FPGA Implementation of Modular Exponentiation Using Montgomery Method

Elena Trichina
Chalmers University of Technology
S-41296 Goteburg
Sweden
Email: trichina@cs.chalmers.se

19 April 2000

Keywords: Montgomery multiplication, Field Programmable Gate Arrays.

The design of public key cryptography hardware is an active area of research because the speed of cryptographical schemes is a serious bottleneck in many applications. A cheap and flexible modular exponentiation hardware accelerator can be achieved using Field Programmable Gate Arrays (FPGA). FPGA design presented in this paper is based on a Montgomery multiplication (MM) [1], and uses a high-to-low binary method of exponentiation. Several algorithms suitable for hardware implementation of MM are known. In this paper, as a starting point we use the algorithm described and analysed in [5].

With hand–crafted optimisation we managed to embed a modular exponentiation of 132-bit long integers into one Xilinx XC6000 chip, which is to our knowledge one of the best fine-grained FPGA designs for a modular exponentiation reported so far. 3,000 out of 4,096 gates are used for computations and registers, providing 75% density. Our hardware implementation relies on configurability of FPGAs, but does not use run-time reprogrammability or/and SRAM memory. This makes our design simpler and easy to implement. The price to pay is that more FPGA chips are needed to implement RSA with longer keys. 4 Kgates (one XC6000 chip) is required for modular exponentiation of 132-bit long integers. 512-bit keys need four XC6000 chips connected in a pipeline fashion, or 16 Kgates. Taking into account the total running time, we can estimate the bit rate for a clock frequency of 25 MHz being approximately 800 Kb/sec for 512 bit keys, which is comparable with the rate reported in a fundamental paper of Shand and Vuillemin [3], and an order of magnitude better than that one in [2]. Full description of the FPGA design can be found in [4].

References

[1] P. L. Montgomery, *Modular multiplication without trial division*, Mathematics of Computations, 1985 (44) 519–521.

[2] H. Orup, E. Svendsen, E. And, *VICTOR an efficient RSA hardware implementation*, In: Eurocrypt 90, LNCS, vol. 473 (1991) 245–252

[3] M. Shand, J. Vuillemin, *Fast Implementation of of RSA Cryptography*, In: Proc. of the 11th IEEE Symposium on Computer Arithmetics, 1993. pp.252–259.

[4] A. A. Tiountchik, E. Trichina, *RSA Acceleration with Field Programmable Gate Arrays*, In: Information Security and Privacy, LNCS, vol. 1587 (1999) 164-176.

[5] C. D. Walter, *Systolic Modular Multiplication*, IEEE Trans. on Comput., 1993 (42) 376–378.