Reconfiguring Crypto Hardware Accelerators on Wireless Sensor Nodes

Steffen Peter[1], Oliver Stecklina[1], Jorge Portilla[2], Eduardo de la Torre[2], Peter Langendoerfer[1] and Teresa Riesgo[2]
langendoerfer@ihp-microelectronics.com
[1] IHP, Frankfurt/Oder, Germany; [2]Universidad Politécnia de Madrid, ETSI Industriales, Madrid, Spain

Abstract: **Running strong cryptographic algorithms on wireless sensor nodes is extremely difficult due to their limited resources. Hardware accelerators are a suitable means t speed up the computation and reduce power consumption. The drawback of crypto ASICs is the loss of flexibility. In this paper we will shortly introduce a modular design of elliptic curve accelerators which allows to be adjusted to several NIST recommended curves be replacing its reduction unit. This partial reconfiguration will be executed on a Spartan 3 FPGA. The visualization will be done in the following way. Standard motes will be connected to the FPG. On the motes the algorithms will be executed in software. Switching between ECC with a long key i.e. 571bit and those with short key length e.g. to a key length of 163 bit, has a remarkable effect on the execution time. En-/decrypting messages sent to and received from the motes at the FPGA will show that ECC implementation has been reconfigured according to the selected curve on the motes.**

Index Terms: **Elliptic curve cryptography, FPGA, Reconfigurability, wireless sensor nodes**

I. Introduction

WIRELESS SENSOR NODES are becoming a key technology in a wide range of applications areas. Those areas are environmental monitoring, structural health monitoring, industrial automation, telemedicine and homeland security to mention the most prominent ones. The requirements in those application areas are pretty different. For environmental monitoring no security or reliability features are required whereas in homeland security scenarios video surveillance is envisioned requiring reliable support of video streaming and strong cipher means to ensure proper working of the application. In order to ensure long lifetime of battery powered sensor nodes computational intensive tasks should be realized in hardware. While this reduces energy consumption, it also reduces flexibility with respect to the algorithms use to zero. This is an issue if the sensor nodes cannot be replaced easily or when changes are frequent. In order to resolve the conflict between hardware based efficiency and hardware related in-flexibility the SMART project researches reconfiguration means for wireless sensor nodes. The approaches researched are FPGA based and will be extended to a Reconfigurable Application-Specific Instruction-set Processor (RASIP). In the demo described here we are showing the effects of exchanging parts of the hardware accelerators for elliptic curve cryptography originally reported in [1].

The rest of this paper is structured as follows. We first describe the design of the ECC hardware accelerators. In section 3, we provide background information on the reconfiguration process. The demo set up is illustrated in section 4.

Manuscript received May 7, 2009. The work reported herein constitutes background for the ARTEMIS-2008-100032 SMART project.
Digital Object Identifier inserted by IEEE

II. The Flexible Elliptic Curve Design

A. Elliptic curve cryptography background

Finite field arithmetic is the fundamental backbone of many approaches in cryptography and coding theory. Binary finite fields ($GF(2^m)$) provide efficient algorithms and implementations of the arithmetic operations. For example, additions and subtractions in $GF(2^m)$ are very fast because they can be implemented as simple XOR operations without carry propagation. This renders these fields very favorable for cryptographic applications with long key lengths. Several elliptic curves that are for example recommended for Elliptic curve cryptography (ECC) by the National Institute of Standards and Technology (NIST) [2] use these binary fields.

Elliptic curve cryptography (ECC) provides the same features as for example RSA, i.e. it can support digital signatures, key exchange authentication etc. due to the fact that two key – a secret and a public one are used per individual communication partner. ECC is much better suited for use in WSN due to the fact that it achieves the same level of secrecy with much smaller key length than RSA e.g. The level of secrecy achieved by a 233 bit ECC key corresponds to the one achieved by a 2048bit RSA key. This reduces the energy consumption for communication, and the mathematical operations of ECC are also less computational intensive as those of RSA. For these reasons we are focusing on ECC.

B. ECC Operation and flexible Design

The most important operation in $GF(2^m)$ is the polynomial multiplication. Regarding flexibility the actual multiplication is not the biggest issue. For example, a 233 bit multiplier can multiply 163 bit values, if unused bit positions are padded with zeros. The actual problem is that each multiplication of two elements of $GF(2^m)$ results in a product with a size of 2m-1 bit, which is too long for the finite field $GF(2^m)$. The conversion to an equivalent element in the borders of the finite

field is called reduction. Corresponding to classic finite fields the reduction is a division with remainder, i.e. long product is divided by the irreducible polynomial, which defines the finite field. But this modulo operation is too slow to be feasible. The hard wired reduction (HWR) is the state of the art [3, 4]. It is very fast, very small, but it is tailored for exactly one field. With the knowledge of the irreducible polynomial it is possible to build a chain of XOR operations that performs the reduction within one step as a direct mapping from the long product to the m bit polynomial. But such a chain is bound to one field size and one irreducible polynomial. The slightest change results in a completely different XOR chain.

In order to provide the envisioned SMART node with a means to support several elliptic curves we realized the HWR for five NIST curves. These reduction units can be integrated with the more general multiplication unit which can support multiplication of up to 571bit long polynomials. Figure 1 shows the overall design consisting of the multiplier controller, the operand selection, the partial multiplier and the reduction. The reduction units in light grey with the dashed border are non activated reduction units which will be integrated in the design and replace the currently active when required. The exchange of the reduction units on the FPGA board used by the smart modes is explained in the following section.

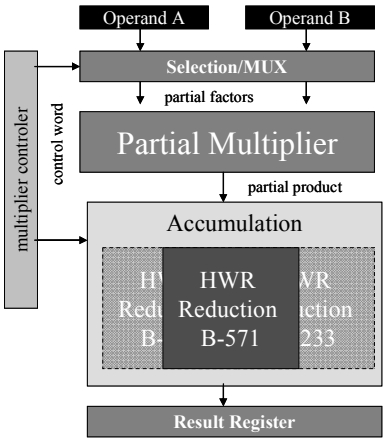

Fig. 1. Structure of the polynomial multiplication unit, indicating the different HWR given as dashed boxes..

III. Partial Reconfiguration in WSNs

Typically, reconfigurable devices like CPLDs or FPGAs are not included in sensor nodes, mainly due to their high power consumption. On the other hand, solutions in which such devices are integrated can target a wider set of computationally intensive applications, as well as they can add flexibility to the sensor node. For instance, in the 'Cookie' platform used in this work, a Spartan 3 FPGA from Xilinx and an ADuC841 microcontroller from Analog Devices are used to carry out processing tasks. In the near future, a new processing subsystem version, composed of a Texas MSP430 microcontroller and an Actel Igloo FPGA will be available. This low power version will consume 10 µA in sleep mode, which is a real competitive power consumption value compared with the state of the art platforms. As a reference, the node version used in this paper consumes 30 mA, while the widely used Crossbow TelosB platform that does not include HW reconfigurability consumes 5.1 µA (in standby mode). Even though, some solutions for WSN nodes include reconfigurable HW elements. A modular reconfigurable platform has been developed by Microsoft Research Labs [5]. In this platform, several layers can be added to the final platform, with several microprocessors depending on the application requirements, and the reconfigurable HW (CPLD in this case) is used to achieve communication abstraction through the entire modular platform.

In the SMART project, a new reconfigurable platform will be developed with the objective of achieving low power consumption, but with the aim to cope with today's high end tasks (for WSN levels) like video processing and transmission, or communications encryption and compression, and having in mind that these tasks might rely on reconfigurable HW.

Reconfiguration of WSN nodes involves the modification of both the SW running on the microcontroller and the HW that is programmed in the FPGA. Additionally, reconfiguration can be useful for two general scenarios: reconfiguration at network level.

Reconfiguration at network level is used during deployment. The function of each node is defined (including sensor interfaces, data processing, encryption and other parameters) or, when the network function is changed (like in an emergency situation).

The second scenario, reconfiguration at node level, mainly involves HW reconfiguration, where computation intensive tasks take advantage of the HW parallelism in order to lighten the microcontroller, like in [6].

On the other side, reconfiguration of the FPGA can be either full or partial. One advantage of exploiting partial reconfiguration, apart from the possibility of changing part of the configuration in run-time, is that partial configuration files are much smaller than complete ones, requiring lower memory, lower bandwidth and lower energy consumption. A virtual architecture is needed within the FPGA, where reconfigurable areas and interfaces between reconfigurable blocks are defined and maintained in all configurations.

Two methods are defined for reconfigurable file provision: they can be set through the network, or they can be stored in any non-volatile memory inside the node. The first method is mostly used at network level reconfigurations, and the second one at node level. Low performance FPGA families from Xilinx do not include ICAP (Internal Configuration Access Port), which is an internal hw fixed element which enables fast reconfiguration, including partial reconfiguration. In our case, no ICAP is available, so the microcontroller uses an output port to act as a JTAG controller, from which the programming files, either full or partial, can be set into the FPGA. This method enables any type of reconfiguration, nut with the speed limitations of the serial access mechanism provided by the JTAG interface. The microcontroller might either take the programming files from the associated flash memory or from

the network, depending on the operation type.

Next figure shows the control layer of the 'Cookie' sensor node developed at UPM, with the microcontroller and the FPGA. The bottom part shows the complete layered structure, which includes, apart from the control layer, with power supply, sensor and communication layers.

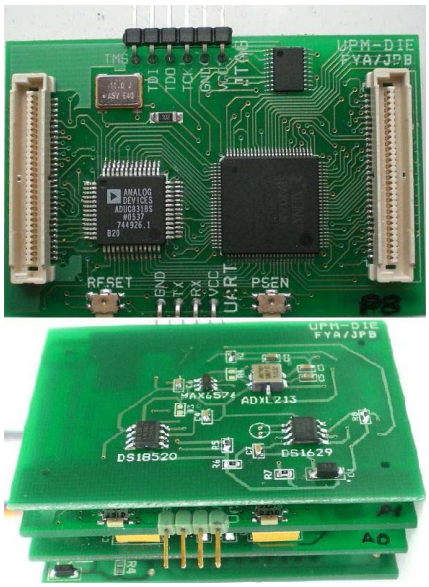

Fig. 2. Cookie processing node and complete Cookie node layered structure.

IV. Demonstrator Set-up

The experiment described in this paper shows the benefits of using reconfigurable HW for the communication encryption tasks, showing that reconfiguration is feasible in such feature-restricted platforms. It shows that features like updating the security algorithms –e.g. when they are broken or higher security is requested - are becoming feasible. By supporting new algorithms in hardware – as shown here - increasing the power consumption of the resource constraint devices can be avoided.

Visualizing security is extremely difficult. In the best case, i.e. all security features work fine nothing is to be seen. In order to illustrate the different ECC to be supported by the SMART node we will use the varying complexity of their computation. So, the demonstrator set-up will consist of:

- A standard wireless sensor node i.e. an Olimex MSP430-easyWeb2 equipped with an MSP430F149 and a 2x16 LCD.
- A Spartan 3 FPGA
- A host system (laptop) for controlling the application, input of user data etc.

The Olimex node and the FPGA will be connected by a cable to allow reliable data exchange. On both types of devices we will run ECC. The software executed on the Olimex is based on the Miracl library [7]. On the FPGA our ECC design will be used for the crypto operations. The elliptic curve under use will be exchanged on request. When changing the curve the processing time of the OLMEX node is reduced/increased significantly depending on whether the switch is from long key sizes to small ones or vice versa. By that interested SECON participants can experience the change directly. In addition the text sent will be decrypted on the receiver side (Olimex or FPGA) and then be displayed at the screen to show that both ends of the connection have used the same algorithm. The following figure shows the experimental set-up. The user can insert data at the laptop, they will be transferred to the Spartan FPGA, there they are encrypted and sent to the Olimex sensor node. On the node the text will be decrypted and displayed on the onboard LCD. By involving SECON participants and their own text we assure that all operations are done in real time.

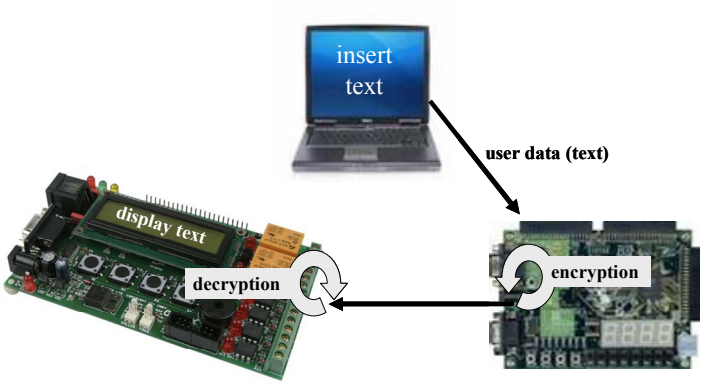

Fig. 3: Demonstrator set-up

Acknowledgment

This work was supported in part by the ARTEMIS JU under the ARTEMIS-2008-100032 SMART project.

References

[1] St. Peter, P. Langendörfer Flexible Hardware Reduction for Elliptic Curve Cryptography in GF(2m) Proceedings of Design Automation and Test in Europe (DATE) Conference, IEEE Society Press, 2007

[2] F. U.S. Department of Commerce/NIST. Digital Signature Standard (DSS), FIPS PUB 186-2, Jan. 27, 2000.

[3] H. Eberle, N. Gura, and S. C. Shantz. A cryptograhpic processor for arbitrary elliptic curves over. In ASAP, 2003.

[4] N. A. Saqib, F. Rodr´ıguez-Henr´ıquez, and A. D´ıaz-P´erez. A parallel architecture for fast computation of elliptic curve scalar multiplication over GF(2m). In IPDPS, 2004..

[5] D. Lymberopoulos, N. B. Priyantha, F. Zhao, "mPlatform: a reconfigurable architecture and efficient data sharing mechanism for modular sensor nodes," in Proc. of the 5th IEEE/ACM International Conference on Information Processing in Sensor Networks, IPSN'07, pp. 128-137, April. 2007

[6] H. Hinkelmann, P. Zipf, M. Glesner, "Design concepts for a dynamically reconfigurable wireless sensor node", in Proceedings of the1st NASA/ESA Conference on Adaptive Hardware and Systems, AHS'06, pp. 436-441, Jun 2006UPM

[7] Shamus Software, Multiprecision Integer and Rational Arithmetic C/C++ Library, available at www.shamus.ie/index.php?page=elliptic-curves