

Optimum Implementation of Elliptic Curve Cryptosystems on the SRC-6E Reconfigurable Computer

Nghi Nguyen¹, Kris Gaj¹, David Caliga², Tarek El-Ghazawi³

¹ George Mason University, ² SRC Computers, ³ The George Washington University

1. Introduction and Related Work

Reconfigurable Computers are general-purpose high-end computers based on a hybrid architecture and close system-level integration of traditional microprocessors and Field Programmable Gate Arrays (FPGAs). It is expected that programming of reconfigurable computers should not require any knowledge of hardware design, assuming that sufficiently large library of elementary operations has been earlier developed and made available to end users.

The emergence of reconfigurable computers offers a great promise in terms of progress in many traditionally hard cryptographic problems [1]. Many of such problems, such as integer factorization, elliptic curve discrete logarithm problem, or counting the number of points on an elliptic curve have been shown in theory to execute substantially more efficiently in hardware [2, 3]. At the same time, no prototypes confirming these claims have been reported in the open literature for practical sizes of cryptographic parameters because of the prohibitive cost of specialized hardware.

Although a lot of work has been done in the area of reconfigurable computing and run-time reconfiguration, we are aware of only a few practical implementations of general-purpose reconfigurable computers [4, 5]. SRC-6E from SRC Computers, Inc. was chosen for our study [4]. Our goal was not only to confirm the great potential for effective use of reconfigurable computers in cryptography, but also to determine the current and possible future limitations of the reconfigurable computing technology. We chose as our benchmark a relatively complex cryptographic operation: Elliptic Curve scalar multiplication over $GF(2^n)$ with a polynomial basis representation [6, 7, 8]. This operation forms a basis of Elliptic Curve Cryptosystems and is perfect for our goals, as it involves a three-level hierarchy of operations. Our goal was to find out, which level functions need to be implemented by a hardware designer as library macros, and at what level the software designer can take over. Our paper gives an answer to this question for the current generation of reconfigurable computers, and provides some predictions on how this answer can evolve in time.

2. Methodology

The implementation of Elliptic Curve Cryptography on a reconfigurable computer requires the implementation of three levels of functions:

The low level functions are multiplication ($a \cdot b$), inversion (a^{-1}), and addition ($a+b$) in $GF(2^n)$. The medium level functions include elliptic curve point addition ($P+Q$) and doubling ($2P$). The highest level function, scalar multiplication, kP , is composed of a sequence of point additions and doublings depending on the bits of the binary representation of a large integer k .

At each level, a function can be implemented using three major approaches: as a VHDL or Verilog hardware macro, as a hardware function described in C, or as a purely software C function. These approaches offer a clear trade-off between the efficiency of the design, and the time necessary to develop an acceptable solution.

In our paper, we investigate several possible combinations of these implementation approaches, and compare them in terms of the quality of the solution (latency, area), as well as amount of effort necessary to complete the solution.

Our implementation has been tested using test vectors obtained from LiDiA, the public domain library for computational number theory [9]. The execution time of operations within the SRC User FPGA was measured in the number of clock cycles using the standard SRC macros. The execution time of the C functions was measured in time units using the C timer functions of the Linux operating system.

Our implementations of elliptic curve cryptosystems were compared among each other, as well as against purely software implementation based on LiDiA, running on the microprocessor board of SRC-6E.

3. Results and Conclusions

Implementation of Elliptic Curve Cryptosystems over $GF(2^n)$ is particularly challenging for reconfigurable computers because the primary optimization criterion is latency rather than throughput, and there is only limited amount of parallelism involved in the medium level operations, such as point addition.

Our study revealed the optimum boundary between hardware and software, and between the description of hardware in VHDL vs. C for the three-level hierarchy of operations constituting the Elliptic Curve scalar multiplication. This boundary had to take into account the trade-off between the benchmark performance and the designer productivity, and the limitations of the current generation of reconfigurable computers. This boundary is likely to change in the future in favor of describing more hardware operations in high level languages, such as C.

We have shown how the hardware architecture and programming model of SRC-6E has influenced the choice of the architectures for all component operations and the overall design of our benchmark application. We have found out that at the current stage of the development of reconfigurable computers, an intimate knowledge of the system hardware architecture and programming model, and the associated overheads, might be necessary to fully utilize the potential offered by this promising technology.

We clearly demonstrated that reconfigurable computers offer a great promise for solving complex cryptographic problems with the speed of specialized hardware and flexibility and productivity of software implementations.

References

1. Buell, D. A. and Davis J. P., Reconfigurable Computing Applied to Problems in Communications Security. 2002 MAPLD International Conference Proceedings
2. Bernstein, D. J., "Circuits for integer factorization: a proposal," available at <http://cr.yp.to/papers.html#nfcircuit>
3. Shamir, A., Tromer, E.: Factoring Large Numbers with the TWIRL Device, (preliminary draft) available at <http://www.wisdom.weizmann.ac.il/~tromer/>
4. SRC Inc. Web Page, <http://www.srccomp.com/>
5. Star Bridge Systems Web Page, <http://www.starbridgesystems.com/>
6. Enge, A.: Elliptic Curves and Applications to Cryptography, Kluwer Academic Publishers, 1999
7. Wolkerstorfer, J.: Dual-Field Arithmetic Unit for $GF(p)$ and $GF(2^m)$. CHES 2002, 500-513
8. Gura, N. et al.: An End-to-End Systems Approach to Elliptic Curve Cryptography. CHES 2002, LNCS 2523, (2002) 349-365
9. LiDiA. A library for computational number theory, Technical University of Darmstadt. Available from <http://www.informatik.tu-darmstadt.de/TI/LiDiA/Welcome.html>

Corresponding author:

Kris Gaj
George Mason University
Phone: (703) 993-1575
E-mail: kgaj@gmu.edu