Michał MOSDORF [1], Wojciech ZABOŁOTNY [2]
[1] WARSAW UNIVERSITY OF TECHNOLOGY, INSTITUTE OF COMPUTER SCIENCE, ul. Nowowiejska 15/19, 00-665 Warsaw
[2] WARSAW UNIVERSITY OF TECHNOLOGY, INSTITUTE OF ELECTRONIC SYSTEMS, ul. Nowowiejska 15/19, 00-665 Warsaw

Implementation of elliptic curve cryptography for 8-bit and 32-bit embedded systems – time efficiency and power consumption analysis

M.Sc,. eng. Michał MOSDORF

PHD student at Institute of Computer Science of Faculty of Electronics and Information Technology. Graduate of Computer Science at Faculty of Electronics and Information Technology (2009). Conducts research in field of software reliability in embedded systems environment.

e-mail: m.mosdorf@ii.pw.edu.pl

Ph.D., eng. Wojciech ZABOŁOTNY

MSc thesis in electronics in 1989 on Warsaw University of Technology. PhD thesis in 1999 "Methods of Estimation of Maximum Frequency in the Transcranial Doppler Signal" also in Warsaw University of Technology. Currently assistant professor in Institute of Electronic Systems in Warsaw University of Technology Since 2002 member of Polish CMS Group cooperating with LHC experiment in CERN.

e-mail: m.mosdorf@ii.pw.edu.pl

Abstract

Results of research that compares possibilities of securing transmission from biomedical embedded telemetry devices with elliptic curve cryptography algorithms performed on 8-bit and 32-bit microcontrollers is presented in the paper. The review of possible ways of implementing cryptographic protocols based on elliptic curves in embedded systems with usage of commercial MIRACL library and open-source GMP arithmetic library was performed. We have used MIRACL library to compare time efficiency and energy costs of elliptic curve point multiplication on selected AVR and ARM embedded platforms. Additionally we have implemented basic ECC library based on open-source GMP library for ARM microcontrollers to check efficiency of algorithms based on different number representations. Results obtained during the work showed that despite of the fact that selected ARM microcontroller active supply current is higher than selected AVR device active supply current, total energy cost associated with elliptic point curve multiplication is much smaller in case of ARM microcontrollers.

Keywords: ECC, ARM, AVR, GMP, MIRACL, energy cost.

Implementacja algorytmów kryptograficznych opartych na krzywych eliptycznych dla 8-bitowych i 32-bitowych systemów wbudowanych – analiza wydajności i zużycia energii

Streszczenie

Celem pracy była analiza możliwości wykorzystania kryptografii opartej na krzywych eliptycznych w wybranych systemach wbudowanych w celu realizacji bezpiecznej transmisji danych biomedycznych W pracy porównano możliwości implementacji algorytmów kryptograficznych bazujących na krzywych eliptycznych w środowisku 8-bitowych mikrokontrolerów AVR oraz 32-bitowych mikrokontrolerów ARM. Za pomocą komercyjnej biblioteki MIRACL zbadano wydajność obliczeniową oraz koszt energetyczny związany z operacją mnożenia punktu na krzywej eliptycznej w środowisku wybranych systemów wbudowanych. Dodatkowo bazując na bibliotece GMP wykonano implementację podstawowych operacji na krzywych eliptycznych dla wybranego mikrokontrolera ARM. Za pomocą wykonanej implementacji porównano wydajność operacji na krzywych eliptycznych realizowanych dla różnej reprezentacji liczb (reprezentacji binarnej i NAF (ang. Non-Adjacent Form)). Wyniki pokazują, iż rozpatrywana rodzina mikrokontrolerów 32-bitowych charakteryzuje się mniejszym kosztem energetycznym operacji mnożenia punktu na krzywej eliptycznej oraz większą wydajnością obliczeniową niż układy 8-bitowe.

Słowa kluczowe: ECC, ARM, AVR, GMP, MIRACL, koszt energetyczny.

1. Introduction

Cheap and low power radio links using unlicensed ISM (Industrial, Scientific and Medical) radio-frequency bands are used in many areas to eliminate inconvenient cabling. Also in the biomedical applications the radio links allow to build wearable monitoring systems, which combine capability to record biomedical parameters during normal activity of the patient, with possibility to quickly notify the medical center about any significant problems e.g. via the GSM connection. These links may be based on different standards like Bluetooth (IEEE802.15.1), or ZigBee (IEEE 802.15.4), or on self-developed protocols. In all cases however the integrity and confidentiality of the data should be protected by appropriate cryptographic protocols

Software solutions in embedded systems that provide cryptographic protocols functionality can be based on ready solutions such as embedded Linux with SSL support or non operating system environment that requires developers to provide specific implementation of cryptographic protocols. The main advantage of the first solution is the short time of software design and implementation but on the other hand it requires more complex and expensive processors such as (AVR32, ARM9 or ARM11) that are capable of running e.g. Linux operating system. Such solutions also are characterized by higher power consumption and therefore shorter battery operation time, which can be critical in portable and especially in wearable telemetry systems.

The most complex part of a cryptographic protocol is typically the asymmetric key encryption/decryption algorithm. The most widely used and well tested asymmetric key algorithms are RSA [12] and Elliptic Curve Cryptography (ECC) [2]. Elliptic curve cryptography [1, 2] in comparison to RSA allows to use the shorter cryptographic keys at the same level of security. Because of that ECC reduces computations time and energy cost associated with performing cryptographic algorithms. Due to these advantages ECC is being widely used in various constrained embedded environments.

In this paper we discuss and compare possibilities of using elliptic curve cryptography on selected AVR and ARM microcontrollers that are not capable of running Linux operating system with built-in SSL support. We mainly focus on efficiency of elliptic curve point multiplication and energy cost associated with it. Additionally we present possibilities of using open-source GMP library [4] to implement ECC system on 32-bit ARM microcontroller. For the research purpose we have chosen two AVR microcontrollers: ATmega128 powered by 5V and ATmega644 powered by 3.3V. Additionally from various ARM microcontrollers available on the market we have selected widely available AT91SAM7X256 powered by 3.3V. All arithmetic operations on elliptic curve points were done in F_p finite field and with usage of standard NIST [5, 6] and SCEG [7, 8] elliptic curve parameters.

2. Related work

Because of its advantages ECC is widely used in constrained embedded systems. Authors in [9] described ECC implementation for clocked at 7MHz AVR ATmega128 microcontroller used in wireless sensor networks. Implementation was based on work undertaken in [11] with some additional optimizations for chosen platform. Authors have used $F_{2^{113}}$ finite field with optimal normal base representation. Authors have achieved 6.88 s for ECDSA signature generation, 24.17 s for ECDSA signature verification, 24.07 s for ElGamal encryption and 17.87 s for ElGamal decryption.

Another example of ECC usage in embedded system environment can be found in [10]. Authors describe implementation of ECC library over F_p finite field for ARM7TDMI processor clocked at 80 MHz. During work authors have designed and implemented software modules responsible for performing arithmetic's on modulo p level and elliptic curve point level. As a result authors have achieved 46 ms and 92 ms for 160-bit ECDSA signature and verification.

3. Elliptic Curve Cryptography

Elliptic curve defined over field K is described by Weierstrass equation [2]:

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

For fields with characteristics different from 2 and 3 (e.g. finite field F_p) this equation can be simplified to [2]:

$$y^2 = x^3 + ax + b\,. \tag{2}$$

ECC is based on arithmetic operations performed in additive group of elliptic curve points defined over finite field e.g. F_p or F_{2^m}. During work we have used F_p finite field. Rules for point addition and doubling for that field are defined as follows [2]:

Point addition. If P=(x_1,y_1) and Q=(x_2,y_2) lie on the elliptic curve then addition of this points equals: P+Q=(x_3,y_3) where:

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2\,, \tag{3}$$

$$y_3 = \left(\frac{y_1 - y_2}{x_2 - x_1}\right)(x_1 - x_3) - y_1\,. \tag{4}$$

Point doubling. If P=(x_1,y_1) lies on the elliptic curve then point doubling equals: P+Q=(x_3,y_3) where:

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a\,, \tag{5}$$

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1\,, \tag{6}$$

where

$$\lambda = (y_1 + y_2)/(x_1 + x_2)\,.$$

Besides point addition and doubling mathematics of elliptic curves defines additive point multiplication operation kP=Q where P,Q∈E(K), k∈N. This operation can be implemented as sequential additions and doublings of elliptic curve point. Strength of ECC is based on ECDLP (Elliptic Curve Discrete Logarithm Problem) that is defined as follows. For given P,Q∈E(K) find k∈N that kP=Q. At present time there are no known solutions to this problem in sub-exponential time.

4. Elliptic curve point multiplication efficiency

For time evaluation purpose we have chosen AVR ATmega128 with clock frequency 11.059 MHz and ARM AT91SAM7S256 with clock frequency of 48 MHz. Test software was created with the MIRACL library [3] and it included single elliptic curve point multiplication (additive group generator from ECC standards). Time measurements were performed using internal timer modules of selected microcontrollers. Measurements were performed for various key lengths ranging from 160 to 512 bits. Due to memory constraints in case of AVR microcontrollers it was not possible to test multiplication time for key lengths of 384 and 512 bits.

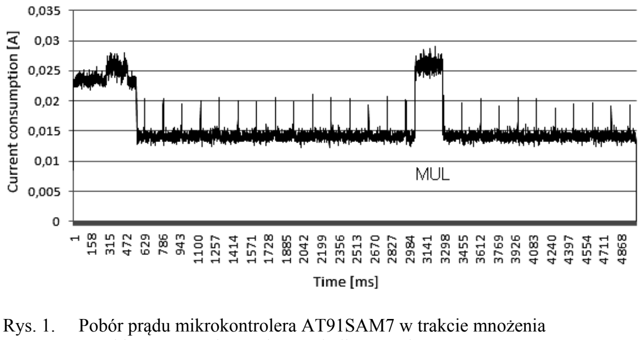
Table 1 shows elliptic curve point multiplication time results measured for selected embedded platforms. Results show superior ARM performance that is caused by difference in processor word length (4 times faster multiplication) and additionally 4.3 times higher clock frequency. It is worth to mention that selected ARM microcontroller is incapable of executing code directly from FLASH memory at full speed. Because of memory latency, ARM core must insert additional wait cycle during memory access at speed of 48 MHz. Therefore selecting different ARM microcontroller could provide better multiplication time results. Additional speed up in case of AVR microcontrollers can be achieved by selecting devices that provide higher clock frequencies e.g. 20 MHz.

Tab. 1. Czas mnożenia punktu na krzywej eliptycznej dla wybranych mikrokontrolerów AVR i ARM
Tab. 1. Elliptic curve point multiplication time for selected AVR and ARM microcontrollers

Key Length	160	192	224	256	384	512
	Elliptic curve point multiplication time, s					
ARM	0,0912	0,1232	0,1973	0,2891	0,6859	1,0839
AVR	0,8359	1,2859	2,1097	3,1526	–	–

Elliptic curve point multiplication time does not directly disqualify any of the selected embedded platforms in the discussed application. Therefore we have additionally performed energy cost (work of electric current) measurements associated with point multiplication. For this purpose we have implemented measurement circuit based on MAX4172 amplifier and additional ARM microcontroller responsible for data acquisition.

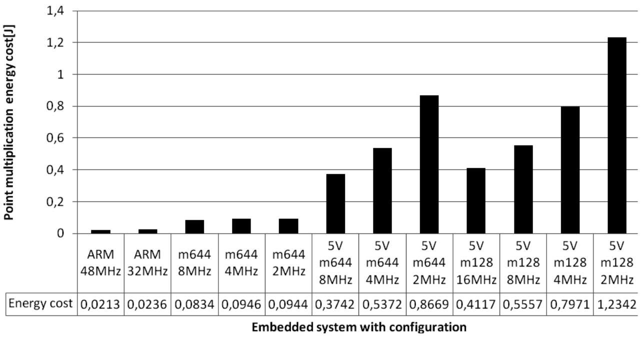
We have measured current consumption of selected embedded systems during 256-bit elliptic curve point multiplication. Figure 1 shows example of acquired current consumption measurement for AT92SAM7 microcontroller. Initial high current level is caused by the ECC system initialization. Low current consumption level represents microcontroller sleep time with periodic current spikes caused by timer module interrupts. Multiplication operation can be seen as high current consumption period market with MUL characters.



Rys. 1. Pobór prądu mikrokontrolera AT91SAM7 w trakcie mnożenia 256-bitowego punktu na krzywej eliptycznej
Fig. 1. Power consumption of AT91SAM7 microcontroller during 256-bit elliptic curve point multiplication

With gathered results from selected embedded platforms with various configurations we have computed energy costs associated with elliptic curve point multiplication. Figure 2 shows results for AT91SAM7X (3.3V), ATmega644 (3.3 V and 5 V) and

ATmega128 (5 V) microcontrollers. Selected devices have the following active supply current: AT91SAM7X (48 MHz, UART and timer enabled) – 26 mA, ATmega644 (3,3 V, 8 MHz) – 4,1 mA, ATmega128 (5 V, 16MHz) – 35 mA.



Rys. 2. Porównanie kosztu energetycznego mnożenia punktu na krzywej eliptycznej dla wybranych systemów wbudowanych w różnych konfiguracjach (ARM – AT91SAM7X245, m644 – Atmega644, m128 – Atmega128)

Fig. 2. Energy cost comparison for elliptic curve point multiplication in different embedded systems with various configuration (ARM – AT91SAM7X245, m644 – Atmega644, m128 – Atmega128)

In general case lowering device clock frequency causes drop in current consumption in microcontroller active mode. However on the other hand computation time increases. In case of devices powered from 5 V power supply it can be seen that lowering of clock frequency causes multiplication energy cost increase. Devices powered by 3.3 V power supply showed similar measured work of electric current for examined clock frequencies. Results also show that selected ARM AT91SAM7S microcontroller can perform elliptic curve multiplication with much lower energy costs what makes it better solution for portable telemetry systems that require battery power supply.

5. ECC implementation based on GMP library

During our work we have also implemented simple ECC library based on open-source GMP library. Main goal of this implementation was to provide portable software that could provide cryptographic protocols primitives working in both embedded and PC environment. We have tested our implementation on selected ARM microcontroller with clock rate at 48 MHz and compared results with MIRACL library tests results.

Table 2 shows elliptic curve point multiplication time in affine coordinates with standard binary number representation and with NAF number representation [2] compared to MIRACL library results. Multiplication was done with simple right-to-left method described in [2]. Multiplication that uses the NAF representation is typically 20-30% faster what is caused by minimalization of non-zero bits count in number representation.

Tab. 2. Porównanie czasu mnożenia punktu na krzywej eliptycznej dla standardowej reprezentacji binarnej oraz reprezentacji NAF z czasem osiągniętym przez bibliotekę MIRACL

Tab. 2. Elliptic curve point multiplication time achieved with standard binary representation and NAF representation compared to MIRACL multiplication results

Key Length	160	192	224	256	384	512
	Elliptic curve point multiplication time, s					
MIRACL	0,0912	0,1232	0,1973	0,2891	0,6859	1,0839
GMP NAF	0,2584	0,3290	0,3990	0,4976	1,2900	2,7779
GMP	0,2957	0,5281	0,5281	0,7016	1,6296	3,4340

6. Conclusions

In the presented paper we have discussed possibilities of implementing elliptic curve cryptographic protocols in embedded systems environment. We have used commercial MIRACL library to test ECC efficiency on selected AVR and ARM platforms. Performed elliptic curve point multiplication energy cost and time efficiency tests showed that selected ARM microcontroller has significantly better performance in discussed field than examined AVR devices. Comparison of elliptic curve point multiplication energy cost showed that 32-bit devices despite higher active supply current (in case of 3.3 V power supply of 8-bit devices) require much less energy to perform ECC cryptographic operations than tested 8-bit microcontrollers.

Implemented during work portable ECC library based on open-source GMP library showed worse but still acceptable performance in comparison to MIRACL library. Research undertaken with this implementation allowed to test efficiency of selected elliptic curve arithmetic algorithms with different binary number representation. Results show that in case of standalone embedded systems based on ARM microcontrollers it is possible to create ECC implementation that utilizes free open-source arithmetic software. Unfortunately GMP does not support 8-bit devices at this moment.

Work results show, that by using elliptic curve cryptography in less complex embedded systems that cannot run operating systems with build-in SLL support, it is possible to reduce overall system cost and energy requirements. Comparison of selected 32-bit and 8-bit microcontrollers point multiplication cost also shows that 32-bit devices are much more suitable for discussed application.

7. References

[1] Koblitz N.: A Course in Number Theory and Cryptography. Springer, Berlin, Germany, 1994.

[2] Hankerson D., Menezes A. J., Vanstone S. A.: Guide to Elliptic Curve Cryptography, Springer, 2004.

[3] Scot M.: Using MIRACL in embedded applications. September 2004.

[4] The GMP team: GNU MP, The GNU Multiple Precision Arithmetic Library. Edition 4.2.4, September 18, 2008.

[5] Brown M., Hankerson D., López J., Menezes A.: Software Implementation of the NIST Elliptic Curves Over Prime Fields. Lecture Notes In Computer Science; Vol. 2020, Springer 2001.

[6] NIST: Recommended Elliptic Curves for Federal Government Use. July 1999.

[7] Cetricom Research: Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography. September 2000.

[8] Cetricom Research: Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters. September 2000.

[9] Blaß E. O., Zitterbart M.: Efficient Implementation of Elliptic Curve Cryptography for Wireless Sensor Networks. TeleMatics Technical Reports 2005.

[10] Aydos M., Yanık T., Koc C.K.: High-Speed Implementation of an ECC-based Wireless Authentication Protocol on an ARM Microprocessor. 16th Annual Computer Security Applications Conference, New Orleans, Louisiana, December 11-15, 2000.

[11] Rosing M.: Implementing Elliptic Curve Cryptography. Manning Publications Co.,1999

[12] Menezes A. J., Oorschot P. C., Vanstone S. A.: Handbook of Applied Cryptography. CRC Press, 1997.

otrzymano / received: 14.05.2010
przyjęto do druku / accepted: 02.07.2010 *artykuł recenzowany*