Elliptic Curve Cryptography support for ARM based Embedded systems

Sandro Bartolini, Paolo Bennati, Roberto Giorgi, Enrico Martinelli

Dept. Ingegneria dell'Informazione, University of Siena, Via Roma 56, 53100 Siena, Italy

ABSTRACT

Elliptic Curve Cryptography (ECC) is emerging as an attractive approach to public-key cryptography for constrained environments, because of the small key sizes and computational efficiency, while preserving the same security level as the standard methods. The performance of public-key cryptography methods is critical in embedded environments such as applications for wireless, handheld internet devices and smart cards with small memory and strict CPU-latency constraints. Power control is also important for embedded systems as well as security against Differential Power Analysis (DPA).

We examined the performance of a set of ECC kernel benchmarks and proposed ISA extensions to support secure and efficient execution, on a ARM processor, which is a very common platform for embedded system applications. An evaluation of possible ARM instruction set extension for Elliptic Curve Cryptography over binary finite fields GF(2^m) is presented. With almost no cost at hardware level, we found an average 33% reduction of the total number of dynamically executed instructions.

Finally, we analyzed the power requirement to achieve an efficient and secure execution from a power standpoint. Some preliminary results showing the power consumption of the benchmarks are presented.

KEYWORDS: ACACES; poster session; Elliptic Curves Cryptography; ECC; Differential Power Analisys; DPA; Diffie-Hellman; El-Gamal.

1 Introduction

Cryptography algorithms can be divided into two categories: *private-key* (symmetric) and *public-key* (asymmetric). Private-key systems use a common private key shared between the communicating parties, while public-key do not require any key exchange.

Mainly public-key algorithms are currently used into encryption schemes (like RSA or El-Gamal [1]), digital signature schemes (like Digital Signature Algorithm, DSA [2]), and key agreement methods (like Diffie-Hellman [3]).

Both private and public key sizes must grow over time to offer acceptable security [2, 4].

The major advantage of ECC is the use of shorter keys with a security level equivalent to finite fields public-key algorithms; as a consequence faster implementations, reduced energy and bandwidth consumption can be achieved. Because of these characteristics, ECC is already incorporated into two important public-key cryptography standards, FIPS 186-2 [2]and IEEE-P1363 [5].

We analyzed the major bottlenecks at function level in ARM processors (90% of mobile phones use it [6]) and evaluated the performance improvement [7, 8], when we introduce some simple architectural support in the ARM ISA. We also address the problem of the power requirements of the benchmark; DPA techniques [9-11] are a serious threat for the security of embedded devices and power analysis could be useful to achieve a more secure execution.

2 Methodology

Microprocessor Configuration				
Fetch queue (instructions)	4		Instruction L1 cache	32 KB, 32-way
Branch prediction	8K bimodal, 2K 4-way BTB		Data L1 cache	32 KB, 32-way
Fetch & Decode width	1		L1 cache hit latency (cycles)	1
Issue width	1 (in order)		L1 cache block size	32 bytes
ITLB	32 entry, fully associative		L2 cache	None
DTLB	32 entry, fully associative		Memory latency (cycles)	24
Functional units	1 ALU, 1 int MUL/DIV		Memory bus width (bytes)	4

Energy Configuration	
Technology	TECH_070
L1 data cache ports	2
L1 data cache temperature	353 K
L1 data cache supply voltage	0.9 V
L1 data cache threshold voltage N Type Cell	0.1902 V
L1 data cache threshold voltage P-Type Cell	0.2130 V
L1 data cache threshold voltage N Type Bitline	0.1902 V

Table 1: Simulated architecture

We used a modified version of sim-profile and sim-outorder simulators of the SimpleScalar toolset [12] for the ARM target [13] to perform performance simulation and Hotleakage [14] to perform power simulation. The parameters used represent an architecture configuration modelled after Intel XScale architecture [15] (Table 1).

The sim-outorder tool was modified to profile to application at a functioning level and we also added all the unimplemented system calls for ARM target, although they were not critical for the execution of the benchmark. The compilation is performed with a version gcc compiler that was modified to recognize newly added instructions.

The description of our benchmark suite, implemented using MIRACL C library [16] is given in Table 2. The binary finite fields and elliptic curves used in tests were chosen according to NIST standard [2].

Benchmark acronym	Benchmark name	Description
ecdh	EC Diffie- Hellman key exchange	Generates a prime suitable for Diffie- Hellman algorithm and calculates a shared key.
ecdsign	EC digital segnature generation	Calculates the message digest of a file using sha algorithm, signs the message using the private key and writes the signature into a file.
ecdsver	EC digital segnature verification	Calculates the message digest of a file using sha algorithm, then verifies the signature using public-key.
ecelgenc	EC El-Gamal encryption	Encrypts a point on the curve using El-Gamal algorithm.
ecelgdec	EC El-Gamal decryption	Decrypts a point on a curve using El-Gamal algorithm.

Table 2: ECC benchmark set

3 The proposed ISA extension

A very large percentage (approximately 40%) of integer instructions as well as load and store operations (approximately 50%) are present in the instruction mix; the finite field operations translate into a large number of logical operations (XOR, shift etc.), and result in a large number of register-memory transfers to operate on m-bit data (e.g. m ranges from 163 to 283 and is much larger than 32 bit register width in ARM).

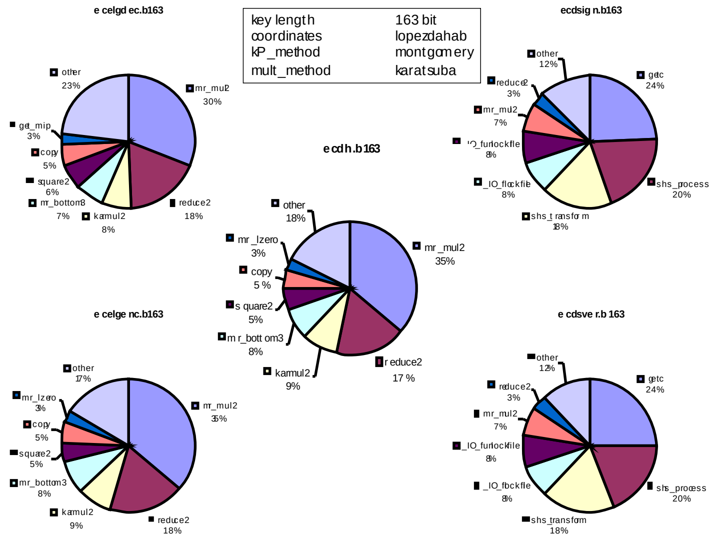

Figure 1: Breakdown of execution time in terms of program functions.

Results (Figure 1) show that in particular, the *mr_mul2* procedure, which multiplies two 32-bit binary finite field polynomials and produces a 64-bit product consumes 34% of the total execution time in average for all benchmarks; *mr_mul2* procedure is translated into about 400 dynamic instructions (roughly 500 cycles), which correspond to about 12 instructions per bit to perform 32-bit polynomial multiplication. Based on the previous analysis, we decided to measure the impact of extending ARM instruction set with an instruction, called MULGF [17], for polynomial word-level multiplication in binary finite fields. The appropriate calls of C procedure for 32-bit polynomial multiplication in software were substituted with a single MULGF instruction. The MULGF instruction was modelled to have a delay of one cycles, as the integer multiplier unit of ARM processor.

The impact of adding the MULGF instruction for word-level polynomial multiplication in finite field is shown in Figure 2 Number of dynamically executed instructions, as well as the execution time, is lower by approximately one-third in average, with projective coordinates. The improvement in execution time is more significant for Diffie-Hellman (54% in number of instructions and 55% for execution time in *GF*(2^{233})) and El-Gamal algorithms (48% for encryption and 37% for decryption in number of instructions, i.e. 39% and 35% in execution time in *GF*(2^{233})), where 32-bit polynomial multiplication is more used. The improvement for digital signature algorithm is more modest (19% in instruction number and 17% in execution time for the same key length), but still significant.

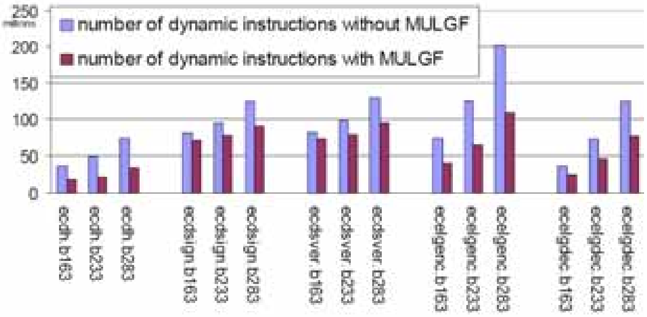

Figure 2: Impact of MULGF instruction. Number of dynamic instructions for projective coordinates before and after adding the MULGF instruction for word-level polynomial multiplication.

4 Power consumption characterization

The use of ECC public-key system is particularly useful for embedded systems in which computation efficiency is required. In addition, embedded systems require also low-power consumption, as the market requests long battery life and satisfying performance.

We will address the problem of power consumption characterization of our ECC benchmarks workload because this aspect is particularly interesting for the physical implementation of embedded secure devices (Figure 3). Our recent (and future) work focuses on the characterization of the benchmark workload based on ECC from a power consumption and DPA point of view; in particular currently we are investigating how the power consumption of the benchmarks can be reduced.

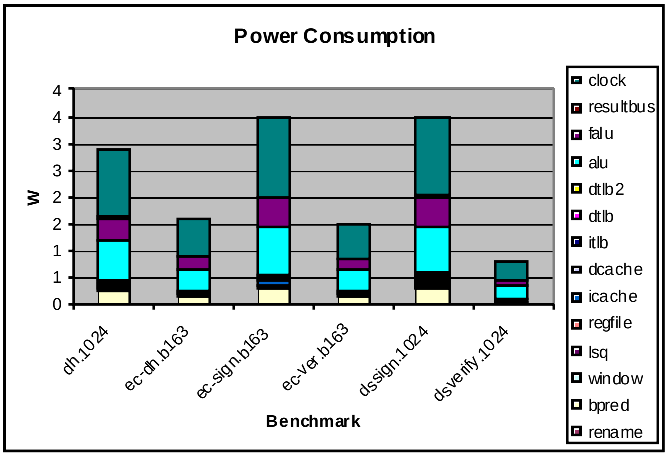

Figure 3: Power ratio in the various part of the system.

Reducing the power consumption is fundamental in embedded system not only to increase battery life but also because it allows the use of techniques to prevent power attack. ECC cryptography is itself lighter than standard methods and we expect a better performance also from this point of view. A power characterization of the benchmarks can be done through HotLeakage [14]. Preliminary results are presented in Figure 3. After a characterization of the power requirements of ECC benchmarks, adaptive techniques will be investigated to improve the benchmark low-power behaviour.

5 Acknowledgment

The power characterization of the benchmarks has been performed with a modified version of HotLeakage retargeted for ARM ISA realized at the Cornell University, Ithaca (USA). We are particularly grateful to Sally McKee for allowing us to use this version.

References

[1] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", in *Proceedings of CRYPTO 84 on Advances in cryptology*. Santa Barbara, California, United States: Springer-Verlag New York, Inc., 1985.

[2] N. I. o. S.a. Technology, "Digital Signature Standard (DSS)", vol. pub 186-2, U. S. D. o. Commerce, Ed.: Federal Information Processing Standards Publication (FIPS), 2000.

[3] W. Diffie and M. E. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644-654, 1976.

[4] V. Gupta, S. Blake-Wilson, and B. M.C. Hawk, "ECC Cipher Suites for TLS", in *IETF internet draft*, 2006.

[5] IEEE P1363 Standard Specifications For Public-Key Cryptography, http://grouper.ieee.org/groups/1363/

[6] ARM Web Site, http://www.arm.com

[7] I. Branovic, R. Giorgi, and E. Martinelli, "Memory Performance of Public-Key cryptography Methods in Mobile Environments", presented at Workshop on MEmory performance: DEaling with Applications, systems and architecture (MEDEA-03), New Orleans, LA, USA, 2003.

[8] S. Bartolini, I. Branovic, R. Giorgi, and E. Martinelli, "A Performance Evaluation of ARM ISA Extension for Elliptic Curve Cryptography over Binary Finite Fields", presented at 16th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD-04), Foz do Iguacu, Brasil, 2004.

[9] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", *Lecture Notes in Computer Science*, vol. 1666, pp. 388-397, 1999.

[10] D. Mesquita, J.-D. Techer, L. Torres, G. Sassatelli, G. Cambon, M. Robert, and F. Moraes, "Current mask generation: a transistor level security against DPA attacks", in *Proceedings of the 18th annual symposium on Integrated circuits and system design*. Florianolpolis, Brazil: ACM Press, 2005.

[11] A. Shamir, "Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies." presented at CHES - Cryptographic Hardware and Embedded Systems, Springer, 2000.

[12] D. Burger and T. M. Austin, "The SimpleScalar tool set, version 2.0", *SIGARCH Comput. Archit. News*, vol. 25, pp. 13-25, 1997.

[13] SimpleScalar Version 4.0 Test Releases, http://www.simplescalar.com/v4test.html

[14] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects," University of Virginia, Charlottesville 2003.

[15] Intel, "Intel XScale Microarchitecture", in *Technical Summary*, 2000.

[16] MIRACL - Multiprecision Integer and Rational Arithmetic C/C++ Library, http://indigo.ie/~mscott/

[17] T. Acar, "High-Speed Algorithms & Architectures For Number-Theoretic Cryptosystems." in *Department of Electrical & Computer Engineering*, vol. Ph.D. Thesis: Oregon State University, 1997.