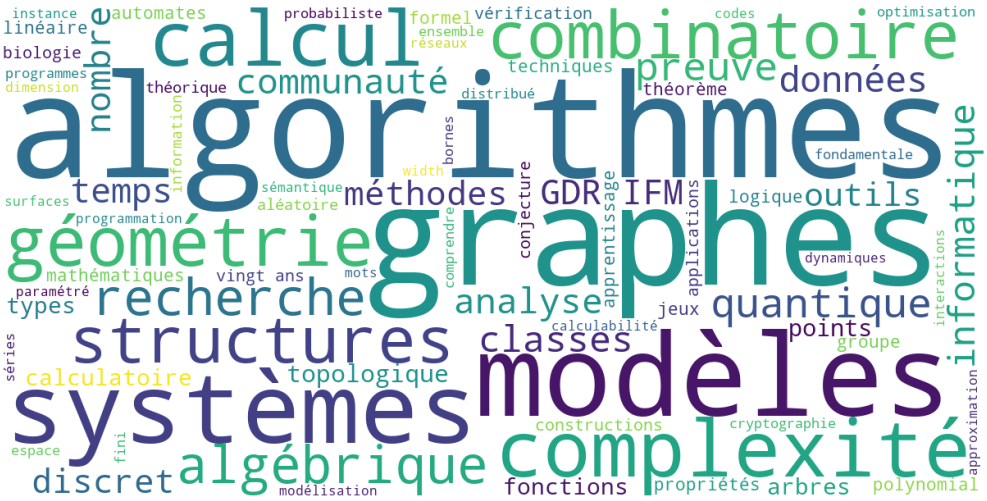
GDR Groupement de recherche
IFM Informatique Fondamentale et ses Mathématiques

Les vingt ans du GdR IFM



Calcul formel, arithmétique et cryptographie

Les vingt ans du GdR IFM, vus du GT Calcul Formel

L'objet premier de ce corpus synthétique est d'opérer un bilan rétrospectif sur l'évolution du calcul formel sur approximativement les vingt dernières années, c'est-à-dire du début des années 2000 et jusqu'à fin 2025, en essayant de distiller ses caractéristiques ainsi que l'évolution scientifique de ses thématiques de recherche et en indiquant quelques éléments saillants et faits marquants qui nous semblent pertinents pour cette période. Pour ce faire, nous avons opté pour recueillir le témoignage des responsables d'équipes, et plus largement les chercheuses et chercheurs, qui composent notre communauté scientifique. Ce qui suit est une brève synthèse structurée de ces retours.

1 Introduction

Le **calcul formel** est une discipline à l'intersection et à l'interface de l'informatique et des mathématiques, plus spécifiquement des mathématiques constructives.

L'aspect informatique consiste en la **conception d'algorithmes manipulant de façon exacte des objets et expressions mathématiques formels comme des polynômes ou des séries** ainsi qu'en l'étude et l'analyse de leur complexité. Le calcul se veut soit exact, soit précis et certifié. Parmi ces algorithmes, on peut citer ceux dédiés aux *briques fondamentales* comme la multiplication d'entiers, de polynômes ou de matrices, ceux réduisant des problèmes classiques à ces briques, comme la division ou l'inversion, ou encore ceux s'appuyant sur des structures plus générales comme les polynômes à plusieurs variables ou les matrices à coefficients polynomiaux.

L'aspect mathématique consiste à **construire, ou à expliciter formellement un objet, par exemple pour prouver son existence** (par opposition à l'énoncé de son existence en exploitant le principe du tiers exclu). On peut notamment citer ici le théorème fondamental de l'algèbre qui a désormais une preuve élémentaire purement constructive [7]. De telles preuves sont aussi en particulier des algorithmes et se prêtent donc aux mêmes traitements : implémentations, analyses de complexité, etc. Cette concordance entre preuves et programmes a notamment apporté un éclairage fort utile aux preuves (factorisation, simplification, etc.). La richesse et l'utilité de ce mariage sont apparues de manière de plus en plus évidente au cours des dernières décennies, dans un cadre bien plus large que le calcul formel, comme en témoigne par exemple l'engouement actuel de la communauté mathématique pour les assistants de preuves tels que Rocq ou Lean. Ainsi, des preuves mathématiques très longues, comme la preuve du théorème de l'ordre impair établissant que tout groupe fini d'ordre impair est résoluble, ont pu être ainsi entièrement vérifiées [9] [1].

À l'intersection de la preuve et de l'algorithmique, le développement des *mathématiques expérimentales* s'est accéléré au cours de la dernière décennie. Le calcul concret, possible grâce aux implémentations disponibles et de plus en plus performantes et précises, permet d'**intuiter des nouvelles propriétés ou des nouvelles structures mathématiques** qui seront éventuellement prouvées ultérieurement.

1. La preuve de Feit et Thompson publiée en 1963 remplit un numéro entier de 255 pages du Pacific Journal of Mathematics.

2 Spécificités de la recherche en calcul formel

Il est très important de mentionner que les avancées considérées aujourd'hui majeures en calcul formel (que ce soit en mathématiques constructives ou en complexité algorithmique) se sont opérées sur le **temps long**. Les résultats marquants qui sont reportés aujourd'hui (fin 2025) sur ce document sont le fruit d'un **effort continu, persistant et de longue haleine** qui a commencé 20 (voire 30) années plus tôt et qui est difficilement perceptible pour les non-spécialistes durant toute la période de ses développements.

On notera par ailleurs une tendance relativement marquante des deux dernières décennies de la recherche en calcul formel : on cherche moins à concevoir et à implémenter des algorithmes résolvant les problèmes dans leur généralité, et on s'intéresse davantage à des sous-classes structurées. Toujours dans un souci de gain de performance, des algorithmes probabilistes ont été par ailleurs conçus et implémentés. Un exemple frappant est la résolution de systèmes polynomiaux par calcul accéléré de bases de Gröbner en combinant l'arithmétique multi-modulaire avec des méthodes probabilistes (certifiables sous conditions de généricité souvent vérifiées en pratique) [8].

3 Algorithmes et complexité

Le calcul exact mène naturellement à des études de complexité en pire cas qui ne sont pas toujours en accord avec ce que l'utilisateur observe expérimentalement. Ces écarts n'ont cessé de motiver des **améliorations notables des bornes de complexités théoriques** soit en exploitant des structures particulières (comme les symétries ou le caractère creux pour ne citer que les plus communes) avant de s'attaquer éventuellement au cas le plus général, soit en améliorant la complexité de certaines briques de base comme la multiplication. Dans ce dernier cas, améliorer ne serait-ce qu'un peu la complexité d'une de ces briques a bien souvent un impact majeur et transversal sur plusieurs bornes de complexité théorique. Nous mentionnerons deux exemples saillants.

1. **Complexité de la multiplication**. Dans le cas des entiers (resp. des polynômes univariés), nous connaissons déjà une réponse partielle : elle est, au pire, quasi-linéaire en la taille (resp. le degré) n. Plus précisément, il existe $a > 0$ tel que le produit de deux entiers de taille n (resp. polynômes de degré n) se calcule en $O(n(\log n)^a)$ opérations binaires (resp. arithmétiques). En 1971, Schönhage et Strassen ont montré que, dans le cas des entiers, a est au plus $1+o(1)$ en conjecturant qu'il serait possible de se débarrasser du $o(1)$. Ces vingt dernières années, cette borne pire-cas théorique n'a cessé d'être améliorée par paliers successifs jusqu'à atteindre la borne présumée $O(n \log n)$ en 2021, **prouvant ainsi une conjecture ouverte depuis un demi-siècle** [10]. Notons également que la complexité théorique du produit de matrices a été améliorée à plusieurs reprises, bien que la quasi-optimalité $O(n^2(\log n)^a)$ reste encore incertaine et en tout cas hors d'atteinte pour le moment.

2. **Réduction de calculs en algèbre linéaire**. L'inversion de matrice, la décomposition LU, le calcul du noyau ou encore le calcul du déterminant ne sont pas plus difficiles que le calcul du produit de matrices carrées. Cependant, le calcul du polynôme caractéristique échappait jusqu'ici à cette règle hormis sous des hypothèses de *généricité* ou via des méthodes probabilistes. En 2021, le premier algorithme pour le **calcul du polynôme caractéristique** dont la complexité en temps déterministe est la même que celle du produit a été proposé [14].

Une des thématiques historiques du calcul formel, la résolution exacte des systèmes polynomiaux d'équations et/ou d'inéquations et inégalités dans le cas réel, a aussi connu des avancées majeures. Alors que les situations pire cas peuvent avoir des sorties de taille gigantesque, les situations génériques et/ou structurées ont permis d'exploiter des leviers menant à des algorithmes dont la complexité est polynomiale voire quasi-optimale en la taille de l'entrée et de la sortie [15].

Le calcul formel ne se cantonne pas seulement au calcul exact. La communauté s'est fortement mobilisée pour développer des **algorithmes symboliques-numériques** afin de tirer partie de la puissance du calcul numérique. À cet égard, l'usage des librairies multi-précisions comme MPFR a été instrumental [2]. On peut citer en particulier les nombreux algorithmes à base de méthode d'homotopie et/ou de déformation, qui sont très efficaces en grande dimension. Mentionnons également la **résolution du 17e problème de Smale**, posant la question du calcul d'une solution approchée d'un système polynomial en temps moyen polynomial, tout d'abord via un algorithme probabiliste puis via un algorithme déterministe [3, 11]. Notons par ailleurs la résolution récente d'un problème majeur et fondamental qui a résisté pendant un demi-siècle : l'**évaluation numérique multi-points d'un polynôme univarié** en un nombre d'opérations binaires quasi-linéaire en son degré [13].

4 Calcul formel et mathématiques

Dans certaines disciplines mathématiques, comme la combinatoire ou la géométrie algébrique réelle, **le traitement algorithmique**, ou constructif, des objets manipulés a grandement contribué à des **avancées novatrices**. Des améliorations quantitatives notables ont été obtenues par ce biais sur les 20 dernières années [2]. En guise d'illustration, citons deux exemples frappants qui montrent la puissance et l'intérêt de l'approche constructive pour **enrichir le corpus de résultats mathématiques**.

1. Estimations précises des **constantes impliquées dans l'inégalité de Łojasiewicz dans le cadre semi-algébrique**. Dans sa version analytique originale, cette inégalité borne la distance d'un point quelconque au zéro le plus proche d'une fonction analytique. Les estimations récentes obtenues dans le cas semi-algébrique par des méthodes algorithmiques sont **plus précises que celles obtenues jusque ici par les géomètres** [1].
2. Le **17e problème de Hilbert** qui demande si tout polynôme multivarié qui ne prend aucune valeur strictement négative est une somme de carrés de fonctions rationnelles. Ce résultat a été démontré par Emil Artin au début du 20e siècle mais sa preuve était non constructive. En particulier aucune borne sur le degré des numérateurs et dénominateurs n'était connue. Des preuves constructives avec des bornes primitives récursives (tours d'exponentielles dont les hauteurs dépendaient du nombre de variables) ont été données plus d'un demi-siècle plus tard. Seulement très récemment, des **bornes élémentaires ont été établies (tours d'exponentielles à hauteur bornée), résolvant ainsi un problème ouvert de longue date**. La preuve utilise notamment les *sous-résultants* qui font partie des objets de base étudiés en calcul formel [12].

Pour souligner l'utilité des **mathématiques expérimentales**, à l'intersection du calcul formel et des mathématiques, on citera les marches de Gessel en combinatoire qui constituent un modèle classique de chemins dans le quart de plan, où un marcheur part de l'origine et effectue des pas parmi l'ensemble $\{\rightarrow, \nearrow, \leftarrow, \swarrow\}$. Pendant longtemps, la nature exacte de la

2. `https://www.mpfr.org`, prix du logiciel libre de recherche 2025, catégorie scientifique et technique.

série génératrice comptant ces chemins est restée ouverte, en particulier sa D-finitude. Le problème a été résolu en 2010 : la **série génératrice associée aux marches de Gessel est algébrique** et donc D-finie, c'est-à-dire solution d'une équation différentielle à coefficients polynomiaux [5]. La preuve repose sur une combinaison d'outils de calcul formel, dont la capacité à effectuer des calculs concrets et corrects. Il s'est agit de **calculer des équations fonctionnelles** satisfaites par les premiers termes de la série, puis de **prouver leur validité** pour la série complète.

Nous finissons cette section avec une mention spéciale pour l'usage des méthodes constructives et du calcul formel dans le cadre spécifique de l'algèbre différentielle avec des implémentations matures et raisonnablement efficaces qui ont naturellement trouvé des applications en biologie et en physique. On citera **les améliorations notables qui ont été apportées au calcul de plusieurs types d'intégrales premières** (algébriques, élémentaires, rationnelles etc.) ainsi qu'à la théorie de **l'élimination différentielle** qui demande une mise en forme particulière du système d'équations fort semblable à la forme triangulaire dans le cadre purement algébrique. L'usage des *chaînes régulières* par exemple a permis un **calcul effectif des bases de Ritt-Raudenbush** (analogue au théorème de base de Hilbert) [16].

5 Logiciel et reproductibilité des résultats

La communauté de calcul formel s'est toujours montrée volontaire pour implémenter, que ce soit à bas niveau comme en `C` ou à plus haut niveau dans un logiciel de calcul formel, les algorithmes qu'elle développe. Les fins sont multiples : pouvoir reproduire des résultats, résoudre des problèmes venant d'applications, déterminer les limites des méthodes et des algorithmes mais aussi développer de nouvelles approches en s'aidant de l'ordinateur. Ainsi, une **évolution logicielle notable s'est produite en calcul formel sur les 20 dernières années**. Elle s'est opérée sur deux niveaux.

1. D'abord, l'apparition de librairies spécialisées de plus en plus performantes qui permettent un calcul exact ou certifié. Citons par exemple FLINT (théorie des nombres, calcul modulaire) et LinBox (algèbre linéaire) [3]. Ce gain en performance n'est pas uniquement dû à de bonnes implémentations mais aussi à des avancées notables et décisives en algorithmique, par exemple [6].

   S'ajoute à cela une **hybridation numérique/symbolique** qui a été extrêmement profitable pour **gagner en performance sans sacrifier l'exactitude des calculs**. On mentionnera la librairie PariGP qui se spécialise entre autres en calcul efficace en théorie des nombres et est lauréate de deux prix prestigieux en 2021 et 2024 [4], ou encore msolve (qui implémente notamment l'algorithme F4 de Faugère) bibliothèque devenue incontournable pour la résolution des systèmes polynomiaux multivariés [5], sans oublier le projet Differential Algebra implémentant l'état de l'art des méthodes pour la résolution de l'équivalent différentiel des systèmes polynomiaux [6].

2. Ensuite, et jusqu'au début des années 2000, les logiciels ou systèmes de calcul formel (*computer algebra systems*) les plus utilisés, bien qu'issus des laboratoires de calcul scientifique, évoluaient pour devenir propriétaires tout en gardant une étroite collaboration

3. `https://flintlib.org`, `https://linalg.org`.
4. `https://pari.math.u-bordeaux.fr`, ACM SIGSAM Richard Dimick Jenks Memorial Prize 2021 et prix science ouverte du logiciel libre de la recherche 2024, catégorie communauté.
5. `https://msolve.proj.lip6.fr`
6. `https://codeberg.org/francois.boulier/DifferentialAlgebra/`

avec la communauté, citons par exemple Maple ou Magma [7]. L'avènement de **plateformes libres** (open source), à l'instar de SageMath ou Macaulay2 [8], qui agrègent de nombreux composants autour du calcul numérique et symbolique, a non seulement permis une **diffusion plus large**, mais aussi et surtout de faciliter et d'**uniformiser l'accès aux librairies spécialisées**.

6 Dissémination du calcul formel

Aujourd'hui SageMath est typiquement utilisé dans l'enseignement de l'option calcul formel en agrégation de mathématiques. Par ailleurs l'outillage logiciel développé au sein de la communauté calcul formel a eu un impact majeur sur des communautés adjacentes. Le champ d'application de ces outils n'a cessé de s'étendre, modifiant ainsi les pratiques de certaines communautés. Par exemple, les **calculs de bases de Gröbner** sont devenus presque routiniers en **cryptographie** pour la cryptanalyse de **crypto-systèmes post-quantiques**, de primitives cryptographiques symétriques.

Le **télescopage créatif**, et ses différentes évolutions [4], pour le calcul d'équations différentielles linéaires (et parfois de formes closes) pour des séries formelles issues de calculs d'intégration ou de sommation multiples à paramètres sert désormais d'outil incontournable pour montrer des équivalences ou des égalités connues, voire pour calculer de nouvelles sommes. Ceci permet notamment le calcul des intégrales de Feynman et est par exemple utilisé au LHC (CERN).

On mentionnera aussi les récentes collaborations industrielles en robotique et en théorie du contrôle [9]. La combinaison de techniques classiques de résolution de systèmes polynomiaux, notamment le théorème de Kantorovich, avec des implémentations numériques robustes et efficaces (dans Julia) ont permis le passage à l'échelle, nécessaire pour de telles applications.

Contributeurs :

Jérémy Berthomieu et Khalil Ghorbal

Références

1 Lorenzo Baldi, Bernard Mourrain, and Adam Parusiński. On Łojasiewicz inequalities and the effective Putinar's Positivstellensatz. *Journal of Algebra*, 662 :741–767, 2025. doi:10.1016/j.jalgebra.2024.08.022.

2 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, 2006. doi:10.1007/3-540-33099-2.

3 Carlos Beltrán and Luis Miguel Pardo. Smale's 17th problem : average polynomial time to compute affine and projective solutions. *J. Amer. Math. Soc.*, 22(2) :363–385, 2009. doi:10.1090/S0894-0347-08-00630-9.

4 Alin Bostan, Frédéric Chyzak, Pierre Lairez, and Bruno Salvy. Generalized Hermite reduction, creative telescoping and definite integration of D-finite functions. In Éric Schost, editor, *ISSAC'18*, pages 95–102. ACM Press, 2018. doi:10.1145/3208976.3208992.

7. https://maplesoft.com, https://magma.maths.usyd.edu.au/magma
8. https://www.sagemath.org, https://macaulay2.com
9. Safran Electronics & Defense, https://pace.gitlabpages.inria.fr

5 Alin Bostan and Manuel Kauers. The complete Generating Function for Gessel Walks is Algebraic. *Proceedings of the American Mathematical Society*, 138, 2010. doi:10.1090/s0002-9939-2010-10398-2.

6 Javad Doliskani, Pascal Giorgi, Romain Lebreton, and Éric Schost. Simultaneous conversions with the residue number system using linear algebra. *ACM Trans. Math. Softw.*, 44(3), January 2018. doi:10.1145/3145573.

7 Michael Eisermann. The Fundamental Theorem of Algebra Made Effective : An Elementary Real-algebraic Proof via Sturm Chains. *The American Mathematical Monthly*, 119(9) :pp. 715–752, 2012. doi:10.4169/amer.math.monthly.119.09.715.

8 Jean-Charles Faugère and Chenqi Mou. Sparse FGLM algorithms. *Journal of Symbolic Computation*, 80 :538–569, 2017. doi:https://doi.org/10.1016/j.jsc.2016.07.025.

9 Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Théry. A Machine-Checked Proof of the Odd Order Theorem. In *Interactive Theorem Proving*, pages 163–179. Springer, 2013. doi:10.1007/978-3-642-39634-2_14.

10 David Harvey and Joris van der Hoeven. Integer multiplication in time $O(n \log n)$. *Annals of Mathematics*, 193(2) :563 – 617, 2021. doi:10.4007/annals.2021.193.2.4.

11 Pierre Lairez. A Deterministic Algorithm to Compute Approximate Roots of Polynomial Systems in Polynomial Average Time. *Found. Comput. Math.*, 17(5) :1265–1292, October 2017. doi:10.1007/s10208-016-9319-7.

12 Henri Lombardi, Daniel Perrucci, and Marie-Françoise Roy. An elementary recursive bound for effective Positivstellensatz and Hilbert 17th problem. *Memoirs of the American Mathematical Society*, 263, 2020.

13 Guillaume Moroz. New data structure for univariate polynomial approximation and applications to root isolation, numerical multipoint evaluation, and other problems. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1090–1099, 2022. doi:10.1109/FOCS52979.2021.00108.

14 Vincent Neiger and Clément Pernet. Deterministic computation of the characteristic polynomial in the time of matrix multiplication. *Journal of Complexity*, 67 :101572, 2021. doi:10.1016/j.jco.2021.101572.

15 Mohab Safey El Din and Éric Schost. A Nearly Optimal Algorithm for Deciding Connectivity Queries in Smooth and Bounded Real Algebraic Sets. *J. ACM*, 63(6), January 2017. doi:10.1145/2996450.

16 William Y. Sit. *The Ritt-Kolchin Theory for Differential Polynomials*, pages 1–70. World Scientific Publishing, 2002. doi:10.1142/9789812778437_0001.

Les vingt ans du GdR IFM, vus du GT « Arithmétique des ordinateurs »

L'arithmétique des ordinateurs a pour but de travailler sur les représentations des nombres, les algorithmes de calcul et les architectures pour calculer plus vite, plus précisément, à moindre coût ou de façon plus sûre. Ce domaine s'appuie sur les mathématiques, l'informatique et l'architecture des circuits. Les domaines d'application sont très variés : calcul scientifique, cryptographie, preuves assistées par ordinateur, traitement du signal, spatial... La communauté du GT-ARITH s'est structurée ces 20 dernières années autour de plusieurs thématiques.

1 Maîtriser la précision numérique

Une thématique importante du GT ARITH est la **maîtrise de la précision numérique**. Bien que ce sujet soit ancien, il est primordial pour les académiques et les industriels de savoir si leurs programmes calculent bien. Cela inclut à la fois une borne sur les erreurs d'arrondis et une vérification qu'il n'y a pas de comportement exceptionnel (par exemple *overflow* ou division par zéro). Un large choix d'outils a été développé sur cette thématique, sur des bases statistiques (CADNA), d'interprétation abstraite (Fluctuat, EVA) ou d'arithmétique d'intervalles (Gappa). Hors des cas d'applications de ces outils, la littérature s'est enrichie de nombreuses preuves d'algorithmes ou de théorèmes flottants. Un tel exemple récent est un algorithme d'estimation rapide et fiable du risque de collision spatiale, actuellement embarqué dans le mini-satellite expérimental OPS-SAT de l'Agence spatiale européenne [1]

Une évolution dans cette thématique est l'apparition de démonstrations formelles de ces résultats (bibliothèque Flocq [2] notamment) qui permet plus de garanties sur la correction du programme ou de l'algorithme.

Avec le développement de ces outils, la maîtrise de la précision numérique a pu passer à l'échelle avec des applications HPC (high-performance computing). D'ailleurs, le GT ARITH est devenu un GT commun entre le GdR IFM et le GdR C4P (Calcul : Paradigmes, Parallélisme, Performance, Précision) créé très récemment.

2 Arithmétique virgule-flottante

L'arithmétique **virgule-flottante** est un outil fondamental commun aux membres du GT ARITH. Afin de présenter le plus clairement possible le standard IEEE-754 [1], mais aussi de motiver des améliorations potentielles à y apporter, un groupe dirigé par Jean-Michel Muller a rédigé une somme [12] qui est actuellement considérée comme la référence internationale sur le sujet.

Une des améliorations significatives portée par une partie des membres du GT est la promotion de l'arrondi correct lors de l'évaluation de larges classes de fonctions mathématiques (essentiellement, les fonctions élémentaires). Au cours des 20 dernières années, des progrès substantiels ont été faits par plusieurs membres du GT. Ils permettent à présent l'évaluation des fonctions élémentaires avec arrondi correct en précision double (la précision de prédilection du calcul scientifique).

1. https ://lejournal.cnrs.fr/articles/un-algorithme-pour-eviter-les-debris-spatiaux

Le GT a également été fortement influencé par les **évolutions matérielles** des fabricants de puces. Il y a 20 ans, les formats disponibles et répandus étaient le *binary32* et le *binary64*, anciennement dénommés simple et double précisions. Même si la multiprécision en logiciel (MPFR) était disponible, le matériel se concentrait sur ces deux formats. Avec l'essor de l'IA, un grand nombre de formats plus petits sont maintenant disponibles, notamment sur les GPUs (*fp16, bfloat16, fp8, fp6, fp4*). En lien avec la maîtrise de la précision numérique, il est utile de savoir quels algorithmes fonctionnent encore à si petite précision et comment émuler une précision plus grande.

On peut aussi utiliser ces formats à bon escient en essayant d'optimiser la vitesse d'exécution, la mémoire ou la quantité de données échangées (en HPC), c'est le but de la *mixed-precision*.

Tout cela reflète le **lien fort entre matériel et logiciel** dans ce GT puisque les évolutions matérielles influent sur nos applications mais nos résultats influent sur le matériel (usage du *round-to-odd* notamment).

En plus de ces formats fondés sur la virgule flottante, se sont développés d'**autres formats de nombres** plus adaptés à certaines applications. Par exemple, LNS (*Logarithmic Number System*) stocke une approximation du logarithme du nombre, rendant les multiplications simples et les additions complexes [13]. D'autres formats plus exotiques ont été développés sur la période. On peut citer la famille de format posits qui ont une taille variable de la mantisse et de l'exposant, ce qui les rend plus dynamiques mais bien plus complexes à implémenter en matériel [4].

Une autre application où la précision numérique est importante est celle des **filtres numériques**. Ces programmes sont composés de calculs (en virgule flottante ou fixe) à l'intérieur d'une boucle. Le nombre important d'itérations rend les bornes d'erreurs difficiles à trouver et à prouver. Sur les filtres LTI (*Linear Time-Independant*), ce problème a été résolu avec une borne d'erreur prouvée optimale [8]. Autour de cette application, plusieurs outils ont été réalisés pour estimer les intervalles dynamiques et la qualité numérique nécessaire pour une implémentation en C ou sur FPGA [16].

3 Arithmétique et cryptographie

Les activités du GT-ARITH trouvent aussi des liens importants avec le GT-C2. Les approches **arithmétiques pour la cryptographie** ont pour but de permettre des calculs rapides et sûrs sur des grands nombres (au plus quelques milliers de bits) dans des corps finis. Dans le cadre de la cryptographie classique, les opérations modulaires sont très présentes. Ce travail reste pertinent avec l'évolution des standards cryptographiques et de l'architecture des processeurs. À titre d'exemple les processeurs Intel proposent des instructions pour calculer sur des corps finis de caractéristique deux depuis quelques années déjà.

Certaines représentations sont utiles dans ce domaine. Les représentations **RNS** (Residue Number System) et **PMNS** (Polynomial Modular Number Systems), qui ont l'avantage de paralléliser certaines opérations, offrent de bonnes performances dans ce contexte. Par exemple un des algorithmes-phares en cryptographie homomorphe, l'algorithme CKKS [9], nécessite des calculs sur des polynômes de très haut degré avec des coefficients de très grande taille et tire avantage de ces représentations. De plus, sous certaines conditions, ces représentations sont redondantes et permettent plusieurs écritures pour chaque valeur. Il devient alors possible d'introduire de l'aléa dans les écritures et d'effectuer des calculs dont l'exécution sur processeur **réduit les fuites** pouvant être exploitées par un attaquant [7, 3].

Plus généralement, l'algorithmique des **réseaux euclidiens** joue un rôle-clé en cryptographie à clé publique depuis une trentaine d'années. Ainsi, l'algorithme LLL, fondamental au sein de ce domaine, est un outil de cryptanalyse essentiel qui a été rendu pratique grâce aux implémentations flottantes [15, 11] actuellement disponibles dans le logiciel `fplll`. Depuis une quinzaine d'années, cette algorithmique est également au cœur de la conception de nouvelles primitives cryptographiques. En effet, avec la menace de plus en plus prégnante que représente l'apparition d'un ordinateur quantique, les techniques cryptographiques à clé publique classiques ne sont plus sûres. C'est pourquoi, le NIST (National Institute of Standards and Technology) a ouvert en 2017 un concours pour développer des algorithmes résistant à cette menace. Ces nouveaux algorithmes, dits **post-quantiques**, s'appuient sur de nouveaux problèmes (notamment dans des réseaux euclidiens, mais aussi sur des codes correcteurs, sur des isogénies, etc.). De nouvelles approches arithmétiques ont démontré la possibilité d'implémentations rapides et sûres de plusieurs propositions à ce concours.

4 Arithmétique et opérateurs matériels

Les activités du GT-ARITH sont très liées au support matériel et suivent deux approches. La première consiste à exploiter au mieux les ressources offertes par les processeurs modernes. Ce sont des approches logicielles qui cherchent à utiliser le parallélisme et l'hétérogénéité de ces processeurs. L'autre approche consiste à développer des **architectures d'opérateurs** ou d'applications complètes sur FPGA ou ASIC.

Ces 20 dernières années des outils pour réaliser des implémentations matérielles ont été développés dans la communauté du GT-ARITH. Dans les implémentations matérielles, la précision n'est pas fixe et peut être librement choisie par le développeur. La maîtrise de la qualité numérique en **virgule fixe** et le choix de la précision optimale est difficile mais permet de construire des architectures de petite taille, basse consommation ou encore pour du calcul approché [10].

Par exemple, **FloPoCo** [6, 5] est un générateur de cœurs arithmétiques pour différentes technologies cible (FPGA, VLSI, ...). Le premier but de cet outil est de concevoir des opérateurs ad-hoc permettant d'obtenir des résultats plus précis avec moins de matériel et en moins de temps. Le deuxième but est de permettre un calcul juste. Les opérateurs FloPoCo sont soigneusement conçus pour garantir qu'aucun bit inutile pour le résultat final ne soit calculé.

Une troisième approche intermédiaire est explorée en adaptant le jeu d'instructions des processeurs. Le jeu d'instructions ouvert `Risc-V` prévoit la customisation de son jeu d'instruction. Cette approche permet d'utiliser des arithmétiques adaptées à l'application visée [14].

Contributeurs et contributrices :

Sylvie Boldo, Nicolas Brisebarre et Laurent-Stéphane Didier.

Références

1 IEEE standard for floating-point arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pages 1–84, July 2019. doi:10.1109/IEEESTD.2019.8766229.

2 Sylvie Boldo and Guillaume Melquiond. *Computer Arithmetic and Formal Proofs : Verifying Floating-point Algorithms with the Coq System.* ISTE Press - Elsevier, December 2017.

3 Jérôme Courtois, Lokman Abbas-Turki, and Jean-Claude Bajard. Resilience of randomized RNS arithmetic with respect to side-channel leaks of cryptographic computation. *IEEE Transactions on Computers*, 68(12) :1720–1730, 2019.

4 Florent De Dinechin, Luc Forget, Jean-Michel Muller, and Yohann Uguen. Posits : the good, the bad and the ugly. In *Proceedings of the Conference for Next Generation Arithmetic 2019*, pages 1–10, 2019.

5 Florent De Dinechin and Martin Kumm. Application-specific arithmetic. *Cham : Springer International Publishing*, 2024.

6 Florent De Dinechin and Bogdan Pasca. Designing custom arithmetic data paths with FloPoCo. *IEEE Design & Test of Computers*, 28(4) :18–27, 2011.

7 Laurent-Stéphane Didier, Fangan-Yssouf Dosso, Nadia El Mrabet, Jérémy Marrez, and Pascal Véron. Randomization of arithmetic over polynomial modular number system. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pages 199–206. IEEE, 2019.

8 T. Hilaire. From filters/controllers to code – contributions to fixed-point arithmetic implementations under accuracy constraint. Habilitation à Diriger des Recherches (HDR) – Sorbonne Université, 2024.

9 Joon-Woo Lee, Eunsang Lee, Yongwoo Lee, Young-Sik Kim, and Jong-Seon No. High-precision bootstrapping of rns-ckks homomorphic encryption using optimal minimax polynomial approximation and inverse sine function. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 618–647. Springer, 2021.

10 D. Menard, R. Rocher, O. Sentieys, N. Simon, L-S. Didier, T. Hilaire, B. Lopez, E. Goubault, S. Putot, F Vedrine, A. Najahi, G. Revy, L Fangain, C. Samoyeau, F Lemonnier, and C. Clienti. Design of fixed-point embedded systems (defis). In ECSI European Electronic Chips and Systems Initiative, editors, *2012 Conference on Design and Architectures for Signal and Image Processing (DASIP), Karlsruhe, Germany, October 23 - 25, 2012*, pages 365–366. ECSI - European Electronic Chips and Systems design Initiative, 2012.

11 Ivan Morel, Damien Stehlé, and Gilles Villard. H-LLL : using Householder inside LLL. In Jeremy R. Johnson, Hyungju Park, and Erich L. Kaltofen, editors, *Symbolic and Algebraic Computation, International Symposium, ISSAC 2009, Seoul, Republic of Korea, July 29-31, 2009, Proceedings*, pages 271–278. ACM, 2009. URL : `https://doi.org/10.1145/1576702.1576740`, `doi:10.1145/1576702.1576740`.

12 Jean-Michel Muller, Nicolas Brunie, Florent de Dinechin, Claude-Pierre Jeannerod, Mioara Joldes, Vincent Lefèvre, Guillaume Melquiond, Nathalie Revol, and Serge Torres. *Handbook of floating-point arithmetic*. Birkhäuser/Springer, Cham, second edition, 2018. URL : `https://doi.org/10.1007/978-3-319-76526-6`, `doi:10.1007/978-3-319-76526-6`.

13 Jean-Michel Muller, Alexandre Scherbyna, and Arnaud Tisserand. Semi-logarithmic number systems. *IEEE Trans. Computers*, 47(2) :145–151, 1998. URL : `https://doi.org/10.1109/12.663760`, `doi:10.1109/12.663760`.

14 Geneviève Ndour, Tiago Trevisan Jost, Anca Molnos, Yves Durand, and Arnaud Tisserand. Evaluation of variable bit-width units in a RISC-V processor for approximate computing. In Francesca Palumbo, Michela Becchi, Martin Schulz, and Kento Sato, editors, *Proceedings of the 16th ACM International Conference on Computing Frontiers, CF 2019, Alghero, Italy, April 30 - May 2, 2019*, pages 344–349. ACM, 2019. URL : `https://doi.org/10.1145/3310273.3323159`, `doi:10.1145/3310273.3323159`.

15 Phong Q. Nguyen and Damien Stehlé. An LLL algorithm with quadratic complexity. *SIAM J. Comput.*, 39(3) :874–903, 2009. URL : `https://doi.org/10.1137/070705702`, `doi:10.1137/070705702`.

16 R. Rocher, D. Menard, O. Sentieys, and P. Scalart. Accuracy evaluation of fixed-point based LMS algorithm. *Digital Signal Processing*, 20(3) :640–652, 2010. URL : `http://hal.inria.fr/inria-00450935/en/`, `doi:doi:10.1016/j.dsp.2009.10.007`.

Les vingts ans du GdR IFM, vus du GT « Codage et cryptographie »

Ces vingt dernières années ont été marquées par une accélération spectaculaire de la recherche dans les thématiques de la cryptographie et des codes correcteurs d'erreurs avec l'émergence de nombreux sujets nouveaux. Les applications de la cryptographie ont explosé et beaucoup d'outils théoriques datant de plus de 40 ans sont maintenant utilisés en pratique. La communauté internationale s'est très largement développée et le GT C2, commun aux GdR IFM et SI (Sécurité Informatique), a fait de même. Son nombre de membres a considérablement augmenté et le GT s'est structuré avec les journées C2 (devenues annuelles depuis 2025, réunissant plus de 200 participants par édition) et son séminaire (trois journées pleines par an, dont une hors de Paris).

De nouveaux algorithmes et de nouvelles manières de prouver des calculs sont apparus, certains transformant totalement la recherche. Plusieurs avancées viennent de résultats et techniques en codes correcteurs d'erreurs, rapprochant ainsi ces deux parties de la communauté, notamment sur les problèmes de cryptographie à base de codes correcteurs d'erreurs.

Les recherches du GT ont été menées dans tous les sous-thèmes fondateurs de C2, tels que la cryptographie symétrique, asymétrique et les codes correcteurs d'erreurs. Les membres du GT se sont fortement mobilisés sur toutes ces thématiques, en partie aiguillonnés par les compétitions de standardisation du NIST et le développement du quantique. Dans la suite, les avancées sont présentées selon ce découpage thématique, mais plusieurs résultats sont transversaux ou mobilisent des outils propres à plusieurs sous-thèmes.

1 Cryptographie symétrique.

Ce domaine a été structuré principalement par les compétitions du NIST (*National Institute of Standards and Technology*) et la conception de nouveaux schémas plus efficaces. Depuis 2001, l'Advanced Encryption Standard (AES) est largement déployé pour remplacer le DES, notamment grâce aux instructions matérielles présentes dans certains processeurs. Après 25 ans de cryptanalyse, seulement 7 tours parmi les 10 de la version avec une clef de 128 bits sont atteignables par la cryptanalyse, les versions de 192 et 256 bits présentant d'autres faiblesses.

Au début des années 2000, une série d'attaques contre les fonctions de hachage a été proposée (attaques différentielles contre MD4, MD5, SHA-1 et contre le mode opératoire de Merkle-Damgård). À noter qu'un fort impact du GT fut la première collision sur SHA-1 [13], amenant au retrait de cette fonction de hachage de plusieurs standards. Pour sélectionner une nouvelle fonction, la compétition SHA-3 a été lancée, et son vainqueur, KECCAK [3], a introduit la structure éponge utilisant une permutation publique et pouvant avoir une sortie de taille arbitraire. Cette construction permet d'obtenir différentes primitives cryptographiques comme des fonctions pseudo-aléatoire à sortie extensible (XOF), tel SHAKE, des codes d'authentification de messages, et des schémas de chiffrement authentifié.

Depuis 2010, les compétitions de chiffrement léger pour les environnements contraints, de chiffrement authentifié (devenu une norme de facto après toutes les attaques contre TLS dans les années 2010), et de fonction de hachage pour les mots de passe, ont aussi amené des progrès remarquables avec les vainqueurs ASCON, SCRYPT et ARGON2. De plus, la problématique de la cryptographie à bas coût à utiliser dans des systèmes ultra-légers a renouvelé certains critères de performance et en a défini de nouveaux, par exemple la faible latence. Enfin, aujourd'hui la recherche s'oriente vers de nouvelles primitives symétriques

efficaces sur $\mathbb{F}_p$, plus adaptées à un usage combiné avec les systèmes de preuves, les signatures, ou le chiffrement homomorphe. Très naturellement, ces travaux, accompagnés de leur volet cryptanalytique, ont vu émerger de nouveaux critères d'évaluation de performances et de nouvelles classes d'analyse (« cryptanalyse » ou « attaques ») : attaques par invariants linéaire et non-linéaire, attaques différentielles à clef fixée...

Du côté de la cryptanalyse, des progrès remarquables ont ainsi été obtenus sur plusieurs fronts. De nouvelles techniques ont émergé, comme les algorithmes reposant sur la *division property* qui permettent des attaques intégrales plus précises contre Misty1 [14], l'utilisation de MILP (Mixed Integer Linear Programming) pour rechercher automatiquement les attaques. Enfin, les attaques génériques exploitant les graphes des fonctions ont permis des avancées importantes à la fois en termes de cryptanalyse et de preuves de sécurité. Du côté des preuves, des progrès ont été effectués pour obtenir des schémas et modes opératoires dont la sécurité est supérieure à la borne dérivée du paradoxe des anniversaires grâce à des outils avancés de probabilités.

Enfin l'éventuelle arrivée de l'ordinateur quantique a conduit la communauté à définir un cadre essentiellement quantique pour la cryptanalyse (« superposition ») dans lequel on peut considérer de nouvelles classes d'attaques.

2 Codes correcteurs d'erreurs : évolution des problèmes.

Au cours des vingt dernières années, les codes correcteurs ont connu une évolution marquante : initialement conçus pour résoudre des problèmes classiques de communication et assurer la fiabilité des transmissions, ils sont progressivement devenus des outils de première importance dans des domaines beaucoup plus larges. Leur structure mathématique s'est révélée particulièrement adaptée à la cryptographie moderne, notamment dans la construction de systèmes basés sur les preuves à divulgation nulle de connaissance, où l'on exploite leurs propriétés de robustesse et de cohérence [2]. Ces outils sont très demandés par exemple pour améliorer les blockchains. Parallèlement, l'essor du stockage distribué et de la nécessité de garantir l'intégrité des données a conduit au développement de codes localement testables et localement décodables, permettant de vérifier ou de récupérer une information en n'accédant qu'à une petite partie des données [6]. Ces travaux illustrent à quel point la théorie des codes a évolué et s'est diversifiée. Enfin, dans le monde quantique lui-même, des constructions de bons codes quantiques (codes LDPC quantiques, codes de Tanner quantiques) ont été proposées, avec plusieurs percées fondamentales. Ces codes quantiques sont essentiels pour lutter contre la décohérence et le bruit inhérents au contexte quantique, et permettre la réalisation de l'ordinateur quantique.

3 Cryptographie post-quantique.

La menace de l'ordinateur quantique a déclenché une révolution dans le domaine, explicitement concrétisée en 2016 par l'appel du NIST à la conception de systèmes dits post-quantiques. En effet, l'algorithme de Shor [12] permettrait, avec un ordinateur quantique suffisamment puissant et efficace, d'attaquer une grande partie des constructions asymétriques utilisées aujourd'hui dont la sécurité repose sur les problèmes de la factorisation et du logarithme discret. De nouveaux problèmes conjecturés difficiles aussi pour les ordinateurs quantiques, alternatifs à la factorisation et au logarithme discret, ont été très étudiés ces dernières années. Ils reposent sur les réseaux euclidiens, les codes, les isogénies entre courbes, et les systèmes polynomiaux multivariés. Les fonctions de hachage permettent aussi de construire des schémas

de signature dits post-quantiques. Enfin, de nouveaux schémas issus de techniques de calcul sécurisé multipartite (MPC) ont permis d'obtenir des signatures très efficaces.

Dans ce contexte, le NIST a organisé plusieurs compétitions pour mettre en place de nouveaux standards de chiffrement et de signature à clé publique. Proposer un candidat à de tels appels demande un travail important de conception, détermination des paramètres, preuve de sécurité, programmation d'implémentation de référence et mise en place de vecteurs de tests et de spécification. Ces compétitions se déroulent en plusieurs étapes, avec une élimination de candidats à chaque étape jusqu'au choix des finalistes. La première compétition, qui a commencée en 2016, a abouti à la sélection de cinq premiers standards : le mécanisme d'encapsulation de clé (et chiffrement) CRYSTALS-Kyber [4] (maintenant standardisé sous le nom de ML-KEM), la signature CRYSTALS-Dilithium [7] (standardisée sous le nom de ML-DSA), la signature Falcon (encore en cours de standardisation), la signature Sphincs+ (standardisée sous le nom de SLH-DSA) et le mécanisme d'encapsulation de clé (et chiffrement) HQC (encore en cours de standardisation). Parmi ces constructions, les trois premières font reposer leur sécurité sur des problèmes difficiles sur les réseaux euclidiens, la dernière sur un problème issu de la théorie des codes.

Historiquement, le cryptosystème de McEliece, quasi-contemporain de RSA, est fondateur de la cryptographie à clé publique à base de codes correcteurs. Mais, depuis une vingtaine d'années, la cryptographie reposant sur les réseaux euclidiens s'est imposée comme la solution la plus aboutie et mieux comprise pour remplacer la cryptographie asymétrique utilisée aujourd'hui. En plein essor depuis les travaux de Regev en 2005 [10], elle permet de construire des schémas de chiffrement et de signature à la fois sûrs et efficaces, et également des constructions avec des fonctionnalités avancées allant jusqu'au chiffrement complètement homomorphe.

Suite à la première compétition, une seconde a été lancée en 2023 pour proposer d'autres signatures avec comme contrainte que leur sécurité ne devrait pas reposer sur les problèmes difficiles sur les réseaux euclidiens. Certaines des soumissions proposées à cette compétition, qui est encore en cours, reposent sur des problèmes issus du domaine des codes correcteurs, d'autres sur le problème de la résolution d'équations algébriques.

Enfin, pour la cryptographie fondée sur les isogénies, une attaque dévastatrice [5, 9, 11] contre le schéma de chiffrement SIKE en 2022 a finalement été transformée en outil de construction puissant, permettant la construction de nouveaux schémas beaucoup plus efficaces, comme par exemple la signature SQIsign, et remettant sur le devant de la scène les courbes elliptiques.

La recherche a encore des progrès à faire pour mieux comprendre la sécurité de ces schémas, obtenir des schémas plus efficaces, des implémentations sûres ou des constructions avancées. En particulier, la cryptanalyse quantique progresse pour améliorer l'algorithme de Shor et comprendre la sécurité quantique des nouvelles hypothèses.

4 Cryptographie classique.

Ces nouvelles considérations ne doivent pas nous faire négliger la cryptographie classique à base de logarithme discret (sur les corps finis ou sur les courbes elliptiques) ou de factorisation qui est quasiment la seule utilisée en pratique. Dans ces domaines, le problème du logarithme discret sur les corps finis s'est finalement révélé facile à résoudre dans les corps de petite caractéristique, grâce un algorithme « quasi polynomial » [1]. Ces attaques ont été conduites en pratique dans des réalisations logicielles. En particulier, un effort continu sur deux décennies a assuré la permanence et la qualité du logiciel CADO-NFS de factorisation, qui a

notamment permis d'obtenir des records toujours inégalés de factorisation de clé publique RSA (250 chiffres, 828 bits).

Depuis les années 2000, la cryptographie à base de logarithme discret sur les courbes elliptiques remplace RSA. Le coup de grâce est donné en 2018 avec le passage à TLS 1.3 qui promeut la notion de *forward secrecy* pour le chiffrement, même si RSA reste utilisé pour les signatures dans les certificats. Les courbes sécurisées et les implémentations sur Internet utilisent les modèles d'Edwards EdDSA et X25519. Les courbes elliptiques présentant un couplage permettent de concevoir de nouveaux schémas avec des fonctionnalités avancées comme le chiffrement basé sur les attributs ou fonctionnel. La recherche pour la sécurité des courbes, l'efficacité des couplages, le hachage vers les courbes, a ouvert de nouveaux sujets de recherche. Les courbes permettent aujourd'hui de concevoir des systèmes de preuves succinctes et non-interactives, comme celles utilisées dans certaines blockchains.

5 Sécurité des calculs.

Une petite révolution est apparue en 2009 quand Gentry a montré qu'il était possible de construire des schémas complètement homomorphes [8], offfrant ainsi la possibilité de calculer sur des données chiffrées sans les déchiffrer. Les constructions existantes d'un tel chiffrement reposent toutes sur les hypothèses de réseaux euclidiens. Plusieurs générations de systèmes ont été proposées et le domaine est très actif. La cryptographie symétrique a aussi proposé des constructions efficaces avec le chiffrement cherchable, grâce auquel il est possible d'effectuer des recherches par mots-clés dans une base de données chiffrée. Enfin, le calcul sécurisé entre plusieurs parties a effectué des progrès impressionnants, permettant son utilisation pratique et la protection de la vie privée dans différentes applications. Il permet à un groupe de participants, chacun détenant son propre secret, de calculer une fonction de tous leurs secrets, par exemple le max, sans qu'aucun participant ne révèle aux autres son propre secret. Il a été montré au niveau international, que contrairement aux idées reçues, le calcul multipartite est performant en pratique, et il en a suivi un développement accéléré de la recherche dans ce domaine. Comme mentionné précédemment, le calcul multipartite sécurisé a aussi servi de concept clé pour la construction d'algorithmes de signature, à travers le paradigme *MPC in the head*.

6 Implémentations sécurisées.

Depuis la fin des années 90, les attaques par canaux auxiliaires forment un domaine de recherche très important au sein de la communauté CHES. Aujourd'hui, l'implémentation des systèmes classiques est bien comprise et les preuves de sécurité par masquage, à la suite des travaux fondateurs de Ishai, Sahai et Wagner, et différents modèles de sécurité, ont permis des avancées majeures. Dans ce domaine, les outils statistiques d'apprentissage ont eu une influence très importante. L'utilisation des outils formels a aussi amené de réels progrès pour la vérification des preuves, jusqu'à l'implémentation de la cryptographie. Enfin, les attaques logicielles ont aussi fait leur apparition depuis 2005 et les mécanismes matériels d'accélération sont également finement étudiés.

Contributeurs et contributrices.

Daniel Augot, Pierre-Alain Fouque, Eleonora Guerrini et Adeline Roux-Langlois.

Références

1 Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 1–16. Springer, Berlin, Heidelberg, May 2014. doi:10.1007/978-3-642-55220-5_1.

2 Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable zero knowledge with no trusted setup. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 701–732. Springer, Cham, August 2019. doi:10.1007/978-3-030-26954-8_23.

3 Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 313–314. Springer, Berlin, Heidelberg, May 2013. doi:10.1007/978-3-642-38348-9_19.

4 Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber : A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367. IEEE Computer Society Press, April 2018. doi:10.1109/EuroSP.2018.00032.

5 Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 423–447. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_15.

6 Alexandros G. Dimakis, Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, and Kannan Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Inf. Theory*, 56(9) :4539–4551, 2010.

7 Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium : A lattice-based digital signature scheme. *IACR TCHES*, 2018(1) :238–268, 2018. URL : https://tches.iacr.org/index.php/TCHES/article/view/839, doi:10.13154/tches.v2018.i1.238-268.

8 Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. doi:10.1145/1536414.1536440.

9 Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 448–471. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_16.

10 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. doi:10.1145/1060590.1060603.

11 Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 472–503. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_17.

12 Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5) :1484–1509, 1997.

13 Marc Stevens, Pierre Karpman, and Thomas Peyrin. Freestart collision for full SHA-1. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 459–483. Springer, Berlin, Heidelberg, May 2016. doi:10.1007/978-3-662-49890-3_18.

14 Yosuke Todo. Integral cryptanalysis on full MISTY1. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 413–432. Springer, Berlin, Heidelberg, August 2015. doi:10.1007/978-3-662-47989-6_20.