Editorial: Hardware Architectures for Algebra, Cryptology and Number Theory

Kris Gaj[a], Rainer Steinwandt[b]

[a]*ECE Department, George Mason University, U.S.A.*
[b]*Department of Mathematical Sciences, Florida Atlantic University, U.S.A.*

Over the past few years, the interaction of research in computer engineering with research in algebra and number theory has intensified. This interaction is especially visible in cryptography and cryptanalysis, but covers also other areas, such as bioinformatics, coding theory, and image processing. This special issue attempts to explore this interaction, by highlighting recent advances in the development of efficient hardware architectures for algebra, cryptology, and number theory.

The call for papers for this special issue listed several examples of topics of interest and resulted in a pool of 30 submissions, from which we selected the papers finally comprising this special issue. At the end of a comprehensive review and revision process stood five papers, resulting in an acceptance rate of less than 17%. Independent of the final outcome of the review process, we would like to express our appreciation to all submitting authors for their valuable contributions. For manuscripts that in the end could not be included, we hope that the reviewer feedback was perceived by the authors as constructive and helpful. Numerous external reviewers generously offered their expertise and time, and we greatly appreciate their invaluable help in the evaluation process.

Below, we provide a quick bird's-eye view of the research fields of interest to this issue, and show how the five selected papers fit in this view, and enhance and enrich what was accomplished so far.

Modern cryptography is based on the use of two major classes of cryptosystems. Symmetric key ciphers, with a cryptographic key shared between sender and receiver, are used for bulk encryption of data at speeds reach-

Email addresses: kgaj@gmu.edu (Kris Gaj), rsteinwa@fau.edu (Rainer Steinwandt)

ing several Gbits/s in software and up to hundreds of Gbits/s in hardware. Public key cryptosystems, which do not require sharing a secret key, are much slower in nature and are used primarily for the secure exchange of keys for symmetric key ciphers and for the generation and verification of digital signatures.

Number theory and algebra have always been instrumental in the design and security evaluation of public key ciphers. One of the most famous schemes, named after the initials of its inventors Rivest, Shamir, and Adleman, is a prime example for this influence. RSA was perhaps one of the first practical applications of some fundamental and timeless problems of number theory and algebra, such as distinguishing prime numbers from composites, factoring of large integers, and performing basic arithmetic operations on large integers. Right from the inception of their ingenious scheme in the late 1970s, the designers of RSA faced the problem of implementing their cipher fast enough to make it practical. Since the microprocessors of the 1980s were not up to the task, turning to integrated circuits was a natural path to follow. Initial attempts in this direction eventually paved a way for a series of future implementations and helped to shape the field of cryptographic engineering. This field combines techniques from algebra, cryptology, and number theory with methods from digital system design using the two most prevalent semiconductor technologies of today: ASICs (Application Specific Integrated Circuits) and FPGAs (Field Programmable Gate Arrays).

In the mid 1980s a new mathematical platform found its way into the cryptographic community. Elliptic Curve Cryptography (ECC), discovered independently by Koblitz and Miller, offers public key cryptosystems using remarkably small key sizes. They have the advantage of providing the same level of security as earlier schemes, such as RSA, using much smaller operands, and as a result much smaller and cheaper digital circuits. ECC is based on more advanced concepts of algebra and number theory, such as Galois fields, operations in the group of points on an elliptic curve, and the elliptic curve discrete logarithm problem. The two most commonly used Galois fields are binary fields, GF(2^m), and prime fields, GF(p). The efficient implementation of ECC involves optimizing basic operations at the three major levels: i) in the Galois field (multiplication, squaring, inversion), ii) in the group of points on an elliptic curve (addition, doubling), and iii) scalar multiplication of an integer by a point on a curve, kP. In 1989, Koblitz generalized the concept of ECC, by introducing HyperElliptic Curve Cryptography (HECC). The idea here is to obtain cryptosystems that offer even

smaller operand sizes, at the cost of making basic operations at the group level somewhat more complex.

In this issue, we include three articles, describing the state of the art in hardware architectures for three primary classes of public key cryptosystems: RSA, ECC, and HECC.

In *Tripartite Modular Multiplication*, Sakiyama et al. describe a novel scheme for fast modular multiplication, being a basis of RSA and ECC over prime fields. This scheme combines in one hardware architecture three ingenious algorithms: Karatsuba-Ofman multiplication, dating back to an old Russian paper from 1963, Barrett modular reduction, a subject of a Master's thesis defended at Oxford in 1984, and Montgomery modular multiplication, described in Mathematics of Computation by Montgomery in 1985. In 2005, Kaihara earned the best paper award at the CHES (Cryptographic Hardware and Embedded Systems) workshop, by demonstrating a way to parallelize modular multiplication by combining a classical modular multiplier and a Montgomery multiplier. Sakiyama et al. take this idea one step farther, by demonstrating a way of performing modular multiplication using three or more independent computations. Their scheme is suitable for parallel implementations of RSA and ECC using ASICs, FPGAs, and multi-core processors.

In *Optimized FPGA-based Elliptic Curve Cryptography Processor for High Speed Applications*, Järvinen describes the fastest to date FPGA-based ECC processor, capable of verifying over 100,000 digital signatures per second. His architecture takes advantage of multiple optimizations, such as a specific type of elliptic curve (Koblitz curve), fixed field size (GF(2^{163})), fixed polynomial (a NIST-recommended sparse pentanomial), use of τ-adic expansions and Frobenius maps (instead of the traditional double-and-add scalar multiplication), pre-computations, and pipelining. Different values of parameters, and thus different levels of security, can be accomplished by taking advantage of the reconfigurable nature of FPGAs.

In *Design and Design Methods for Unified Multiplier and Inverter and its Application in HECC*, Fan et al. present a novel approach for implementing Hyperelliptic Curve Cryptography (HECC) over GF(2^m). Their approach is based on combining two basic Galois field operations, multiplication and inversion, into one universal unit: Unified Multiplier and Inverter (UMI). The authors demonstrate an advantage of their approach in terms of area, by implementing HECC using only 14.5 kGates in ASIC technology. They also demonstrate an advantage in terms of throughput to area ratio by imple-

menting a high-speed HECC processor using Xilinx FPGAs, with the product latency times area smaller than in any other design reported in the literature to date. The hardware architecture of UMI is parameterized, and allows an optimum choice of the digit size, separately for multiplication and inversion, depending on the features of the specific technology.

Apart from the design of public key cryptosystems (a central subject of modern cryptography) and their efficient implementation (a central subject of cryptographic engineering), one of the most successful applications of algebra and number theory is breaking ciphers. Traditionally, symmetric key ciphers have been broken most efficiently using ASIC-based brute-force codebreaking machines, such as Deep Crack, built in 1998 to break DES (Data Encryption Standard). On the other hand, the attacks against public key ciphers have been implemented most often in software, using supercomputers and/or distributed clusters of computers connected through the Internet. This trend has changed recently in favor of FPGA-based machines, such as COPACOBANA, developed in 2006 by two German university groups, and applied since then to breaking both symmetric key and public key schemes, including DES, A5, KeeLoq, and ECC. The emergence of new designs in this area, traditionally delegated to the basements of national intelligence agencies, has been sped-up by a new workshop, called SHARCS (Special-purpose Hardware for Attacking Cryptographic Systems), started in 2005.

In *Hardware SLE Solvers: Efficient Building Blocks for Cryptographic and Cryptanalytic Applications*, Rupp et al. give a comprehensive overview of hardware architectures for SLE (Systems of Linear Equations) solvers. The investigated architectures are based on the well-known and widely used concept of systolic arrays, introduced by Kung and Leiserson in 1978. Systolic arrays are highly-regular VLSI circuits, built of a small variety of universal cells, arranged into regular one-dimensional or two-dimensional arrays, in which data flows simultaneously in at least two directions. Their primary advantage is the ease of design and high scalability. All designs for SLE solvers presented in the paper by Rupp et al. have been fully implemented in modern FPGAs, and the concrete results are summarized in the paper. The investigated sizes of SLEs (number of equations $\times$ number of variables) vary from 20×20 to 90×90. These sizes are limited only by the amount of resources available in the current generation of FPGAs. SLE solvers as described here are of interest when attacking symmetric key stream and block ciphers, using a new class of codebreaking algorithms called algebraic attacks.

These attacks have been shown to be very successful in breaking stream ciphers, such as the former GSM standard A5/2 or Crypto-1 used in Mifare Classic cards. An additional interesting application of small to medium size SLE solvers is their use in efficient implementations of a new class of digital signature schemes based on multivariate quadratic polynomials.

Apart from the use of algebra and number theory in cryptology, one of the primary practical applications of these research fields is coding theory, dealing with protecting communication against transmission errors. This problem is particularly important in wireless and mobile communications, where the mobility of users increases the probability of errors, and the small size of portable devices imposes restrictions on the complexity, size, and power consumption of encoders and decoders. In *Efficient Hardware Implementation of A Highly-Parallel 3GPP LTE/LTE-Advance Turbo Decoder*, Sun and Cavallaro propose a novel design for a decoder in the emerging 3GPP (3rd Generation Partnership Project) wireless standard called LTE-Advanced (Long Term Evolution – Advanced). This standard is considered one of the most promising 4G wireless technologies, and the commercial services based on it were only recently launched, first in Scandinavia, in December 2009, and then in the United States and Japan, in 2010. One of the primary challenges addressed by the authors of the paper is the design of an efficient interleaver, a part of the decoder that deals with protecting data against burst errors. The authors' design implemented as an ASIC using a 65-nm CMOS technology, achieves a speed in excess of 1 Gbit/s, which is the 4G speed requirement for low mobility communication (e.g., communication between pedestrians).

We hope that this compilation of five papers gives an idea of the fascinating opportunities for interdisciplinary research connecting techniques from algebra, cryptology and number theory with techniques from computer engineering. Possibly, we will have a chance to meet and exchange ideas with readers of this special issue in person at one of the conferences or workshops in the field, such as the annual CHES Cryptographic Hardware and Embedded Systems Workshop.

Finally, we would like to thank Francisco V. Fernández, the Editor-in-Chief of this journal. His encouragement, help and patience throughout the entire review process did not go unnoticed, and we greatly appreciate his support in the completion of this special issue.

Kris Gaj and Rainer Steinwandt