Public key cryptography empowered smart dust is affordable

Steffen Peter
Peter Langendörfer*
Krzysztof Piotrowski
IHP GmbH, Frankfurt/Oder, Germany
E-mail: {peter, langendoerfer, piotrowski}@ihp-microelectronics.com
*Corresponding author

Abstract: Public key cryptography (PKC) has been considered for a long time to be computationally too expensive for small battery powered devices. However, PKC turned out to be very beneficial for issues such as key distribution, authentication etc. In the recent years first research groups started to cope with the challenges applying PKC in resource-constrained environments. One result is that in particular ECC seems to be very suitable for such environments, because it provides the same level of security as RSA does while requiring much shorter keys. In this paper we evaluate the power consumption resulting from using various PKC approaches with respect to calculations and transmission of signatures etc. Our findings here clearly indicate that software realisations of PKC lead to relatively long duty cycles (operating intervals) which in turn require significant amount of energy. In contrast, the energy required for computation is negligible if the PKC is performed by power efficient hardware accelerators. In such cases the corresponding transmission power becomes much more significant. So we argue for dedicated hardware for elliptic curve cryptography in order to reduce energy consumption and to prolong life time of sensor nodes. Since additional hardware equals to additional cost, we are focussing on hardware accelerators that are optimised with respect to silicon area consumption. Our solution that supports an ECC key length of 163 bit takes about 1.02 mm^2 cell area in a 0.25μm technology and needs about 12.8 μWs per point multiplication. Due to its small size the accelerator can be manufactured for about 0.05 USD in mass production.

Keywords: Wireless Sensor Networks (WSN), Public Key Cryptography (PKC), Elliptic Curve Cryptography (ECC), Power Consumption, Hardware Accelerator

Biographical notes: Peter Langendoerfer received his diploma in computer science from the Technical University of Braunschweig in 1995 and his doctoral degree from the Brandenburg University of Technology at Cottbus (BTU) in 2001. Since 2000 he is with the IHP in Frankfurt (Oder). There, he is leading the mobile middleware group. He has published more than 55 refereed technical articles, filed five patents in the security/privacy area and worked as guest editor for the Journal of Super Computing (Kluwer), Computer Communications (Elsevier), Wireless Communications and Mobile Computing (Wiley) and ACM Transactions on Internet Technology. He is/was a TPC member of Globecom, VTC, ICC, WWIC and Wirelesscom and many more conferences. His research interests include mobile communication (especially privacy and security issues), protocol engineering, and automated protocol implementation.

Krzysztof Piotrowski received his master of computer science degree in 2004 from the University of Zielona Gora, Poland. After receiving the degree he joined the IHP. He is currently a member of the mobile middleware group working on solutions for security issues of wireless sensor networks. He has published about 15 refereed technical articles. His research interests include security and privacy issues in wireless networks especially in resource constrained environments as well as cryptographic and cryptanalytic methods for developing and evaluating payment systems.

Steffen Peter received his diploma in computer science from the Brandenburg University of Technology at Cottbus (BTU) in 2006. He is currently member of the mobile middleware group where he is working in the research of solutions for security issues of wireless sensor networks. He has published 6 refereed technical articles. His research interests include security and privacy in mobile environments with emphasis on efficient hardware implementation for this purpose.

1 INTRODUCTION

Wireless sensor networks (WSN) are considered to be the enablers for a wide range of new applications. Already discussed scenarios are ranging from support applications in the agriculture domain, i.e., monitoring of weather conditions or surveillance of cattle herds, over health monitoring for buildings and persons to military operations and homeland security scenarios. The level of security that has to be ensured in those areas varies and is increasing from the agriculture domain to military and homeland security. But the concrete level of security depends on the application that runs on top of the sensor nodes or is using their data. Imagine that WSNs are used in an agriculture scenario to ensure that the plants are treated in accordance to a certain quality procedure, i.e., to the BIO label, which is quite popular in the EU. In that case monitoring parameters such as fertiliser concentration are very important, and their manipulation may be in the interest of the farmer. At least data integrity is required in such an application to ensure reliable results. So there is no application area that does not have security requirements.

Security of WSN has attracted a lot of research interest during the last years. Security solutions based on symmetric key cryptography have been the major focus due to the very constrained resources such as processing and battery power as well as memory. A lot of work has been done to solve the key distribution and authentication problems (Camtepe and Yener, 2004; Chan et al, 2003; Du et al, 2003; Eschenauer and Gligor, 2002; Liu and Ning, 2003; Zhu et al, 2003; Perrig et al, 2001). During the last two years PKC has attracted some attention, due to its capability to simplify tasks such as key distribution and ensuring data integrity. Some papers directly address the feasibility of using PKC in WSN by evaluating parameters such as memory (Bazizi, 2003) and processing time (Malan et al, 2004; Glaubatz et al, 2004). Others are discussing new architectures that try to exploit the different efforts needed to run public and private key operations, to reduce the processing burden of the wireless sensor nodes (Zhang et al, 2005; Watro et al, 2004; Ning et al, 2005; Wang et al, 2006; Gupta et al, 2005; Watro et al, 2004). All these approaches have their own merit, but they still try to circumvent the problem at hand: processing PKC operations on small microcontrollers. It takes a long time and thus a serious amount of energy. This is unacceptable for WSN that shall be in place and operating unattended for decades. Therefore we argue for equipping sensor nodes with hardware accelerators, which help to reduce the energy consumption per PKC operation dramatically. We are focussing on elliptic curve cryptography (ECC) since it provides a good level of security even with relatively short key sizes. This helps to keep the following parameters small: memory needed to store keys, number of bits to be transmitted additionally if a certain packet is signed and area needed to realise an appropriate hardware accelerator. Beneath this considerations there also formal aspects that back up the decision to focus on ECC, e.g., the fact the US military simply forbids to use RSA when ciphering data but insists in using ECC (Lattin, 2006).

Copyright © 200x Inderscience Enterprises Ltd.

In this paper we evaluate the power consumption resulting from using PKC with respect to calculations and transmission of signatures etc. Our findings here clearly indicate that software realisations of PKC lead to relatively long duty cycles (operating intervals) which in turn requires significant amount of energy. In these cases the influence of the transmission power on the life time of the sensor node is negligible. So we argue for dedicated hardware for elliptic curve cryptography in order to reduce energy consumption and to prolong life time of the sensor node. The benefit of ECC is that it provides the same level of security as RSA does while requiring much shorter keys. This leads to reduced processing effort and helps to reduce the energy needed to transmit e.g. a digital signature. As long as PKC operations are executed in software, the transmission power that is additionally needed is negligible compared to the processing energy. But if much more power efficient hardware accelerators are used the transmission power becomes significant, and here the short key length of ECC provides a significant advantage. Since transmission energy is a significant factor of the sensor node life time, the key length is of high importance. The additional hardware needed for PKC operations results in additional cost. So we are focussing on hardware accelerators that are optimised with respect to area consumption. Our solution takes about 1.02 mm^2 in a 0.25μm technology and needs about 12.8 μWs for an elliptic curve point multiplication. Due to its small size the accelerator can be manufactured for about 0.05 USD.

The rest of this article is structured as follows. We first discuss related work. Section 3 then investigates the power consumption of software implementations of ECC on microcontrollers. There we also take into account the transmission energy. Our ECC hardware design is presented in section 4, including a short introduction of ECC background. Thereafter we evaluate the life time of a sensor node with respect to energy that is needed for the PKC operations. The paper concludes with a short summary and an outlook on further research.

2 RELATED WORK

In this section we focus on papers reporting on implementation issues of ECC for resource-constrained devices. Indeed, beside PKC other approaches (usually based on symmetric cryptography) have been proposed in order to realise key exchange, signatures and the like on resource-constrained devices, e.g. TESLA (Perrig et al, 2000) or SPINS (Perrig et al, 2001). However these solutions require complex protocols that suffer from other constraints. For example TESLA needs synchronized clocks. Since it is the scope of this paper to point out that PKC operations can be used by very resource-constrained devices we do not provide detailed analysis of this kind of protocols.

2.1 Software Implementations

Malan et al (2004), Gura et al (2004) and Wang et al (2006) are discussing the performance of their software implementations of ECC on microcontrollers. Malan et al (2004) used elliptic curves (EC) defined over binary Galois field (GF(2^m)). This class of EC is known to be predestined for implementation in hardware and to lead to quite slow software implementations. The two other implementations used EC over prime Galois fields (GF(p)), which can be implemented very efficient in software. These implementations clearly outperform Malan et al (2004), and the implementation reported in Gura et al (2004) is still three times faster than the one reported in Wang et al (2006). Due to its superior performance we only use Gura et al (2004) in comparisons in the rest of this papers.

2.2 Hardware Implementations

In Glaubatz et al (2004) hardware implementations of Rabin's Scheme and NtruEncrypt are investigated. We consider the former as out of scope since it requires pretty long keys, leading to significant processing and transmission power. The latter might be an alternative to ECC since its key length is in the same range as that of ECC and the energy needed to encrypt a single bit at a clock frequency of 500KHz is about 400pWs in a highly parallelised implementation. Dedicated ECC hardware accelerators have been presented in Satoh and Takano (2003), Gura et al (2002), Orlando and Paar (2000), Saqib et al (2004), Wolkerstorfer (2005) and Elliptic semi. (2006). They will be discussed in more detail in section 4.2.6 in relation to our design. In order to realise our own ECC accelerator we used the approaches published in Dyka and Langendoerfer (2005), López and Dahab (1998) and Itoh and Tsujii (1988). In Dyka and Langendoerfer (2005) the authors presented an iterative application of the Karatsuba method, that helps to realise area efficient polynomial multipliers. López and Dahab (1998) and Itoh and Tsujii (1988) presented optimised algorithms version of the Montgomery point multiplication and computation of the multiplicative inverse in GF(2^m), respectively.

3 POWER CONSUMPTION OF SENSOR NODES

In this section we will provide data about the power consumption of sensor nodes when executing PKC realised as software. We first analyse the most commonly used micro controllers. Based on these results we calculate the energy needed to execute PKC operations. Finally we estimate the energy consumed to send the resulting data and compare it to the processing energy.

3.1 Sensor node

The sensor nodes we are focussing on in this paper can be divided into two groups depending on the processing unit. The first group is the Mica family (Crossbow, 2005) (MICA2DOT, MICA2 and MICAz), based on the ATmega128L (Atmel, 2006) microcontroller from ATMEL. The second group includes sensor nodes based on the MSP430F1611 from Texas Instruments (2005), like TelosB (Crossbow, 2006) and Tmote Sky (Moteiv, 2006). Since the design of the Tmote Sky is based on TelosB in this paper we will refer to TelosB only.

Table 1: Time needed by the sensor nodes to perform SSL/TLS handshake(Gupta et al, 2005) and the resulting Performance ratios (PR) for RSA and ECC with MICA2DOT as normalisation basis

Sensor node	RSA-1024 handshake	ECC-160 handshake	PR	
			ECC	RSA
MICA2DOT	22.00 s	1.60 s	1.00	1.00
MICA2/MICAz	12.00 s	0.87 s	1.85	1.83
TelosB	5.70 s	0.50 s	3.20	3.86

We used the information from the microcontrollers' documentations (Atmel, 2006; Texas Instruments, 2005) to calculate the overall energy consumption and also the amount of energy consumed per clock cycle. In each case the estimated power consumption is calculated at 3V power supply voltage and at the maximal clock frequency as specified for the node. The TelosB with TI MSP430F1611 running at 8 MHz requires only 1.5 nWs per clock cycle, whereas the MICA2DOT with ATMEL ATmega128L at 4 MHz, and the MICA2 and MICAz with ATmega128L at 7.37 MHz need 4.1 nWs and 4.0 nWs respectively. This shows that the MSP430 requires only about 40 percent of the energy consumed by ATmega running at about the same clock frequency. As a basis for further calculations we determined the performance ratios of the different sensor node types. For the MICA family we use the clock frequency ratio without further consideration, since they used the same microcontroller but at different frequencies.

In Gupta et al (2005) the time to perform an SSL/TLS handshake using RSA as well as ECC was measured on TelosB and MICA nodes. Thus, these measurements are ideal to calculate the performance ratio of the two kinds of nodes. Table 1 shows the measurements from Gupta et al (2005) as well as the performance ratios we calculated, using the results for MICA2DOT node as normalisation basis.

The computing performance of the TelosB is about 3.2 compared to the performance of the Mica2Dot. Compared to the Mica2/MicaZ nodes the TelosB is still about 1.75 times faster. This advantage results from its 16-bit processing unit.

Based on processing time and energy consumption, we calculated the power consumed by the nodes while processing the above mentioned operations (see Table 2). Using these results we create another factor, the power consumption ratio - the power consumed by the cryptographic operations normalised using the power consumed by the least effective node. Since the MICA family uses the same micro processor the difference in the power consumption of

Table 2: Power consumption and power consumption ratios (PCR) of the sensor nodes to perform SSL/TLS handshake

Sensor node	RSA-1024 handshake	ECC-160 handshake	PCR	
			ECC	RSA
MICA2DOT	363.0 mWs	26.4 mWs	1.00	1.00
MICA2/MICAz	360.0 mWs	26.1 mWs	0.99	0.99
TelosB	68.4 mWs	6.0 mWs	0.23	0.19

Table 3: Power consumption for signature generation/verification on a Mica2Dot (Wander et al, 2005)

Crypto-system	Signature	
	Generation	Verification
RSA-1024	304.0 mWs	11.9 mWs
ECC-160	22.8 mWs	45.1 mWs
RSA-2048	2302.7 mWs	53.7 mWs
ECC-224	61.5 mWs	122.0 mWs

Mica2dot and MicaZ/Mica2 is negligible. But the TelosB node requires only 23 percent of the power consumed by the Mica nodes performing the same ECC operations. (see Table 2).

Knowing the performance and power consumption ratios for these sensor nodes we can proceed to a more detailed study on the power consumption of public key cryptography in WSN.

3.2 Cryptographic operations

In this section we are focussing on operations like encryption, decryption, signature generation and verification. Despite we are aware of the fact that other operations such as hash value calculations, random number generation and testing whether a number is a prime also contribute to the energy consumption of security means. In order to calculate the power consumption of both node families we used the measurements presented in Wander et al (2005) (see Table 3). In a first step we calculated the processing time per operation taking into account that in Wander et al (2005) the power consumption of active Mica2Dot is said to be 13.8mW. Then we calculated the power consumption of the Mica2Dot node using our estimations at 16.5mW from subsection 3.1. The power consumption values for Mica2/MicaZ and TelosB have been retrieved by applying the power consumption ratio calculated in the last subsection[1]. Tables 4 and 5 present the estimated power consumption and time needed by Mica2/MicaZ and TelosB nodes to perform signature generation and verification. Even for the most powerful TelosB the RSA private key operations are very time and energy consuming.

The numbers presented in Table 4 and Table 5 clearly show that the use of RSA in sensor networks is infeasible. But they also indicate the asymmetry in the effort of RSA public and private key operations. So, as long as only public key operations have to be performed RSA is a reasonable choice, but as soon as mutual authentication between nodes is requested RSA is no longer a candidate.

[1]Please note that we have omitted the intermediary results for clarity reasons

Table 4: Estimated time and power consumption for signature generation/verification on a Mica2/MicaZ

Crypto-system	Signature	
	Generation	Verification
RSA-1024	359.9 mWs 12.04 s	14.0 mWs 0.47 s
ECC-160	27.0 mWs 0.89 s	53.4 mWs 1.77 s
RSA-2048	2725.7 mWs 91.18 s	63.6 mWs 2.13 s
ECC-224	72.8 mWs 2.41 s	144.4 mWs 4.78 s

Table 5: Estimated time and power consumption for signature generation/verification on a TelosB

Crypto-system	Signature	
	Generation	Verification
RSA-1024	69.0 mWs 5.66 s	2.7 mWs 0.22 s
ECC-160	6.3 mWs 0.52 s	12.4 mWs 1.02 s
RSA-2048	523.1 mWs 42.89 s	12.2 mWs 1.00 s
ECC-224	16.9 mWs 1.39 s	33.5 mWs 2.76 s

The ECC performance is quite balanced and if public and private key operations are taken into account, it is much better than the one of RSA. But spending more than a second to generate or verify a single signature is still too expensive if very long lifetimes, e.g. several years shall be ensured. The time needed for a cryptographic operation limits also the maximum frequency of its occurrence. In most cases it should not be a problem, but imagine a situation where a sensor node has to sign or encrypt every reading it makes. If RSA-1024 is applied, the TelosB sensor node needs more than 5 seconds for signing its readings. This limits the sensing rate to a maximum of once every 5 seconds. In addition it results in a duty cycle of 100 percent, which will lead to a very short life time of the battery powered sensor node. Using ECC-160 for signing or encrypting the sensor data would reduce the consumed time to less than 1 second. This would reduce the duty cycle of the sensor node to 10 percent, in case the sensing rate is kept constant, and extend the node's lifetime by factor ten.

In addition to the calculation power the transmission power has to be considered when estimating lifetime.

3.3 Power Consumption of Transmission

An important issue for the applicability of PKC is the energy consumption introduced by sending signed or encrypted data. Here, in case of encryption, we only consider packets in the size of the used key or smaller, since for all other packets the much more efficient symmetric cipher mechanisms will be applied.

The energy needed to transmit a bit depends on the transceiver used, as well as on the protocols and the channel state. But in this paper we intentionally neglect the latter facts, e.g. we do not take into account retransmissions of packets. Since RSA and ECC would be used under

Table 6: Power consumption (PC) of ZigBee transceiver CC2420

Type of communication	Current consumption [mA]	PC (@U=3V) [mW]	PC per bit (250.4 kBit/s) [μWs/bit]
RX	18.8	56.4	0.226
TX -5 dBm	14.0	42.0	0.168
TX 0 dBm	17.4	52.2	0.209

Table 7: Power consumption (PC) of the 433 MHz and 868 MHz transceiver CC1000

Type of communication	Current consumption [mA]		PC (@U=3V) [mW]		PC per bit (38.4 kBit/s) [μWs/bit]	
MHz	433	868	433	868	433	868
RX	7.4	9.6	22.2	28.8	0.578	0.750
TX -5 dBm	8.9	13.8	26.7	41.4	0.696	1.078
TX 0 dBm	10.4	16.5	31.2	49.5	0.812	1.290

exactly the same conditions, this assumption of an idealised world has only little influence on the result. The ratio between energy needed when RSA is used and when ECC is used should be nearly constant. We are aware of the fact that ECC based approaches will suffer less from bad channels states, due to their shorter packet size.

All four types of sensor nodes use single chip transceivers. Mica2 and Mica2Dot use 433 MHz or 868 MHz radio chip CC1000 (Texas Instruments, 2006a) and MicaZ and TelosB use ZigBee 2.4 GHz radio chip CC2420 (Texas Instruments, 2006b). The two radio types differ in performance. ZigBee devices transmit data with 250 kbit/s data rate with maximum power of 0 dBm and CC1000 chip allows data rates up to 76.8 kbit/s with maximum power of 10 dBm (433 MHz) or 5 dBm (868 MHz). The Mica nodes, which use the cc1000 chip applies Manchester encoding reducing the maximum transmission rate to 38.4 kbit/s.

For fairness reasons, we compare the energy efficiency of both chips at transmission power supported by both, i.e. at -5dBm and 0dBm, see Table 6 and Table 7. This data shows that the higher power consumption of cc2420 is compensated by the lower cost of per bit transmission.

The influence of the cryptographic means applied on the energy consumption depends on the amount of data that has to be transmitted. We decided to use the transmission of a digital signature for the evaluation, since its size is exclusively determined by the cipher mechanism applied. The RSA signature is represented by an integer smaller than the used modulus, and in case of ECDSA the signature are two integers smaller than the order of the base point of the used curve. Thus, in case of RSA signature the size of it is about the key size, and for ECDSA the size of a signature is about double the key size.

The costs of reception of signatures of different lengths are presented in Table 8 for both transceivers. The power consumption of sending those signatures is shown in Table 9. Comparing these values with those presented in Table 3 and Table 4 clearly indicates that the energy needed for sending signatures is negligible if they are calculated in software. For the TelosB node the cost of the communication is about 1 percent for ECDSA-160, for Mica2Dot and Mica2 it is about 2 percent. For MicaZ the significance of communication costs goes below 0.3 percent. In order to point out very clear that the applicability of PKC does not depend on power consumed by transmitting keys, signatures etc. we assume the following worst case scenario. The least power hungry micro controller (MSP430) is used in combination with the most power hungry transceiver (cc1000), so the energy for signature generation is minimised whereas the energy for transmitting it is maximised. Even in this scenario the transmission energy is still less than ten percent, and receiving power is less than four percent of the calculation power.

Table 8: Power consumed during reception of a signature on cc2420 and cc1000 single chip transceiver

Signature	Size [bit]	cc2420 [μWs]	cc1000 433 MHz [μWs]	cc1000 868 MHz [μWs]
ECDSA-160	320	72.32	184.96	240.00
RSA-1024	1024	231.42	591.87	768.00
ECDSA-224	448	101.25	258.94	336.00
RSA-2048	2048	462.85	1183.74	1536.00

Table 9: Power consumed while sending a signature on cc2420 and cc1000 single chip transceiver with -5 dBm and 0 dBm output power

Signature	Size [bit]	cc2420 [μWs]	cc1000 433 MHz [μWs]	cc1000 868 MHz [μWs]
Output power -5 dBm				
ECDSA-160	320	53.76	222.72	344.96
RSA-1024	1024	172.03	712.70	1103.87
ECDSA-224	448	75.26	311.80	482.94
RSA-2048	2048	344.06	1425.41	2207.74
Output power 0 dBm				
ECDSA-160	320	66.88	259.84	412.80
RSA-1024	1024	214.01	831.49	1320.96
ECDSA-224	448	93.63	363.78	577.92
RSA-2048	2048	428.03	1662.98	2641.92

4 HARDWARE

In the previous sections we exposed that the computation of the operations needed for the asymmetric cryptography requires huge amount of runtime and power. A solution that can improve both issues is dedicated hardware that performs the critical operations. In this section we evaluate an efficient hardware design that accelerates the cryptographic operations and reduces the required energy. Afterwards we compare the properties with previous hardware solutions and evaluate the corresponding impact, in particular on the power consumption, for the sensor nodes.

4.1 Elliptic Curve Cryptography

The elements of the elliptic group $\mathbb{E}$ are two-dimensional points with x and y coordinates. Every point (x, y) on the curve satisfies an equation such as

$$y^2 + xy = x^3 + ax^2 + b, \quad (1)$$

where a and b are parameters defining the curve. The coordinates x and y are members of a finite field. Even though every finite field is feasible for ECC, we restrict our evaluation to ECs over GF(2^m), because of the binary character of the coefficients and not needed carries, what render them especially suited for hardware implementations. The parameter m is the bit length of the coordinates of the EC. In this work we describe an accelerator for the 163 bit elliptic curves in GF(2^{163}) as they are for example recommended by the NIST (2000). Since elliptic curves on these fields have been standardised and analysed for years, high level of experience and security has been achieved. This provides a fundamental basis for special ECC hardware accelerators.

The group of points that satisfy the initial EC-equation provide an additive finite Abelian group. This implies that the additive operation is defined in a way that every addition of two elements of $\mathbb{E}$ results in another element that belongs to the group ($C = A + B$). Having the addition of two points, one can define the repeated addition ($Q = P + P + ... + P$) as the scalar multiplication of a point by an integer k, ($Q = k \cdot P$). This elliptic curve point multiplication (ECPM) is the most important operation in ECC. The fundamental idea of ECC is based on the assumption that it is computationally infeasible to invert the ECPM, i.e. to find k for given Q and P. The acceleration of that ECPM is the primary target of the hardware design presented in this paper.
Several algorithms for the computation of the ECPM have been proposed. An overview is presented in Hankerson et al (2000). The Montgomery point multiplication (MPM), introduced in Montgomery (1987), is the fastest known ECPM approach for hardware designs for random ECs over $GF(2^m)$. An improved version of the algorithm was presented by López and Dahab (1998) and is shown in Algorithm 1. It requires approximately $6m$ field multiplications, $5m$ field squarings, $3m$ field additions, and one multiplicative field inversion for one ECPM. The López and Dahab algorithm is the approach that is applied in our design.

Algorithm 1: Montgomery kP multiplication (Hankerson et al, 2000)

input : $k = (k_{t-1}, ..., k_1, k_0)_2$ with $k_{t-1} = 1$,
$P = (x, y) \in E(F_{2^m})$
output: $kP = (x_1, y_1)$

$X_1 \leftarrow x$; $Z_1 \leftarrow 1$; $X_2 \leftarrow x^4 + b$; $Z_2 \leftarrow x^2$;
for $i = t - 2$ **downto** 0 **do**
  if $k_i = 1$ **then**
    $T \leftarrow Z_1$; $Z_1 \leftarrow (X_1 Z_2 + X_2 Z_1)^2$;
    $X_1 \leftarrow xZ_1 + X_1 X_2 T Z_2$;
    $T \leftarrow X_2$; $X_2 \leftarrow X_2^4 + bZ_2^4$; $Z_2 \leftarrow T^2 Z_2^2$;
  else
    $T \leftarrow Z_2$; $Z_2 \leftarrow (X_2 Z_1 + X_1 Z_2)^2$;
    $X_2 \leftarrow xZ_2 + X_1 X_2 T Z_1$;
    $T \leftarrow X_1$; $X_1 \leftarrow X_1^4 + bZ_1^4$; $Z_1 \leftarrow T^2 Z_1^2$;
  end
end
$x_1 \leftarrow X_1/Z_1$;
$y_1 \leftarrow y + (x + x_1)\cdot$
$[(X_1 + xZ_1)(X_2 + xZ_2) + (x^2 + y)(Z_1 Z_2)]/(xZ_1 Z_2)$;
return $((x_1, y_1))$

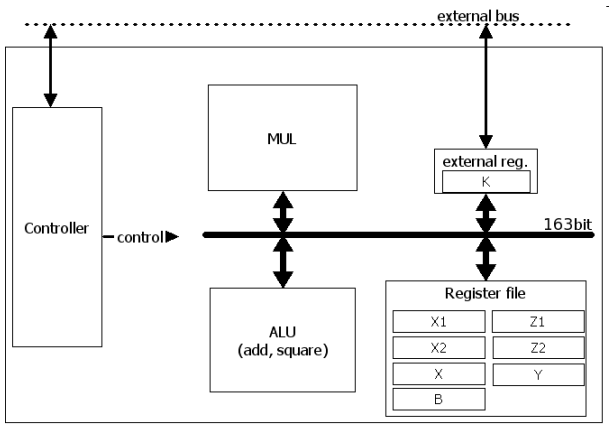

Figure 1: ECC 163 block diagram. Polynomial multiplier (MUL) and ALU with functionality for adding, squaring and manual setting of data words, are the 'working horses' of the chip. The controller block, which can be externally accessed, controls the bus access. The design has eight separate registers, seven in the internal register file and one that can also be accessed via an external bus.

4.2 Design of an ECC 163 Co-processor

In this section we describe the hardware that performs the ECPM. Actually, the ECPM algorithm is executed in a controller unit that controls bus access and the functional units (FUs) of the ECC design. The FUs perform field operations in GF(2^m) such as polynomial multiplication, addition and squaring. The multiplicative inversion in GF(2^m) is performed as subprogram of the controller program. The design described in this section has a default field size of $m = 163$. The data words are represented in polynomial basis, i.e. every stored bit represents one digit of the binary polynomial.
Figure 1 depicts the block diagram of the 163 bit ECC hardware accelerator. It shows the FUs, the registers, the 163 bit bus that connects the components and the control unit. The units are described in the following subsections.

4.2.1 ALU

The ALU combines the functionalities of addition, squaring, and setting the accumulator register (AX), which is embedded in the ALU. In GF(2^m), and therefore in the ALU, the addition is a simple XOR-operation of a provided input word and AX. The squaring operation computes squaring in GF(2^m) by inserting one zero between every input data bit, because in GF(2^m) squares can be

express as expressed in equation 2.

$$c(x) = a(x)a(x) = \sum_{i=0}^{m-1} a_i x^i \cdot \sum_{i=0}^{m-1} a_i x^i = \sum_{i=0}^{m-1} a_i x^{2i} \quad (2)$$

This operation is followed by a reduction of the long squaring result to a 163 bit value that fits the chosen field. The reduction is a hard-wired part of the squaring function and optimised for the particular finite field.
Indeed, the squaring operation could be performed by the multiplication unit. However, since the squaring function is much faster and much more efficient than a full multiplication, which requires several clock cycles for an operation, we decided to spent the dedicated squaring unit.
Every operation of the ALU, i.e. adding and squaring with reduction, is finished within one clock cycle. The ALU requires merely 0.08mm^2 silicon area in 0.25μm CMOS technology.

4.2.2 Polynomial Multiplier

The multiplier is the most important FU of the design. It is not only the largest unit but also the most utilised one. The duty time is more than 90 percent and usually it requires more than half of the accelerator's silicon area. This is why an efficient design of the multiplier is the key for an efficient ECC hardware design. It is also the reason why we take a closer look at the multiplication unit below.

The major issue with the polynomial multiplier is that especially combinatorial multipliers become very large and slow for longer factors. To counter this problem, in Dyka and Langendoerfer (2005) the iterative Karatsuba multiplier (IKM) approach was presented. It uses smaller combinatorial multiplication blocks, and applies them repeatedly following the Karatsuba method in order to perform a larger polynomial multiplication. The IKM design for a 233 bit multiplication unit presented in Dyka and Langendoerfer (2005) is the starting point for the investigation concerning improved IKM design that will be the most important functional unit of our ECC design. It consists of three main parts:

Selection logic: selects and combines the factors of the partial multiplication.

Partial multiplier: performs the partial multiplication within one clock cycle.

Accumulation block: computes the final product by accumulating the partial products.

The number of clock cycles and the required silicon area depend on the size of the segmentation. For our 163 bit ECC design we are considering the following IKM configurations:

- two-segment IKM requiring three clock cycles using an 82 bit partial multiplier
- four-segment IKM needs nine clock cycles of 41 bit partial multiplications
- eight-segment IKM with 27 clock cycles using a 21 bit partial multiplier

Table 10: Accumulation table of the IKM (Dyka and Langendoerfer, 2005)

Partial multiplication	Accumulations							
$[a_0 \cdot b_0][0]$					⊕	⊕	⊕	⊕
$[a_0 \cdot b_0][1]$				⊕	⊕	⊕	⊕	
$[a_1 \cdot b_1][0]$				⊕	⊕	⊕	⊕	
$[a_1 \cdot b_1][1]$			⊕	⊕	⊕	⊕		
$[a_2 \cdot b_2][0]$			⊕	⊕	⊕	⊕		
$[a_2 \cdot b_2][1]$		⊕	⊕	⊕	⊕			
$[a_3 \cdot b_3][0]$		⊕	⊕	⊕	⊕			
$[a_3 \cdot b_3][1]$	⊕	⊕	⊕	⊕				
$[(a_0 \oplus a_1) \cdot (b_0 \oplus b_1)][0]$					⊕		⊕	
$[(a_0 \oplus a_1) \cdot (b_0 \oplus b_1)][1]$				⊕		⊕		
$[(a_0 \oplus a_2) \cdot (b_0 \oplus b_2)][0]$					⊕	⊕		
$[(a_0 \oplus a_2) \cdot (b_0 \oplus b_2)][1]$				⊕	⊕			
$[(a_1 \oplus a_3) \cdot (b_1 \oplus b_3)][0]$				⊕	⊕			
$[(a_1 \oplus a_3) \cdot (b_1 \oplus b_3)][1]$			⊕	⊕				
$[(a_2 \oplus a_3) \cdot (b_2 \oplus b_3)][0]$			⊕		⊕			
$[(a_2 \oplus a_3) \cdot (b_2 \oplus b_3)][1]$		⊕		⊕				
$[(a_0 \oplus a_1 \oplus a_2 \oplus a_3) \cdot (b_0 \oplus b_1 \oplus b_2 \oplus b_3)][0]$					⊕			
$[(a_0 \oplus a_1 \oplus a_2 \oplus a_3) \cdot (b_0 \oplus b_1 \oplus b_2 \oplus b_3)][1]$				⊕				
	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0

Table 10 shows the accumulation scheme for the four segment version. Both factors are split in four segments so that the polynomial multiplication $A(x) \cdot B(x) = C(x)$ can be represented by

$$a_3a_2a_1a_0 \cdot b_3b_2b_1b_0 = c_7c_6c_5c_4c_3c_2c_1c_0.$$

The left part of the table shows the selection of the partial factors and the partial multiplication, and the right part show the accumulation plan.
For the hardware-IKM described in Dyka and Langendoerfer (2005), the selection and accumulation blocks of the multiplier become large with higher segmentation. This is due to a complicated data path that indeed reduces the number of total executed XOR operations, but leads to an irregular data path structure. We solved the issue by implementing a data path that does not reduce the number of operations but has a much more regular structure and thus requires less silicon area.

Applying IKM for the fourfold segmentation requires nine clock cycles to compute the final result of the multiplication. In each clock cycle a partial product of the size of $2n$ is computed. This partial product has to be accumulated to the determined position in the full product. For four segment IKM, seven different positions are possible. The positions can be represented by a seven bit command word which is generated by a small controller block. The value of this command word depends on the current clock cycle of the multiplication. The data path is organised as shown in Figure 2. If a command bit is set, the partial product is forwarded to the corresponding XOR operation, otherwise the XOR operation is performed with zeros, which results in no change at the relative position. Because of the overlapping XOR operations it is necessary to perform this process in two stages. The intermediate result after the first stage is not stored but is forwarded directly to the second stage. The result of the second stage

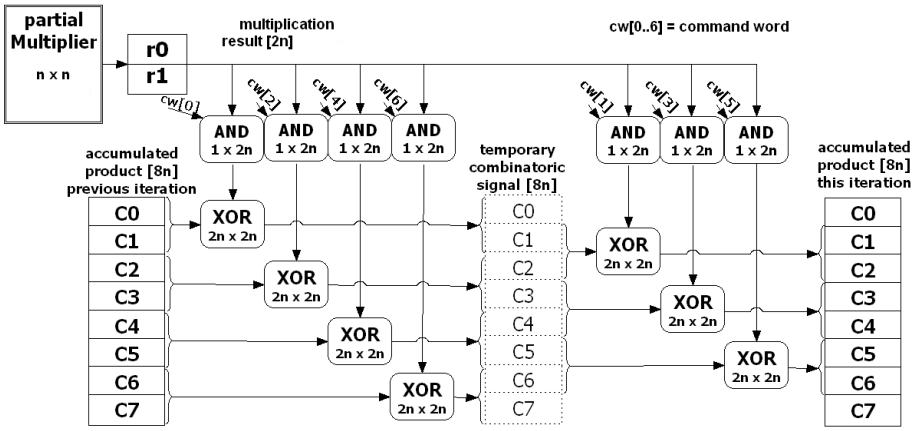

Figure 2: Highly regular configurable structure for the accumulation of partial products in the IKM process. The XOR operations are iteratively performed corresponding to the accumulation table (Table 10). For a four segment 163 bit multiplier n is 41 bits. With attached combinatorial reduction only c0 to c3 must be stored.

is stored in the result register, and is used again in the next iteration.
The selection process is done the same way. Small control words determine the XOR operations that have do be executed. Due to the very regular structure, our approach reduces the area required for the control logic of selection and accumulation to 0.14mm^2 from 0.39mm^2 needed by the original method.
A further improvement is the integration of the reduction step into the accumulation block. Traditionally, the reduction is performed after the multiplication is finished, i.e. after the nine partial multiplication steps were performed. This means that for the multiplication $c = a \cdot b$, c is $((p_0 + p_1 + ... + p_7 + p_8) \mod r)$, where p_i are the partial accumulations and $mod\ r$ represents the reduction with the irreducible polynomial. Instead, we perform a reduction after every iteration step, since it can be proven that $c = (p_0 \mod r + p_1 \mod r + ... + p_8 \mod r)$. Thus, the partial results $c4$, $c5$, $c6$, and $c7$, shown in Figure 2, do not need to be stored. Indeed, now the hard-wired reduction is integral part of the multiplier. For a single EC this design is very efficient. For the 163 bit four-segment multiplier, which requires nine clock cycles for the polynomial multiplication in GF(2^{163}) the silicon area is 0.45mm^2 measured for our 0.25 μm CMOS technology.
Additionally to the nine clock cycles for the computation, the multiplier-FU needs two clock cycles for setting up a multiplication. But the design allows to set up the next multiplication while the current one is still being computed. Thus, an efficient schedule can be provided, which does not contain set up cycles while the multiplier core is idle.

4.2.3 Registers

There are eight registers required for the execution of the MPM. The Register for the variable k is not embedded in the internal register file but is an external register that can be accessed through an external bus. This external register is the interface, where external units can write and read data words. The names of the register are based on the names in the inner loop of the MPM algorithm. They can be used independently of this algorithm, since every register can be read and written directly from the internal bus.

Table 11: Properties of GF(2^{163}) multiplication units synthesised for 33 MHz. Due to their execution speed, the 2-segment version requires the least total Energy per multiplication but needs the largest area. Our favourite design is the 4-segment version with a size of $0.45mm^2$ and an energy consumption of 8.5nWs.

Multiplier (163 bit)	Time [ns]	Area [mm^2]	Power [mW]	Energy [nWs]
4-segment	270	0.45	32	8.5
2-segment	90	0.79	48	4.3
8-segment	810	0.35	19	15.0

4.2.4 Controller

The control unit is the place where the bus access is managed and the main program is executed by accessing the FUs. In this block the MPM is executed, as it is described in Algorithm 1. The control unit manages the concurrent execution of the GF(2^m) operations, i.e. fast squaring and addition operation can be executed, while a field multiplication is performed. Additionally to the MPM, the controller also manages the execution of the multiplicative inversion in GF(2^m) using the Itoh-Tsuji approach presented in Itoh and Tsujii (1988). The needed 8 multiplications and 162 squarings require 245 clock cycles using the four-segment multiplier.
In the current design, the controller programs are hard wired. The selection of the program is done by a command word that is provided over the external bus.

Table 12: Post-synthesis parameters for the execution of one 163 bit ECPM on accelerator designs using different embedded polynomial multipliers simulated for 33 MHz.

Mul Setup	Size [mm^2]	Needed clk cycles	Time [ms]	Power [mW]	Energy [μWs]
163_8	0.90	26999	0.81	31	25.1
163_4	1.02	9251	0.24	46	12.8
163_2	1.36	5344	0.18	66	10.6

4.2.5 Results

The three designs (163_2, 163_4, 163_8) with field multiplication units of different speed, were implemented and synthesised as IC in order to determine their parameters. The required area, number of gates and maximum clock speed were measured for the 0.25μm CMOS technology. Additionally for each design the power consumption and required time were determined in a simulation environment by performing exemplary point multiplications. The results of these investigations are shown in Table 12. The power consumption was determined by Synopsys PrimePower (Synopsys, 2005b) and size is the cell area reported by Synopsys Design Compiler (Synopsys, 2005a).

The data show that the number of needed polynomial multiplication (986 field multiplications for one 163-bit ECPM) times the clock cycles per multiplication is a very good approximation of the total time required for one ECPM. It implies that the speed of the field multiplier is still the dominant factor of the design. Only the two-segment design needs more cycles than expected due to the bus access bottleneck.

Regarding the synthesis parameters, the 163_8 design is about 20 percent smaller than 163_4, but is also much slower. The threefold required time, compared to 163_4 is also the reason why the total energy for a full ECPM is the double of the four segment version, even though the average power consumption is lower.

Comparing the four segment multiplier with the two segment implementation, more than 30 percent larger area is needed by the latter, where the required time is not even the half. The speed advantage, compared to 163_4, is less than 30 percent. Despite this and the higher average power consumption for 163_2, its total energy consumption for a point multiplication is about 20 percent less than for 163_4. It is a classical trade-off between time and area that is shown by the three designs. Faster execution time implies more area, but also less total energy. The 163_4 solution has the most convenient parameters, since it is not as large as 163_2, but much faster than 163_8. The choice, which design should be used in practice, depends on the application area, in particular on the questions whether higher performance is worth more silicon.

In our WSN scenario we want both, the least possible power and chip area. This is why we have chosen the 163_4 design with about 1 mm^2 silicon area and less than 13 μWs energy consumption per ECPM as our default 163 bit ECC hardware accelerator.

Table 13: Comparison of $GF(2^m)$ ECPM hardware designs.

Ref	Field	Platform	Time	Size
our	GF(2^{163})	0.25 μm ASIC	0.08	1.0mm^2, 35Kgates
our	GF(2^{163})	Xilinx XC2VP70	0.11	5598 LUTs
Satoh (2003)	GF(2^{163})	0.13μm ASIC	0.19	117.5 Kgates
Gura et al (2002)	GF(2^{163})	Xilinx XCV2000E	0.14	19508 LUTs
Orlando (2000)	GF(2^{167})	Xilinx XCV400E	0.21	3002 LUTs
Saqib et al (2004)	GF(2^{191})	Xilinx XCV3200E	0.06	≈30000 LUTs
Wolkerstorfer (2005)	GF(2^{191})	0.35 μm ASIC	6.21	1.31mm^2
Elliptic semi. (2006)	GF(2^{233})	0.13 μm ASIC	6.68	71 Kgates

4.2.6 Comparison with Previous Hardware Approaches

Table 13 shows the parameters of previous hardware implementations of accelerators for EC scalar multiplication. Due to different hardware configurations and different amount of functionality the numbers cannot be compared directly. In order to provide a better comparability we listed the size for our design for both target platforms ASIC and FPGA.

The design presented in Satoh and Takano (2003) supports not only ECs based on binary extension fields but also curves on prime fields $GF(p)$. This renders the design that is presented as ASIC running at 510Mhz to the most configurable EC coprocessor. The hardware proposed in Gura et al (2002) also supports not only one curve but all ECs based on binary extension fields $GF(2^m)$ up to a size of $m = 256$. The design described in Orlando and Paar (2000) is a very area efficient implementation of an EC based on $GF(2^{167})$. It does not reach the speed of the fastest designs but is very small. With a LUT number of 3002 it requires about half of the area of our corresponding design on the FPGA. In contrast, there is about the doubled time compared to our implementation. The fastest known implementation has been reported in Saqib et al (2004). This design performs a ECPM on $GF(2^{191})$ within less than 60μs. On the downside this single curve implementation requires a huge amount of area. With its estimated 30000 LUTs it is five times larger but only slightly faster than our implementation. The ASIC design presented in Wolkerstorfer (2005) is another very small design. The ASIC manufactured in a 0.35 μm technology has a size of 1.31mm^2 and supports two fields, but requires more than 6 ms for an ECPM. The power consumption reported for this design is 213μWs for an ECPM in $GF(2^{191})$. It is, due to the poor performance, about the tenfold of our design. The commercially offered design by Elliptic semi. (2006) is another design that reports the power consumption. A 233 bit operation requires 6 ms and a total energy of 140μWs for the 50 MHz design, manufactured as 0.13 μm ASIC.

To summarise, our design outperforms most other approaches except those that were optimised for speed or area only.

4.3 Impact on Sensor nodes

We have realised two approaches to connect the microcontroller with the hardware accelerator, first a system on

Table 14: Performance data of ECC PKC operations executed on accelerator designs running at 8 MHz.

Bit-size	Size [mm^2]	Frequency [MHz]	Time [ms]	Power [mW]	Energy [μWs]
163	1.02	8	1.1	11.8	13
233	1.41	8	1.6	16.8	27

chip, and second an external coprocessor. The former requires a system bus that connects the microprocessor with the additional functional blocks - but it is all on one chip. The latter needs a separately packaged chip that generates additional costs. A benefit is that such an implementation can be used by every sensor node from the shelf. A third approach that we will investigate in detail in future work, is the extension of the instruction set of the processor with embedded cryptographic hardware blocks.
But independent of the actual implementation, regarding to our measurements, parameters of time and energy consumed by the cryptographic operations are reduced by three orders of magnitude. In section 3.1 we showed that an ECC-160 handshake, an operation that corresponds to the 163 ECPM, takes 0.5 - 1.6 seconds, dependent on the selected sensor node. The power consumption thereby varies from 6 to 26 mWs. In contrast our hardware solution needs 0.08 ms and 13 μWs. Even with a clock frequency of 8 MHz it requires about 1 millisecond (see Table 14). We evaluate the particular effect of the power and performance differences on the node's lifetime in the next section. The area needed to realise the hardware accelerators clearly indicates that cost is no longer an issue even for very inexpensive devices.

5 SENSOR LIFETIME ESTIMATION

In this section we discuss the sensor node lifetime with respect to their energy sources, i.e. batteries. We start with an estimation of the energy available from batteries and discuss which part of this energy can really be used by the sensor nodes. Based on these results we calculate the maximal lifetime of a sensor node with respect to the frequency in which PKC operations have to be executed.

5.1 Energy provided by batteries

The standard power source for the above mentioned nodes are batteries. The Mica2, MicaZ and TelosB nodes are powered by 2 AA cells and Mica2Dot is powered by CR2354 lithium coin cell battery. To estimate the available amount of energy we need to know the capacity of the batteries.

Alkaline battery The rated capacity of an AA alkaline battery is about 2500 mAh. However, the manufacturers define the capacity as the amount of energy that can be provided until the voltage of a single AA cell reaches 0.8 V. And since the sensor nodes are powered by two AA batteries the voltage of such a battery pack is 1.6 V.

The voltage of a new alkaline AA cell is usually about 1.6 V and as the current is drawn the voltage drops almost linear. We will use this effect to estimate the amount of energy that can be delivered by the double AA cell battery pack that delivers nominal 3.2 V. Assuming linear or almost linear voltage drop to 1.6 V the average voltage for the pack is 2.4 V. The product of time and current is said to be 2500 mAh, which means that the energy that could be delivered is equal to 6000 mWh or simply 21600 Ws.

The time within which the batteries reach the cut-off (1.6 V) voltage depends on the value of the current. We assume that the current is constant and divide the range from 3.2V to 1.6V into four intervals each representing a voltage drop of 0.4V. Thus, the energy capacity available by the battery pack is divided into four partitions depending on the voltage range as follows:

- 3.2 V – 2.8 V — 31.25 % of 21600 Ws → 6750 Ws,
- 2.8 V – 2.4 V — 27.10 % of 21600 Ws → 5850 Ws,
- 2.4 V – 2.0 V — 22.90 % of 21600 Ws → 4950 Ws,
- 2.0 V – 1.6 V — 18.75 % of 21600 Ws → 4050 Ws.

The energy that such a battery pack can deliver to a sensor node depends on the voltage range in which the node can operate properly. For instance, if a device accepts voltage range between 2.0 V and 3.2 V then the amount of energy available will reach 81.75 % of the whole battery pack capacity, i.e., the device can consume up to 17550 Ws and 4050 Ws will remain useless.

Coin cells The rated capacity of a cr2354 coin cell used by Mica2Dot is 560 mAh, and according to Panasonic (2006) the discharge characteristics is quite flat while discharging with a small constant current of about 0.5 mA. The starting voltage is about 2.9 V at room temperature and about 80 % of the energy capacity can be delivered until the voltage drops below 2.8 V. The rated energy capacity is about 5500 Ws.

5.2 Estimation of Exploitable Capacity

In this subsection we estimate the energy that is really available for the sensor nodes while powered by an AA alkaline battery pack. Both microcontroller types require a voltage higher than 2.7 V. For the ATmega128L microcontroller used by the Mica family this value is the minimum for operation. The MSP430F1611 used by TelosB can work even with a voltage of 1.8 V, but then its clock frequency is reduced, and it requires at least 2.7 V to be able to write to flash. So, the minimal acceptable voltage depends on the application, and could be adapted by sophisticated energy management strategies. Since we are interested in the worst case, we concentrate here on the voltage range in which all sensor nodes provide full functionality, i.e. on the range from 2.8 V to 3.2 V. Thus, the node powered by two

AA alkaline batteries uses only 31.25 percent of the total capacity, i.e., the node can consume about 6750 Ws until the batteries do no longer support the needed voltage. If a coin cell is used the node can use 4400 Ws, before the voltage drops below the 2.8V threshold.

5.3 Supported number of PKC operations

In subsection 3.2 we estimated the energy consumed by the cryptographic operations for a supply voltage of 3.0 V. Since this is exactly the mean value of the chosen voltage range for the double AA battery pack, the errors in the further estimation for nodes powered by these batteries are minimised. In case of Mica2Dot the nominal voltage of the cr2354 battery is about 0.1 V lower than the value we used in our calculations, but we are convinced that the estimation error may be neglected.

With the values presented so far we calculated the number of public key cryptography operations the nodes and the ECC accelerator running at 8 MHz can perform with the available amount of energy. The results in Table 15 show the amount of PKC operations but without taking into account the transmission of the data. If the calculations are combined with transmission, the presented values are smaller (see Table 16). For software solutions the numbers will be only about 3 percent smaller, in the worst case. In case of our hardware ECC accelerator the number of PKC operations will drop dramatically if the calculations are followed or preceded by transmission. The drop is between 65 and 98 percent depending on the used transceiver chip, i.e., the transmission efficiency becomes very important and in the worst case reduces the amount of PKC operations in hardware by the factor twenty. This means that the for software only several percent of the energy is consumed by the transmission and for hardware implementation it is exactly on the contrary, i.e., only several percent of the energy is consumed during the calculations.

According to our estimations, with the available energy and for a duty cycle of 100 percent, the processing units of Mica2/MicaZ and Mica2Dot are able to run about 2.6 days and 3.2 days, respectively. Under the same conditions the lifetime of TelosB is about 6.5 days. During this time period it is able to generate about 1 million ECC-160 signatures. Our hardware accelerator at 8 MHz would roughly require the same energy, i.e. could run 6.5 days at considered 100 percent duty time with a double AA battery. However, in that time the hardware accelerator could generate more than 500 million ECC-163 signatures. These numbers clearly indicate that the duty cycle should be as small as possible if long lifetimes shall be reached. With respect to PKC operations that means, they should be as fast and energy efficient as possible to ensure that they do not shorten the lifetime, significantly. Table 17 and Table 18 show the effect of PKC operations on the lifetime with respect to the duty cycle caused by these operations for both software and hardware. If PKC operations are executed very seldom e.g. once in an hour a reasonable lifetime can be achieved even with software implementations

Table 15: Estimated amount of signature generation/verification operations on a Mica2/MicaZ and TelosB with the 6750 Ws of energy available by the double AA battery pack as well as on the Mica2Dot with the 4400 Ws available by the cr2354 cell battery. The hardware results were calculated assuming the amount of energy available by two AA batteries and cover only the point multiplication, i.e., without any additional software operations

Node	Crypto-system	Signature	
		Gener.	Verif.
Mica2Dot	RSA-1024	12105	310078
Mica2/MicaZ	RSA-1024	18757	480427
TelosB	RSA-1024	97867	2500000
Mica2Dot	ECC-160	161586	81542
Mica2/MicaZ	ECC-160	250371	126357
TelosB	ECC-160	1078275	543916
Hardware	ECC-163	527343750	263671875
Mica2Dot	RSA-2048	1598	68547
Mica2/MicaZ	RSA-2048	2476	106216
TelosB	RSA-2048	12904	553279
Mica2Dot	ECC-224	59791	30166
Mica2/MicaZ	ECC-224	92656	46745
TelosB	ECC-224	398701	201192
Hardware	ECC-233	250929368	125464684

of RSA. In case of the TelosB with ECC-160 it is even more than eleven years, if the node does nothing else. For the hardware the lifetime would even reach several thousand years. All these lifetime estimations are providing just a theoretical upper bound. Due to the fact that we ignored all other operations that have to be executed and since there are additional energy consumers such as the sensing device. Additionally, the figures do not consider that probably neither devices nor batteries will last such long time. Despite the hypothetical scenario, the data show in an impressive way the impact of the improved cryptographic implementations. In the envisioned applications PKC operations will probably be used in time intervals of several minutes. In that case the life time is increased from less less than two months (Mica2Dot) to several years by applying our hardware accelerators.

6 CONCLUSIONS

In this paper we have evaluated the impact of public key cryptography operations on the life time of sensor nodes. We took into account software implementations introduced in literature as well as our own hardware accelerator for elliptic curve cryptography. Our results clearly indicate that sensor nodes can run PKC operations in software, but at the cost of reduced life time if used too often. The life time depends highly on the energy needed to execute the operation and on the frequency of PKC operations. The energy needed for a single PKC operation such as a signature generation varies between about 6.3 mWs in software on the TelosB and 12.8 μWs if the hardware ECC accelerator that we introduced in this article is used. Thus, with the available energy the number of ECC signatures that can be generated reaches from nearly 1.1 millions in

Table 16: Estimated amount of signature generation/verification operations followed/preceded by the transmission on a Mica2/MicaZ and TelosB as well as on the Mica2Dot. The hardware results were calculated for both, cc1000 and cc2420 transceivers. The sending power is 0 dBm. All results are compared to those from Table 15 and the reduction is presented in percent

Node	Crypto-system	Sig. Gen.	reduce by [%]	Sig. Verif.	reduce by [%]
Mica2Dot	RSA-1024	12061	-0.36	294157	-5.13
MicaZ	RSA-1024	18746	-0.06	472656	-1.62
TelosB	RSA-1024	97564	-0.31	2302968	-7.88
Mica2Dot	ECC-160	159173	-1.49	81181	-0.44
MicaZ	ECC-160	249751	-0.25	126186	-0.14
TelosB	ECC-160	1066873	-1.06	540766	-0.58
Hardware (CC1000)	ECC-163	15859962	-96.99	25414157	-90.36
Hardware (CC2420)	ECC-163	84692597	-83.94	68947906	-73.85
Mica2Dot	RSA-2048	15966	-0.10	66942	-2.34
MicaZ	RSA-2048	2476	-0.02	105449	-0.72
TelosB	RSA-2048	12893	-0.08	533091	-3.65
Mica2Dot	ECC-224	59325	-0.78	30097	-0.23
MicaZ	ECC-224	92537	-0.13	46712	-0.07
TelosB	ECC-224	396509	-0.55	200587	-0.30
Hardware (CC1000)	ECC-233	11158869	-95.55	17316573	-86.20
Hardware (CC2420)	ECC-233	56016598	-77.68	43548387	-65.29

Table 17: Estimated duty cycle and lifetime for RSA-1024 signature generation on a Mica2/MicaZ and TelosB with the 6750 Ws of energy available by the double AA battery pack as well as on the Mica2Dot with the 4400 Ws available by the cr2354 cell battery

RSA-1024 signature generation				
Node	duty cycle [%]	lifetime [days]	duty cycle [%]	lifetime [days]
	every 30s		every 60s	
Mica2Dot	73.43	4.2	36.72	8.4
Mica2/MicaZ	40.13	6.5	20.07	13.0
TelosB	18.87	34.5	9.43	69.0
	every 600s		every 3600s	
Mica2Dot	3.67	42.0	0.61	504.6
Mica2/MicaZ	2.01	65.0	0.33	778.7
TelosB	0.94	345.0	0.16	4140.9

software on a TelosB to over 527 millions in hardware. The life span of the most power efficient microcontroller, i.e., the TelosB node reaches from less than three weeks to more than ten years, depending on the frequency of PKC operations. If dedicated hardware is used the life span can theoretically even reach thousands of years. Theoretically, because our estimations cover the power consumed by the calculations of PKC only. Additional factors such as power consumption in sleep mode and other operations were not considered, not to mention physical limits of the hardware and especially of the batteries that render such huge time frames hypothetical.

However, our investigations clearly show that the use of specialised hardware causes the influence of PKC operations on the life time to be negligible. This is of high importance if we consider systems with a long life time, i.e. decades in case of applications like health monitoring of buildings such as bridges.

Table 18: Estimated duty cycle and theoretical life time for ECC-160 signature generation on a Mica2/MicaZ and TelosB with the 6750 Ws of energy available by the double AA battery pack as well as on the Mica2Dot with the 4400 Ws available by the cr2354 cell battery. The hardware results were calculated assuming the amount of energy available by two AA batteries and cover only the point multiplication, i.e., without any additional software operations

Signature generation ECC-160 for software, ECC-163 for hardware				
Node	duty cycle [%]	lifetime [days]	duty cycle [%]	lifetime [days]
	every 5s		every 30s	
Mica2Dot	33.0000	9.4	5.5000	56.1
Mica2/MicaZ	17.8000	14.6	2.9700	87.8
TelosB	10.4000	62.6	1.7300	375.6
Hardware	0.0220	30094.4	0.0036	183910.1
	every 300s		every 600s	
Mica2Dot	0.5500	499.0	0.2800	1122.7
Mica2/MicaZ	0.3000	877.8	0.1500	1755.6
TelosB	0.1700	3756.0	0.1000	7512.0
Hardware	0.0004	1839101.0	0.0002	3678202.0

Since sensors are thought to be deployed in huge numbers, their costs are an important factor. Especially, additional hardware to execute PKC operations, adds to the bill of material. Our ECC accelerator for B-163 requires only 1.02 mm^2 in a 0.25μm technology, which means that its price is less than 0.1 USD. If it would be manufactured in a 0.13 μm technology, its size would be reduced to only approximately 0.25 mm^2. So its price and its energy consumption would be reduced further.

To summarise, well designed hardware accelerators are well suited to reduce the energy consumption of PKC operations. They can be even used by devices from the subsensor node class such as e-grains or by the smart dust in the future. In addition, sophisticated hardware design allows to reduce the area of hardware accelerators significantly, thus, their cost will no longer be an issue, even for extremely small and cheap devices.

ACKNOWLEDGMENT

The work described in this paper is based on results of IST FP6 STREP UbiSec&Sens. UbiSec&Sens receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

REFERENCES

ATMEL Corporation (2006) 'ATmega128(L) - 8-bit AVR Microcontroller with 128K Bytes

In-System Programmable Flash' Available at: http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf.

Bazizi, A. (2003) 'Security-related Memory Costs in Security Schemes for Sensor Networks', available online at http://www.cs.vu.nl/ chandag/sensors/

Camtepe, S.A. and Yener, B. (2004) 'Combinatorial design of key distribution mechanisms for wireless sensor networks'. *In Proceedings of 9th European Symposium On Research in Computer Security (ESORICS 04).*

Chan H. *et al.* (2003) 'Random key predistribution schemes for sensor networks'. *In IEEE Symposium on Research in Security and Privacy.*

CrossBow Technology Inc. (2005) 'MPR/MIB Users Manual' Available at: http://www.xbow.com/Support/Support_pdf_files/MPRMIB Series_Users_Manual.pdf.

CrossBow Technology Inc. (2006) 'TelosB Mote Platform Datasheet' Available at: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

Du, W. *et al.* (2003) 'A pairwise key pre-distribution scheme for wireless sensor networks'. *In Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS).*

Dyka, Z. and Langendoerfer, P. (2005) 'Area Efficient Hardware Implementation of Elliptic Curve Cryptography by Iteratively Applying Karatsuba's Method' *DATE '05.*

Elliptic semiconductor (2006) 'CLP-22 Elliptic Curve Point Multiplier Core' Available from http://www.ellipticsemi.com/CLP-22_60102.pdf.

Eschenauer, L. and Gligor, V.D. (2002) 'A key-management scheme for distributed sensor networks'. *In Proceedings of the 9th ACM Conference on Computer and Communications Security.*

Gaubatz, G. *et al.* (2004) 'Public keys cryptography in sensor networks revisited'. *In The Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS).*

Gupta, V. *et al.* (2005) 'Sizzle: A Standards-Based End-to-End Security Architecture for the Embedded Internet, *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*

Gura, N. *et al.* (2002) 'An end-to-end systems approach to elliptic curve cryptography'. *In CHES 02: Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems.*

Gura, N. *et al.* (2004) 'Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs'.

Hankerson, D. *et al.* (2005) 'Software Implementation of Elliptic Curve Cryptography over Binary Fields', *CHES '00: Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems.*

Itoh, T. and Tsujii, S (1988) 'A fast algorithm for computing multiplicative inverses in $GF(2^m)$ using normal bases' *Inf. Comput.*, 78(3):171177

Lattin W. (2006) 'Efficient and authenticated key agreement: Meeting new government security requirements' *Military EMBEDDED SYSTEMS.*

López, D. and Dahab, R, (1998) 'Improved Algorithms for Elliptic Curve Arithmetic in $GF(2^m)$', *Selected Areas in Cryptography.*

Liu, D. and Ning, P. (2003). 'Establishing pairwise keys in distributed sensor networks'. *In Proceedings of 10th ACM Conference on Computer and Communications Security (CCS03).*

Malan, D.J. *et al.* (2001) 'A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography'. *In The First IEEE International Conference on Sensor and Ad Hoc Communications and Networks.*

Montgomery, P. L. (1987), 'Speeding the Pollard and Elliptic Curve Methods of Factorization', *Mathematics of Computation*, volume 48.

Moteiv Corporation (2006) 'Tmote sky - ultra low power IEEE 802.15.4 compliant wireless sensor module' Available at: http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf.

Ning, P. *et al.* (2005) 'An Efficient Scheme for Authenticating Public Keys in Sensor Networks', *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing.*

NIST/FIPS U.S. Department of Commerce (2000) 'Digital Signature Standard (DSS), FIPS PUB 186-2', Available from http://csrc.nist.gov/

Panasonic Industrial Europe GmbH (2006) 'CR2354 Lithium Battery data sheet', Available at: http://www.panasonic-industrial.com/2464.pdf.

Perrig, A. *et al.* (2000) 'Effcient authentication and signing of multicast streams over lossy channels', *IEEE Symposium on Security and Privacy.*

Perrig, A. *et al.* (2001) 'SPINS: Security protocols for sensor networks'. *In Proceedings of Seventh Annual International Conference on Mobile Computing and Networks.*

Orlando G. and Paar C. (2000) 'A High Performance Reconfigurable Elliptic Curve Processor for $GF(2^m)$', *CHES '00: Proceedings of the Second International*

Workshop on Cryptographic Hardware and Embedded Systems.

Saqib, N.A. *et al.* (2004) 'A Parallel Architecture for Fast Computation of Elliptic Curve Scalar Multiplication over $GF(2^m)$' *IPDPS.*

Satoh, A. and Takano, K. (2003) 'A Scalable Dual-Field Elliptic Curve Cryptographic Processor', *IEEE Trans. Comput.*, 52(4):449460.

Synopsys Inc. (2005a) 'Design Compiler', Available at: http://www.synopsys.com/products/logic/design_compiler.html.

Synopsys Inc. (2005b) 'PrimePower: Full-Chip Dynamic Power Analysis for Multimillion-Gate Designs', Available at: http://www.synopsys.com/products/power/primepower_ds.pdf.

Texas Instruments Inc. (2005) 'MSP430 Family of Ultra-lowpower 16-bit RISC Processors' Available at: http://www-s.ti.com/sc/ds/msp430f1611.pdf.

Texas Instruments Inc. (2006a) 'Single-Chip Very Low Power RF Transceiver' Available at: http://www-s.ti.com/sc/ds/cc1000.pdf.

Texas Instruments Inc. (2006b) 'Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee(TM) Ready RF Transceiver' Available at: http://www-s.ti.com/sc/ds/cc2420.pdf.

Wander, A. S. *et al.* (2005), 'Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks' *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications.*

Watro, R. *et al.* (2004) 'TinyPK: securing sensor networks with public key technology', *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor.*

Wang, H. (2006) 'Elliptic Curve Cryptography Based Access Control in Sensor Networks', accepted for publication in *International Journal of Sensor Networks (IJSNET*)

Watro, R. *et al.* (2004) 'Tinypk: securing sensor networks with public key technology' *In SASN 04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*

Wolkerstorfer, J. (2005) 'Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags?' Workshop on RFID and Light-Weight Crypto.

Zhang, Y. *et al.* (2005) 'Securing Sensor Networks with Location-Based Keys', *IEEE Wireless Communications and Networking Conference (WCNC 2005).*

Zhu, S. *et al.* (2003) 'LEAP: Efficient security mechanisms for large-scale distributed sensor networks'. *In Proceedings of 10th ACM Conference on Computer and Communications Security (CCS03).*