J Cryptogr Eng (2011) 1:1–3
DOI 10.1007/s13389-011-0007-x

EDITORIAL

Introduction to the *Journal of Cryptographic Engineering*

Çetin Kaya Koç

Published online: 2 April 2011

The mathematics, computer science and electrical engineering academic communities began paying attention to cryptography sometime in the mid-1970s. The fruits of the work that was then being performed at MIT and Stanford [1,2] now forms the backbone of communication security, including online shopping, logging into email servers and corporate VPNs, and many other activities we perform over the Internet or via cellular communication networks. The remarkable history of the development of cryptography during the last quarter of the last century consists mainly of fragments in several different books or in the memories of our colleagues. Yet in these 25–30 years, just a handful of people in several universities and research centers in the US and Europe created a set of cryptographic algorithms and protocols that demonstrated the meaning and the measure of cryptographic strength [1–10]. The online security of billions of consumers now depends on the infrastructure developed according to these cryptographic principles.

The story would not be complete, however, if we stopped thinking and talking about it right at this point. The invention of cryptographic algorithms and protocols, particularly public-key algorithms, digital signatures, hash functions and message authentication methods, was essential for the security of our computers, laptops, smart phones and servers. However, equally important are the hardware and software realizations for the platforms on which they run.

There are some significant challenges in high-speed, small-space (circuit size or code space) implementations of cryptographic algorithms, especially public-key cryptographic algorithms. The introduction of the RSA algorithm produced a wave of excitement in the academic community, since exact arithmetic applied to such large numbers had previously been known only to a small community of number-crunching mathematicians, e.g., prime number aficionados and Mersenne number hunters.

Popular interest aside, the network and software engineers of the 1980s who were building the next generation of point-to-point communication protocols, email servers and file-transfer protocols were in great need of software implementations of the RSA public-key encryption and decryption algorithms, and the RSA digital signature algorithm. Thus began the story of a new field, which is still looking for a better descriptive name. As a counterpart to the theory of cryptography, this area covers efficient hardware and software realizations of cryptographic algorithms. I have seen many different names for our field: applied cryptography, algorithm engineering, cryptographic hardware and embedded systems, cryptographic engineering. Each one tries to capture one aspect of this exciting field, which is an overlapped area of mathematics, electrical engineering and computer science.

As soon as cryptographic algorithms were invented or published, papers and reports on their hardware and software realizations appeared in conference proceedings and engineering journals. Interestingly, one of the first papers [11] was written by Ron Rivest, the co-inventor of the RSA public-key cryptosystem. I believe this paper, published in 1980, is the very first paper on RSA implementation. The fact that it is a hardware implementation makes it even more interesting, since one would expect reports of the software implementations (algorithmic, benchmarking, etc.) to come first because they are easier and less costly to implement.

In his paper, Rivest describes a single-chip implementation of the RSA algorithm for up to 512-bit modulus. It is

Ç. K. Koç
Istanbul Şehir University, Istanbul, Turkey
e-mail: koc@sehir.edu.tr

Ç. K. Koç (✉)
University of California, Santa Barbara, USA
e-mail: koc@cs.ucsb.edu

a "simple" implementation in the sense that a 512-bit ALU and eight 512- bit registers around it were used to perform the multiplications and reductions needed for modular multiplication of 512-bit integers. The chip could do almost anything: modular exponentiations, generation of prime numbers from a given random seed, GCD computations, etc., but it never actually worked, as reported subsequently by Rivest [12]. However, I believe this marks the beginning of the quest for efficient hardware and software realizations of cryptography by the academic community.

The following 30 years produced significant amounts of academic research and industrial design work; hardware and software implementations of security suites (SSL, TLS) and devices (cell phones, Bluetooth earphones) and larger systems (SSL and IPSec boxes) were designed, developed and deployed. Phases of this development may be seen as follows:

- *Naive algorithms.* This phase started with the invention of the Diffie–Hellman and the RSA algorithms (1976–1978). Trying to benefit from the developments of computer arithmetic during the previous 25 years, the designers aimed to create hardware and software implementations by simply scaling up the size of numbers they worked with. Since the RSA or the Diffie–Hellman algorithms need 512-bit numbers, 512-bit registers were created to deal with such large numbers. Since they need modular multiplication operations, they first multiply the numbers and then reduce them with multiple subtractions, essentially performing a division by the large modulus. Unfortunately and expectedly, such designs were slow, required large space and were not workable or scalable.
- *The Montgomery algorithm.* Such naive approaches were ended with the invention of the Montgomery algorithm [13]. Peter Montgomery showed that it is possible to perform modular reductions without resorting to the costly division operation. In fact, the entire reduction can be performed with just two multiplications, instead of a division. In the following 10–15 years, as the Montgomery algorithm was better understood and applied [14], very efficient software and hardware realizations of public-key cryptography were designed, and software toolkits such as the RSA Labs BSAFE made public-key cryptography accessible to any software designer. These toolkits made it possible to develop and deploy the popular security suites, such as SSL/TLS during the late 1990s. The popular RSA Labs technical reports [15,16] made the material accessible to hundreds of software and hardware design engineers. In some ways, the contributions of the Montgomery algorithm to cryptographic engineering are similar to the contributions of the Fast Fourier Transform algorithm to digital signal processing. It brought down the complexity of an arithmetic operation which is performed in almost all public-key cryptographic functions, making high-speed implementations possible.
- The last phase which will be with us for some time is the introduction of the side-channel attacks and countermeasures, which approximately followed the introduction and development of the CHES Workshop in the summer of 1999. The timing attacks [17] and power attacks [18] papers made us realize that a cryptographic algorithm implemented in software or hardware is a quite different thing from its description on a piece of paper. While it may be nearly impossible to break the RSA algorithm theoretically and obtain the private exponent, since it requires the factoring of large integers that are beyond our current ability both algorithmically and architecturally, it may be quite easy to obtain the very same private exponent practically, by simply observing the timing or power data from a device performing the RSA signature or decryption operation. Academic work in the field of side-channel attacks flourished in the decade that followed, and it significantly affected the way we design cryptographic software and hardware.
- The first CHES Workshop was held in Worcester, Massachusetts in 12–13 August 1999, founded and organized by Christof Paar and myself. It definitely forms a milestone in the development of cryptographic engineering; however the subsequent success of the CHES Workshop was not at all clear to us at that time. We both realized that there was a need for this event, and we wrote in the preface of the CHES 1999 proceedings [19]:

> As it becomes more obvious that strong security will be an important part of the next generation of communication, computer, and electronic consumer devices, we felt that a new type of cryptographic conference is needed. Our goal was to create a forum which discusses innovative solutions for cryptography in practice.

This analysis is still relevant. The CHES Workshop grew significantly from its modest beginning in 1999, and is now the second-largest cryptography conference after Crypto and the premier forum for presenting scientific advances in all aspects of cryptographic hardware and security of embedded systems. About 300 engineers and scientists from over 30 countries participate in the CHES Workshop, submitting nearly 150 papers every year, with an acceptance rate of less than 20%.

Our journal is yet another step in this development of applied cryptography or cryptographic engineering, however named; it is in no way a conclusion. The mathematics, computer science and electrical engineering academic communities were quite excited with the invention of public-key cryptography in 1976–1978. During the past 35 years, we have all designed, developed and implemented cryptographic solutions in order to provide a security infrastructure for our ubiquitous digital life. Without public-key and secret-key cryptographic algorithms, hash and message authentication functions, true and pseudorandom number generators, and their secure and efficient implementations, this infrastructure would not have been possible.

We, the Editorial Board, the Steering Committee, and the Editor-in-Chief of the *Journal of Cryptographic Engineering*, sincerely hope that our journal will have high-quality research and tutorial papers, will be to your liking, and will be an essential part of your crypto library. I also thank the Springer staff, particularly Alfred Hofmann and his diligent team, for helping us to launch our journal.

References

1. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inform. Theory **22**, 644–654 (1976)
2. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
3. Miller, V.: Uses of elliptic curves in cryptography. In: Williams, H.C. (ed.) Advances in Cryptology—CRYPTO 85, Proceedings. LNCS, vol. 218, pp. 417–426. Springer, Berlin (1985)
4. Koblitz, N.: Elliptic curve cryptosystems. Math. Comput. **48**(177), 203–209 (1987)
5. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, Berlin (1993)
6. National Institute of Standards and Technology. Data Encryption Standard (DES), FIPS 46–3, October 1999
7. National Institute of Standards and Technology. Advanced Encryption Standard (AES), FIPS 197, November 2001
8. Daemen, J., Rijmen, V.: The Design of Rijndael. Springer, Berlin (2002)
9. National Institute of Standards and Technology. Secure Hash Standard (SHS), FIPS 180–3, October 2008
10. National Institute of Standards and Technology. Digital Signature Standard (DSS), FIPS 186–3, June 2009
11. Rivest, R.L.: A description of a single-chip implementation of the RSA cipher. Lambda, 1(Fourth Quarter):14–18 (1980)
12. Rivest, R.L.: RSA chips (Past/Present/Future). In: Beth, T., Cot, N., Ingemarsson, I. (eds.) Advances in Cryptology. Proceedings of Eurocrypt 84. LNCS, vol. 209, pp. 159–165. Springer, Berlin (1984)
13. Montgomery, P.L.: Modular multiplication without trial division. Math. Comput. **44**(170), 519–521 (1985)
14. Koç, Ç.K., Acar, T., Kaliski, B.S. Jr.: Analyzing and comparing montgomery multiplication algorithms. IEEE Micro. **16**(3), 26–33 (1996)
15. Koç, Ç.K.: High-Speed RSA Implementation. Technical Report TR 201, RSA Laboratories, pp. 73, (1994)
16. Koç, Ç.K.: RSA Hardware Implementation. Technical Report TR 801, RSA Laboratories, pp. 30, April 1996
17. Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology—CRYPTO 96. LNCS, vol. 1109, pp. 104–113. Springer, Berlin (1996)
18. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) Advances in Cryptology—CRYPTO 99. LNCS, vol. 1666, pp. 388–397. Springer, Berlin (1999)
19. Koç, Ç.K., Paar, C. (eds.): Cryptographic Hardware and Embedded Systems. First International Workshop, Worcester, MA, USA. LNCS, vol. 1717. Springer, Berlin (1999)