benefit of the approach was demonstrated in 2007 by providing record-speed implementations of cryptosystems using curves of genus 1 and 2. Those records have long been superseded, but the approach has remained the same. While MPFQ has been developed mostly for internal usage (it is used to generate some code within CADO-NFS, GF2X and TinyGB), it is freely distributed and will continue to be developed and maintained.

9.2.2 GF2X

GF2X is a small software library focusing on the implementation of one single operation, namely the multiplication of polynomials with binary coefficients. Originally developed in 2007 with Richard Brent (Univ. of Canberra, Australia), GF2X has since evolved to cover several operating systems and hardware architectures, taking advantage of dedicated instruction sets when available. GF2X includes several algorithms that have no other known public implementation for the multiplication of large binary polynomials, such as Schönhage's ternary FFT, Cantor's additive FFT, or the 2009 variant due to Gao and Mateer. GF2X is GPL-licensed and used by NTL and Sage. An LGPL-licensed subset is used by CADO-NFS. The future development plans for GF2X include a 2.0 release, with better run-time level choice of the best assembly routines, inspired by the mechanism optionally used in GMP, and proper support of the ARM architecture.

9.3 Software presently under development

9.3.1 CountG2

CountG2 contains various software pieces for efficient point counting of genus 2 hyperelliptic curves over prime fields. It is a C++ implementation based on the NTL library that was used half a decade ago to set record computations [GS11], but has not been maintained since then. The code needs to be revamped to accomodate the research plan of Section 5, and to make it easier to maintain. New algorithm developments related to genus 3 point counting and interactions with explicit isogenies will be implemented. As a consequence, the name of the library that currently refers explicitly to genus 2 will probably change.

9.3.2 TinyGB

TinyGB is a small software library written in C++ whose main objective is to serve as a hub between existing libraries (linear algebra, finite field arithmetic) in order to compute Gröbner bases with the Faugère F_4 algorithm. It is currently under development. Its goal is not to

increase the efficiency of the building blocks of Gröbner basis computations, but rather to be flexible and modular, in order to provide a framework in which new ideas can be conveniently tested. One critical routine in Gröbner basis algorithms is linear algebra on structured matrices. At the moment, this is achieved in TinyGB by using the FFLAS library[5], and we also plan to provide an interface to the GBLA library[6], which is specialized for matrices arising in Gröbner bases computations. TinyGB relies on MPFQ (see §9.2.1) for the arithmetic over finite fields. The objective is that TinyGB shall provide a convenient experimentation tool to adapt classical Gröbner basis algorithms to the polynomial systems that are encountered in the research themes of the CARAMBA project.

9.3.3 Belenios

Belenios is an electronic voting protocol, together with a free software implementation. Belenios guarantees privacy and verifiability. It is an evolution of the Helios voting system[26, CGGI14], but has been re-implemented independently. This is part of a collaboration with the PESTO (ex-CASSIS) Inria project-team. With Belenios, some security issues of Helios have been fixed (the possibility for the server hosting the ballot box to commit ballot stuffing, in particular), and specificities of the French rules have been taken into account. Belenios is a platform that allows us to advertise the new research ideas (mostly done in PESTO), and to facilitate our interaction with the industry on the e-voting topic, demonstrating what we believe to be the current academic standard in terms of security.

Although still in development, Belenios is mature enough to be used for real-life elections.

10 Positioning in the research community

10.1 Positioning with respect to topics

We are visible members of the international research communities around algorithmic number theory and elliptic and hyperelliptic curve cryptography, and we also have connections to the topics of computer arithmetic and symbolic computation. Beyond this visibility in our fields of expertise, one of the traits which characterize well our scientific methodology is our activity on multiple aspects, from the abstract mathematical levels, down to the low-level implementation.

Major achievements of the team members (records [KAF+10], new algorithms [BGJT14], highly visible cryptanalytic attacks [ABD+15]) have contributed to giving the group a worldwide expertise recognition on the topic of integer factorization and discrete logarithm. We share the international leadership position on this topic with the LACAL group at EPFL, which has been a close partner for several past projects, and with which future collaborations are planned. In the context of the CATREL grant, we also collaborate with the Inria GRACE project in Saclay on algorithmic improvements of NFS, and with the ECO team in LIRMM, Montpellier. We have some common work with University of Paris 6 (Institut de Mathématiques de Jussieu) on this topic, too.

The Number Field Sieve has more number theoretical uses such as class group computations. Those connect us to research groups such as the mathematics department at University of Kaiserslautern, Germany, with which a grant proposal has been submitted. This topic is also relevant for other groups with which we have existing collaborations (see §11): the Inria LFANT project (notably in the Pari/GP context) and the University of Calgary.

We have a significant expertise on algebraic curves in cryptography and worked on all important aspects of the related problems in recent years, both for curves of genus 1 (elliptic curves) and 2, for example [Gau07, DT08, GS11, EGT11, ET14]. We have been involved in the Elliptic Curve Cryptography conference series for many years, either as organizers, invited speakers, program committee or steering committee members. Various aspects of our work on this topic are of

[5] `http://www-ljk.imag.fr/membres/Jean-Guillaume.Dumas/FFLAS/`

[6] `http://www-polsys.lip6.fr/~jcf/Software/GBLA/index.html`

[26] B. Adida, O. de Marneffe, and O. Pereira, *Helios voting system*. `https://vote.heliosvoting.org/`.

particular interest to the groups at Microsoft Research, Redmond (fast arithmetic, endomorphism ring computations, complex multiplication), as well as TU Eindhoven to a lesser extent.

The computer arithmetic topic binds us most to the Inria ARIC and LFANT projects through GNU MPFR and GNU MPC, as well as the user and developer communities around these software packages. Furthermore, since the GF2X package is (optionally) used by NTL, we are in frequent contact with V. Shoup regarding the evolution of the interfacing of the two libraries.

On the topic of symbolic computation in general, we hired P.-J. Spaenlehauer in 2013, coming from the Inria POLSYS project. Collaborations with POLSYS are continuing, on the topic of structured polynomial systems. Our expertise in the domain also covers the polynomial systems arising from the point counting computations in genus 2, which is the topic of a long existing collaboration with the University of London, Ontario.

Still in the symbolic computation area, we work also on high-performance sparse linear algebra algorithms over finite fields, in the context of CADO-NFS, that is, integer factoring and discrete logarithms. We collaborate on this topic with LIRMM in Montpellier, as well as LJK in Grenoble.

Outside our research scope, we briefly mention the topic of post-quantum cryptography. We do not contribute to this research direction, with the possible exception that we might become interested in some particular kinds of Euclidean lattices (ideal lattices, namely) due to the connection with algebraic number theory topics.

10.2 Positioning with respect to other INRIA project-teams

The dynamics of hiring processes have led to a situation where the members of the Inria projects CARAMBA, GRACE, ARIC, LFANT, and to a lesser extent POLSYS have often been colleagues working on similar topic in previous times. We share a common cultural background with these teams, and we also globally agree on the benefit of clearly identified specific research objectives for each team (which does in no way preclude collaboration). We therefore try to give an explanation of the exact positioning of our work with respect to them.

With the GRACE group (2 members of CARAMBA did their PhD there), we collaborate on the development of improved algorithms for discrete logarithms in finite fields. This topic is not pushed forward as being one of the main topics for GRACE, so CARAMBA naturally has the lead on this topic. Beyond that, the arithmetic geometry aspects worked on in GRACE are not studied a lot in CARAMBA, and the coding theory part is not studied at all in CARAMBA.

We have many connections with LFANT: team leaders have known each other for long, one LFANT member is a former PhD student of our group, and post-docs traveled between the groups. Our joint work on genus 2 complex multiplication is a slightly more important objective for LFANT than it is for CARAMBA, and the study of alternative modular functions aiming at optimizing the computation time, which could be an extension, is quite distant from the CARAMBA priorities. On the computations related to endomorphism rings of elliptic curves, the algebraic theta function part is rather done by LFANT, while the analytic aspects are studied more in CARAMBA. On the GNU MPC subject, development is organized and rationalized between the two teams. On the NFS topic, while potential intersections could exist, no duplicate work exists in LFANT.

Three former members of our group (V. Lefèvre, G. Hanrot, D. Stehlé) have joined the ARIC team in Lyon. Euclidean lattices, which are an important topic for ARIC, are mostly outside of the scope of CARAMBA. On the arithmetic side, the development on GNU MPFR is split between ARIC and our group.

Our connections with POLSYS are more recent. P.-J. Spaenlehauer continues to work with POLSYS on the topic of structured polynomial systems, and polynomial systems for the problems arising in the cryptographic and cryptanalytic perspectives are an important part of the CARAMBA proposal. In this context, the TinyGB software which is developed in CARAMBA is to be understood as an experimentation tool to be used for testing algorithmic refinements that are specific to the polynomial systems considered. This is not a piece of software which is meant to compete with the FGb software, especially for polynomial systems that are not related a priori to cryptographic contexts.

10.3 Relationship with LORIA laboratory

Within LORIA, CARAMBA will be part of Department 1: "Algorithms, Computation, Image & Geometry" (teams ABC, ADAGIO, ALICE, CARAMBA, MAGRIT, VEGAS). PhD students from the whole department present their work in a yearly seminar. Depending on topics, CARAMBA seminars occasionally attract attendees from other teams of the department (e.g. some aspects of polynomial systems are related to geometry as studied in VEGAS), and conversely. Fruitful interactions also exist with the PESTO (ex-CASSIS) group, which studies cryptographic protocols from a formal point of view: CARAMBA and PESTO collaborate on the electronic voting topic (cf §9.3.3). Last, the regular seminar "SSL" (*Séminaire de Sécurité du LORIA*) is co-organized by teams CARAMBA, PESTO, as well as MADYNES and CARTE, thereby spanning four teams and three departments within LORIA.

11 National and International Collaborations

11.1 Nationally

In the national landscape, we have strong and fruitful connections with Inria teams such as LFANT, GRACE, POLSYS, and ARIC, in terms which have been detailed in the paragraph above. We also have some potential connections to SECRET, notably on binary fields arithmetic.

Despite not being versed into the formal methods aspect of cryptography, we have fruitful connections to the Inria PESTO (ex-CASSIS) and PROSECCO teams, on the topic of electronic voting on the one hand, and on real-life attacks to cryptographic protocols on the other hand.

We collaborate with other French research groups: with IRMAR (Rennes) on algebraic curves in cryptography, with LIRMM (Montpellier) on the Number Field Sieve, with IMJ (Institut de Mathématiques de Jussieu, University Paris 6, Paris) on NFS again but more on the purely algorithmic side.

11.2 Internationally

Internationally, our colleagues are found both within computer science and mathematics departments. We have existing international collaborations with groups at EPFL, the university of Kaiserslautern, and the university of Calgary, on topics already mentioned above. We also have good connections with Microsoft Research, Redmond, with TU Eindhoven. The regular and very fruitful collaboration with the team of R. Brent (Univ. of Canberra, Australia) has diminished now that R. Brent is officially retired.

12 Team composition and organization

Section 15 provides a short CV for the permanent team members, and lists their current PhD students, often co-supervised by two permanent members. Involvement of permanent and non-permanent staff in the research axes of CARAMBA is detailed in the chapters above. The group occupies five offices of 2-3 persons, making interaction easy, lively, and very frequent. The team runs a seminar (5 to 10 talks a year). The team also has two associate members: M. Videau, who was an assistant professor at Université de Lorraine until December 2014, and now works with the Quarkslab company. She co-supervises Grémy's PhD thesis. Also, L. Sanselme is a teacher in "classes préparatoires", and is interested in the arithmetic-related topics studied by the group, as well as the MPFQ software.

13 Funding

The team members have had, over the previous years, a satisfactory success rate with ANR (Agence Nationale de la Recherche) non-thematic research programs. Those have contributed

significantly to the funding of the group. A current grant (CATREL) extends until end of 2015, and two new grant applications (including an international project) are currently being reviewed. Another significant part of the team resources is covered by the contract with the HTCS company (renewed annually since 2012), which consists in consulting and training work on the topic of the Number Field Sieve.

As a rough measure of the team's budget, the funding for the non-permanent staff working within the group goes as follows. One post-doc is funded by an ANR grant. One post-doc is visiting on a 2-year grant from the Swiss National Science Foundation. Three PhD students are funded by ministry grants, and one is funded on the team's own resources. No temporary engineer staff works in the group as of nov. 2015, although we do obtain funding for temporary engineer contracts every once in a while, roughly every other year or so (the limiting factor for temporary engineer contracts is the difficulty in hiring people, more than budget constraints).

In 2014, P. Zimmermann has applied for an ERC advanced grant, and in 2015 E. Thomé has applied for an ERC consolidator grant. Both applications were unsuccessful.

We intend to proceed with the existing strategy, where the appropriate timing for grant applications is decided exclusively on the basis of scientific opportunity relative to our objectives on the one hand, and the timeline of existing grants on the other hand.

14 Selected publications from team members

References [1] to [26] refer to publications not from team members, cited as footnotes in the text.

Journal articles

[BBKZ15] S. Bai, C. Bouvier, A. Kruppa, and P. Zimmermann, *Better polynomials for GNFS*, Math. Comp. (2015). Available at `http://hal.inria.fr/hal-01089507`. Accepted for publication.

[ET14] A. Enge and E. Thomé, *Computing class polynomials for abelian surfaces*, Experiment. Math. **23** (2014), 129–145.

[BP14] R. Bărbulescu and C. Pierrot, *The Multiple Number Field Sieve for Medium and High Characteristic Finite Fields*, LMS Journal of Computation and Mathematics **17**(special issue A) (2014), 230–246.

[EGT11] A. Enge, P. Gaudry, and E. Thomé, *An $L(1/3)$ discrete logarithm algorithm for low degree curves*, J. Cryptology **24**(1) (2011), 24–41.

[FSS11] J.-C. Faugère, M. Safey El Din, and P.-J. Spaenlehauer, *Gröbner Bases of Bihomogeneous Ideals generated by Polynomials of Bidegree (1,1): Algorithms and Complexity*, J. Symbolic Comput. **46**(4) (2011), 406–437.

[GS11] P. Gaudry and É. Schost, *Genus 2 point counting over prime fields*, J. Symbolic Comput. **47**(4) (2011), 368–400.

[Gau09] P. Gaudry, *Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem*, J. Symbolic Comput. **44**(12) (2009), 1690–1702.

[DT08] C. Diem and E. Thomé, *Index calculus in class groups of non-hyperelliptic curves of genus three*, J. Cryptology **21**(4) (2008), 593–611.

[Gau07] P. Gaudry, *Fast genus 2 arithmetic based on Theta functions*, J. Math. Cryptol. **1**(3) (2007), 243–265.

Peer-reviewed international conferences

[ABD+15] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. Alex Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Béguelin, and P. Zimmermann, *Imperfect Forward Secrecy: How Diffie-Hellman fails in practice*, CCS'15. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 5–17. ACM, 2015.

[BGGM15] R. Bărbulescu, P. Gaudry, A. Guillevic, and F. Morain, *Improving NFS for the discrete logarithm problem in non-prime finite fields*. In E. Oswald and M. Fischlin (eds.), EUROCRYPT 2015 (1), vol. 9056 of *Lecture Notes in Comput. Sci.*, 129–155. Springer–Verlag, 2015.

[CGGI14] V. Cortier, D. Galindo, S. Glondu, and M. Izabachène, *Election Verifiability for Helios under Weaker Trust Assumptions*, Proceedings of the 19th European Symposium on Research in Computer Security (ESORICS'14), 2014.

[FSS14] J.-C. Faugère, P.-J. Spaenlehauer, and J. Svartz, *Sparse Gröbner bases: the unmixed case*. In K. Nabeshima (ed.), ISSAC 2014, 178–185. ACM, 2014.

[BGJT14] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé, *A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic*. In P. Q. Nguyen and E. Oswald (eds.), EUROCRYPT 2014, vol. 8441 of *Lecture Notes in Comput. Sci.*, 1–16. Springer–Verlag, 2014.

[BBD+14] R. Barbulescu, C. Bouvier, J. Detrey, P. Gaudry, H. Jeljeli, E. Thomé, M. Videau, and P. Zimmermann, *Discrete logarithms in* $\mathrm{GF}(2^{809})$ *with FFS*. In H. Krawczyk (ed.), Public Key Cryptography - PKC 2014, vol. 8383 of *Lecture Notes in Comput. Sci.*, 221–238. Springer–Verlag, 2014.

[BDEZ12] R. Barbulescu, J. Detrey, N. Estibals, and P. Zimmermann, *Finding optimal formulae for bilinear maps*. In F. Özbudak and F. Rodríguez-Henríquez (eds.), WAIFI 2012, vol. 7369 of *Lecture Notes in Comput. Sci.*, 168–186. Springer–Verlag, 2012.

[KAF+10] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev, and P. Zimmermann, *Factorization of a 768-bit RSA modulus*. In T. Rabin (ed.), CRYPTO 2010, vol. 6223 of *Lecture Notes in Comput. Sci.*, 333–350. Springer–Verlag, 2010.

[BGTZ08] R. Brent, P. Gaudry, E. Thomé, and P. Zimmermann, *Faster Multiplication in GF(2)[x]*. In A. van der Poorten and A. Stein (eds.), ANTS-VIII, vol. 5011 of *Lecture Notes in Comput. Sci.*, 153–166. Springer–Verlag, 2008.
(extraite automatiquement de Hal)

15 Short Vitae from Permanent Team Members

Jérémie Detrey (35, Inria research scientist)

2010–	Inria research scientist (CR1) at Inria Nancy–Grand Est.
2008–2010	Inria junior research scientist (CR2) at Inria Nancy–Grand Est.
2007–2008	Teaching assistant (postdoc) at B-IT (Bonn, Germany).
2003–2007	Ph.D. thesis at LIP (ÉNS Lyon), under the supervision of F. de Dinechin and J.-M. Muller.
2003	M.Sc. in computer science at ÉNS Lyon.
2000	Admission at École normale supérieure de Lyon.

Research interests: Computer arithmetic, cryptology, hardware implementation.

Ph.D. students:

2014– Co-advisor (with E. Thomé) of S. Covanov's Ph.D. thesis.

2011–2015 Co-advisor (with E. Thomé) of H. Jeljeli's Ph.D. thesis.

2009–2013 Co-advisor (with P. Gaudry) of N. Estibals's Ph.D. thesis

Pierrick Gaudry (42, CNRS senior research scientist)

2010– CNRS senior research scientist (DR2) at LORIA.

2008 Habilitation à diriger les recherches, University of Nancy 1 (UHP).

2005–2010 CNRS research scientist (CR1) at LORIA.

2001–2005 CNRS junior research scientist (CR2) at LIX, École polytechnique.

1998–2001 Ph.D. thesis at LIX (École polytechnique), under the supervision of F. Morain.

1995 M.Sc. in computer science at École polytechnique.

1993 Admission at École normale supérieure de Cachan.

Research interests: Computational number theory, cryptology.

Responsibilities within the scientific community:

2011–2012 Deputy director of the LORIA laboratory

2010–2015 Scientific leader of the Caramel project-team

2006–2009 Coordinator of the CADO project of the ANR Blanc

2003–2005 Vice-head of the TANC project-team at Inria Futurs

Ph.D. students:

2015– Co-advisor (with P.-J. Spaenlehauer) of S. Abelard's Ph.D. thesis.

2013– Co-advisor (with M. Videau) of L. Grémy's Ph.D. thesis

2009–2013 Co-advisor (with J. Detrey) of N. Estibals's Ph.D. thesis

2008–2011 Co-advisor (with T. Lange, TU Eindhoven) of G. Bisson's Ph.D. thesis

Pierre-Jean Spaenlehauer (31, Inria research scientist)

2016 – Inria research scientist (CR1) at Inria Nancy–Grand Est.

2014 – 2015 Inria junior research scientist (CR2) at Inria Nancy–Grand Est.

2013 postdoc at Max Planck Institute for Mathematics (Bonn, Germany).

2012 – 2013 postdoc at Western University (London, Ontario, Canada).

2009 – 2012 Ph.D. thesis in POLSYS project-team (Inria Paris-Rocquencourt/Univ. Paris 6), under the supervision of J.-C. Faugère and M. Safey El Din.

2008 – 2009 M.Sc. Master Parisien de Recherche en Informatique.

2005 – 2008 Engineer student, École polytechnique.

2002 – 2005 Bachelor's degree in Mathematics, Université Louis Pasteur, Strasbourg.

Research interests: Computer algebra, polynomial systems.

Ph.D. student:

2015– Co-advisor (with P. Gaudry) of S. Abelard's Ph.D. thesis.

Emmanuel Thomé (39, Inria senior research scientist)

2015– Inria senior research scientist (DR2) at Inria Nancy–Grand Est.

2012 Habilitation à diriger les recherches, Université de Lorraine.

2006–2015 Inria research scientist (CR1) at Inria Nancy–Grand Est.

2003–2006 Inria junior research scientist (CR2) at Inria Lorraine.

1999–2003 Ph.D. thesis at LIX (École polytechnique), under the supervision of F. Morain.

1997 M.Sc. in computer science at École polytechnique.

1995 Admission at École normale supérieure.

Research interests: Computational number theory, linear algebra.

Responsibilities within the scientific community:

2012–2015 Coordinator of the ANR project CATREL.

2011–2014 Elected member of Inria Evaluation Board.

2009–2012 Partner contact for the CHIC project of the ANR Blanc (project CHIC coordinated in Rennes).

Ph.D. students:

2014– Co-advisor (with J. Detrey) of S. Covanov's Ph.D. thesis.

2013– Co-advisor (with M. J. Jacobson, Jr) of H. Labrande's Ph.D. thesis.

2011–2015 Co-advisor (with J. Detrey) of H. Jeljeli's Ph.D. thesis.

2008–2011 Co-advisor (with G. Hanrot) of R. Cosset's Ph.D. thesis.

Paul Zimmermann (51, Inria senior research scientist)

2008– Inria senior research scientist (DR1) at Inria Nancy–Grand Est.

2001 Habilitation à diriger les recherches, University Nancy 1.

1998-2007 Inria senior research scientist (DR2) at Inria Nancy–Grand Est.

1994-1998 Inria research scientist (CR1) at Inria Nancy–Grand Est.

1991-1994 Inria junior research scientist (CR2) at Inria Rocquencourt then Nancy–Grand Est.

1987–1991 Ph.D. thesis at Inria Rocquencourt, under the supervision of Ph. Flajolet.

Research interests: Computational number theory, arbitrary precision arithmetic.

Responsibilities within the scientific community:

2013– Scientific Director and Chair of the Projects Committee at Inria Nancy–Grand Est.

2011-2014 Elected member of Inria Scientific Board.

2011-2012 Head of the SED engineer team of Inria Nancy–Grand Est.

2010– Vice-head of the Caramel project-team at Inria Nancy–Grand Est.

2006–2009 Vice-head of the Cacao project-team at Inria Nancy–Grand Est.

2000–2006 Head of the Nancy part of the Spaces project-team.

1998–2000 Head of the PolKA project-team at Inria Lorraine.

2005–2007 Elected member of Inria Evaluation Board.

1999–2001 Elected member of Inria Evaluation Board.

2001–2009 Member of the Program Committee of the Arith conference.

Ph.D. students:

2011–2015 Advisor of Cyril Bouvier's Ph.D. thesis.

2007–2010 Advisor of Alexander Kruppa's Ph.D. thesis.

2003–2006 Advisor of Laurent Fousse's Ph.D. thesis.

2003–2005 Advisor of Damien Stehlé's Ph.D. thesis.

1994–1997 Co-advisor (with Pierre Lescanne) of François Bertault's Ph.D. thesis.