Packet Level Authentication: Hardware Subtask Final Report

Juha Forsten, Kimmo Järvinen and Jorma Skyttä
Department of Signal Processing and Acoustics
P.O. Box 3000, 02015 TKK, Finland
{jforsten, kimmo, jsk}@wooster.hut.fi

1/2006–4/2008

1 Introduction

The computational requirements of the Packet-Level Authentication (PLA) are very high. Hence, software implementations fail to meet performance requirements of modern communication networks and hardware acceleration is an absolute necessity. Field Programmable Gate Arrays (FPGAs) have proven to be attractive alternatives for implementing cryptographic algorithms because of the combination of speed and flexibility [1].

Because high security level with short keys and fast performance is essential in the PLA, elliptic curve cryptography was selected. Koblitz curves were selected because they offer faster computation. See the report of the Crypto subtask for more details. All computations were specifically optimized for NIST K-163 elliptic curve standardized by the National Institute of Standards and Technology (United States).

Ideally all operations related to the cryptographic algorithms in the PLA would be delegated to an external accelerator. However, because of the limited time and resources, only elliptic curve operations were accelerated in these proof-of-concept implementations, because they dominate in the overall performance. Elliptic curve point multiplication is the fundamental operation required in all elliptic curve cryptosystems. It is defined by $Q = kP$, where Q and P are points on an elliptic curve represented with two coordinates, e.g. $P = (x, y)$, and k is an integer. The security of elliptic curve cryptography is based on the assumption that it is computationally infeasible to solve k from Q and P if the curve is selected carefully. The research of this subtask focuses on efficient implementation of the point multiplication.

The work related to the subtask was conducted in the Department of Signal Processing and Acoustics (prior to Jan. 2008 known as the Signal Processing Laboratory) of the Faculty of Electronics, Communications and Automation at the Helsinki University of Technology by

1. DI Juha Forsten (full-time/part-time) and
2. DI Kimmo Järvinen (part-time).

The supervisor of the work was Prof. Jorma Skyttä. The tasks were divided so that Forsten designed and implemented communication between the host computer and the FPGA and Järvinen designed and implemented elliptic curve cryptography modules.

2 Research Work

2.1 Hardware Platform

An FPGA device was chosen as a target technology for the PLA hardware prototype as it provides fast and easily reconfigurable platform when changes and redesign are apparent. The hardware platform used in PLA as a testing environment is based on Development Kit made by Altera, which is also the manufacturer of the FPGA device. The DSP DevKit contains one Altera's Stratix II EP2S180 family device and several peripherals like ADCs, DACs, external memory, 10/100 Mbit/s Ethernet connection and general I/O interface. It comes with full development software support which fastens the startup of prototype designs. The FPGA device EP2S180, based on a 1.2-V, 90-nm, SRAM process, contains 71,760 ALMs (Adaptive Logic Modules), equivalent of 179,400 Logic Elements (LEs). The chip contains 96 dedicated digital signal processing (DSP) blocks optimized for high performance DSP applications. For data storage and buffering there are built-in memory blocks total of 1,172,880 bytes.

Altera has an embedded software processor family called Nios II, which provides scalable processor cores for System on Chip (SoC) designs inside FPGA device. The main advantage of the use of embedded software processor core in prototyping environment is the ability to write more processor oriented tasks like controlling logic and some communication tasks in C language instead of VHDL. As the Nios II provides ready-made programmable peripherals like Ethernet interfaces and UARTs, it allows focusing design efforts to the actual VHDL module.

2.2 Communication

The Nios II soft-core processor and the built-in Ethernet interface is used for the communication between host and the crypto hardware accelerator. The data for elliptic curve point multiplication are received and sent using TCP/IP packets. The actual point multiplication is calculated in dedicated elliptic curve cryptography module (see Sec. 2.3) that is connected to the Nios II via bidirectional internal 32-bit interface. The communication software in Nios II uses MicroC/OS-II microkernel and a TCP/IP stack which is implemented using lwIP (lightweight IP) library.

The procedure to calculate the point multiplication in the hardware is following. A host computer sends TCP packet containing needed parameters for calculation (x, y, and k) to the hardware. The program running in the hardware (in Nios II processor) receives TCP packet and send the parameters to the crypto module using internal interface. When calculation is finished the software reads result values to TCP packet and sends it back to the host computer.

Currently calculation of one point multiplication requires about 900 μs, including TCP transmission. This is dominated by relative slow TCP/IP stack implementation in the NIOS II processor as the actual calculation is faster even with only one crypto module (see Sec. 2.3). However, this is a proof-of-concept that is currently faster than software only implementations available.

There a few steps which can be taken to improve performance. One is to use only Ethernet low-level MAC packets that leaves out the slow TCP/IP stack processing. It is then possible to speed up the point multiplicatication by using multiple crypto modules and thus utilizing the parallelism inside the FPGA device. After that the limiting factor is the 100 Mbit/s Ethernet interface that could be further replaced with a 1 Gbit/s interface.

2.3 Elliptic Curve Cryptography Module

The core component of the accelerator is the elliptic curve cryptography module. The main scientific results of the subtask deal with the implementation of this module. The module implements elliptic curve point multiplication on a standardized elliptic curve NIST K-163 which belongs to a family of elliptic curves called Koblitz curves. FPGA-based implementations have been extensively studied during the past few years and a comprehensive review is presented in [2]. Prior to the PLA project, the fastest FPGA implementation for NIST K-163 curve was presented in [3] where one point multiplication required 75 μs on a Xilinx Virtex-E FPGA. Because the PLA sets very high speed requirements, special interest was given for parallel implementations which were not extensively studied in the literature before this project.

The very high speed requirements of the PLA can be met only if several point multiplications can be computed in parallel. Hence, the architecture was designed to be modular so that it can be efficiently parallelized.

The main component is a field arithmetic processor (FAP) supporting addition, squaring and multiplication in the 163-bit finite field. Each FAP computes one point multiplication simultaneously. Time required for a single computation depends on various parameters, of which the most important ones are the number of multipliers in an FAP and the speed of a single multiplier. Because point multiplication decomposes into three hierarchical levels and different parallelization setups can be used on all levels, finding a setup optimizing speed-area ratio of an FAP is a challenging task which has not been sufficiently addressed in previous research. Parallelization in the case of the PLA signature verification (the most complex operation in the PLA) was studied in [CHES07] and it was concluded that up to 166,000 signatures can be verified with a single Stratix II FPGA. A more general parallelization study was presented in [VLSI] where optimal setups for an FAP were searched and a new parallel architecture for low latency point multiplication was suggested. The best results indicated that point multiplication requires only 20.3 μs.

Even further improvements were achieved after a careful study of data dependencies in Koblitz curve point multiplication. It was concluded that the point multiplication time can be reduced to less than 5 μs which outperforms all previously reported results. This work was submitted to Microprocessor and Microsystems journal [MM]. A design based on the work of [MM] was recently presented in [FCCM] and it achieves both high throughput and fast computation of a single operation. The design achieves 46% more verifications

per second than [CHES07] and, hence, it represents the state-of-the-art of the PLA acceleration.

2.3.1 τ-adic Converters

Koblitz curves are a special family of elliptic curves providing notably faster computation than general elliptic curves with a similar level of security. However, they require that the scalar in point multiplication, i.e. k in kP, is converted into a so-called τ-adic expansion. The conversion is not trivial and in order to reach the full potential of the hardware accelerator also the conversion needs to be implemented in hardware. Thus, a hardware architecture of the conversion was designed and published in [ICECS]. Such converters were not presented in the literature prior to our publication.

In certain cases, e.g. in signature generation, random τ-adic expansions can be used and binary to τ-adic converters are not needed. However, the scalar may still be needed in binary format which yields a need for a τ-adic to binary converter. This problem was studied in [SAC] where a converter architecture was also presented.

2.3.2 Double-Base Methods (collaboration with Univ. Calgary, Canada)

An altenative method for implementing elliptic curve point multiplication on Koblitz curves was also studied during the project in collaboration with a research group from the University of Calgary, Canada. The background for this study is in Järvinen's research visit to Calgary during fall 2005, but the work has continued throughout the PLA project.

The work is based on Prof. Dimitrov's (Univ. Calgary) idea where the scalar k is represented by using a double-base τ-adic expansion which is considerably sparser than traditional τ-adic expansions. The sparser the expansion is the lower is the complexity of point multiplication. Hence, considerably faster implementations can be designed. Järvinen has been responsible of an FPGA-based implementation of the new method which was published in [CHES06]. The work was continued by studying the effects of parallel computation (entirely by Järvinen) and an extended version of the article [COMP] was recently accepted for publication in IEEE Transactions on Computers.

2.4 Results

Table 1 presents a comparison of the results of this project to two published results from the literature. The reader is referred to [2] for a more comprehensive survey of the published implementations. This comparison only includes the fastest implementations from [2] which are used for comparing the results of this subtask. The fastest implementation for general curves was presented in [4] and the fastest implementation for Koblitz curves in [3]. Notice that although [4] is faster than [3], Koblitz curves still offer considerable performance increases compared to general curves. The reason why [4] is faster is because it uses a newer FPGA and a better optimized computation architecture. Table 1 clearly shows that the results obtained in this project compete well with other existing works and are actually the fastest implementations presented up to date. Also notice that software implementations usually require some

Table 1: Comparison of point multiplication implementations

Ref.	Device	Time (μs)	OPS	Notes
[3]	Virtex-E	75	13,300	
[4]	Virtex-II	41	24,400	General curve
[CHES06]	Virtex-II	35.8	28,000	Double-base τ-adic
[CHES07]	Stratix II	114.2	166,000	Signature verification
[VLSI]	Stratix II	20.3	49,300	
[MM]	Stratix II	4.9	203,700	
[FCCM]	Stratix II	16.4	161,300	
		35.1	60,600	Signature verification

Table 2: Power consumption of two designs: [CHES07] and [VLSI].

Design	Device	Speed	Power (W)
[CHES07]	Stratix II 2S180C3	166,000 ops	18.6
	Stratix III 3SL340C3	177,000 ops	12.1
	Hardcopy II HC240 (ASIC)	212,000 ops	5.6
[VLSI]	Cyclone II 2C70C7	130 μs	0.86
	Cyclone III 3C10C8	120 μs	0.29
	Stratix II 2S180C3	80 μs	2.26
[CHES07,max]	Stratix III 3SL340C3	306,000 ops	20.9
	Hardcopy II HC240 (ASIC)	850,000 ops	22.4

milliseconds to compute a single point multiplication and, thus, hardware implementations are several hundred times faster.

2.5 Power Consumption

Power consumption of two designs [CHES07] and [VLSI] was studied. Where as the [CHES07] design is targeted to high-performance systems, the [VLSI] design is optimized for low-power and low-cost applications, which means it is smaller and slower than the [VLSI] design used in Table 1.

The values are received from Altera PowerPlay tool by analyzing designs with full signal activity information from timing simulations of 200 μs or 100 μs for [CHES07] and [VLSI], respectively. [CHES07,max] designs estimate situations where the largest possible Stratix III or HardCopy II devices are filled with [CHES07] designs. Such designs have not been implemented and the values are thus highly approximative.

As seen from Table 2, the third generation of Altera FPGAs uses 65-nm technology with power-saving features and thus outperforms earlier generation devices both in speed and power consumption areas. The HardCopy designs are non-reprogrammable ASICs-like devices and therefore have further improvements in speed and power consumption compared to reprogrammable devices.

The studies show that the cryptography module is very scalable both in the performance and the power consumption areas and for that reason usable in wide range of applications.

3 Summary and Conclusions

The results of the subtask indicate that the PLA could be feasible in practice, because reasonable performance levels (166,000 packets per second) can be achieved with moderate cost (a single high-end FPGA). The new method of [MM] was estimated to increase performance values by a further 46% resulting over 240,000 PLA verification per second in [FCCM]. Hence, the PLA and its hardware acceleration deserve more research.

The quality of the results is high which can be seen from the fact that the publications were accepted to prominent venues. The results include articles in IEEE Transactions on VLSI Systems, IEEE Transaction on Computers, and several high quality conferences. The CHES conference [CHES06,CHES07] is commonly considered as the leading conference in the field of hardware cryptography, and the acceptance percentage for submitted papers is only about 30%. The SAC conference [SAC] also accepted only 34% of submissions. Besides, an article is still waiting for decisions from a journal [MM]. The results of the subtask form the majority of Järvinen's doctoral thesis [DSc] which is expected to be published during the second half of 2008.

Publications

[CHES07] Kimmo Järvinen, Juha Forsten and Jorma Skyttä, FPGA Design of Self-Certified Signature Verification on Koblitz Curves, in *Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems, CHES 2007*, Vienna, Austria, Sep. 10-13, 2007, pages 256-271, Springer-Verlag LNCS 4727.

[VLSI] Kimmo Järvinen and Jorma Skyttä, On Parallelization of High-Speed Processors for Elliptic Curve Cryptography, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, in press.

[MM] Kimmo Järvinen and Jorma Skyttä, Fast Point Multiplication on Koblitz Curves: Parallelization Method and Implementations, *Microprocessors and Microsystems*, submitted on Oct. 26, 2007.

[FCCM] Kimmo Järvinen and Jorma Skyttä, High-Speed Elliptic Curve Cryptography Accelerator for Koblitz Curves, in *Preliminary Proceedings of the 16th IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM 2008*, Stanford, California, USA, Apr. 14-15, 2008, final proceedings will be published by IEEE Computer Society in summer 2008.

[ICECS] Kimmo Järvinen, Juha Forsten and Jorma Skyttä, Efficient Circuitry for Computing τ-adic Non-Adjacent Form, in *Proceedings of the 13th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2006*, Nice, France, Dec. 10-13, 2006, pages 232-235.

[SAC] Billy Bob Brumley and Kimmo Järvinen, Koblitz Curves and Integer Equivalents of Frobenius Expansions, in *Revised Selected Papers of the 14th Annual Workshop on Selected Areas in Cryptography, SAC 2007*, Ottawa, Canada, Aug. 16-17, 2007, pages 126-137. Springer-Verlag LNCS 4876.

[CHES06] V.S. Dimitrov, K.U. Järvinen, M.J. Jacobson, Jr., W.F. Chan, and Z. Huang, FPGA Implementation of Point Multiplication on Koblitz Curves Using Kleinian Integers, in *Proceedings of Workshop on Cryptographic Hardware and Embedded Systems, CHES 2006*, Yokohama, Japan, Oct. 10-13, 2006, pages 445-459. Springer-Verlag LNCS 4249.

[COMP] Vassil S. Dimitrov, Kimmo U. Järvinen, Michael J. Jacobson, Jr., Wai Fong Chan, and Zhun Huang, Provably Sublinear Point Multiplication on Koblitz Curves and its Hardware Implementation, *IEEE Transactions on Computers*, in press.

[DSc] Kimmo Järvinen, Hardware Accelerator Architectures for Cryptographic Algorithms (tentative title), Doctoral thesis, estimated publication in fall 2008.

References

[1] T. Wollinger, J. Guajardo, and C. Paar. Security on FPGAs: State-of-the-art implementations and attacks. *ACM Transactions on Embedded Computing Systems*, 3(3):534–574, August 2004.

[2] G. Meurice de Dormale and J.-J. Quisquater. High-speed hardware implementations of elliptic curve cryptography: A survey. *Journal of Systems Architecture*, 53(2-3):72–84, February–March 2007.

[3] J. Lutz and A. Hasan. High performance FPGA based elliptic curve cryptographic co-processor. In *Proceedings of the International Conference on Information Technology: Coding and Computing, ITCC 2004*, page 486–492, Las Vegas, Nevada, USA, April 5-–7, 2004.

[4] B. Ansari and M. Anwar Hasan. High performance architecture of elliptic curve scalar multiplication. Technical Report CACR 2006-01, University of Waterloo, 2006.