Energy-Harvested Lightweight Cryptosystems

Deepak H. Mane

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Patrick Schaumont, Chair
Leyla Nazhandali
Dong S. Ha

April 29, 2014
Blacksburg, Virginia

Keywords: Public Key Cryptography, Elliptic Curves, RFID, Wireless Sensor Node, Energy Harvesting, Throughput, Digital signatures.

Copyright 2014, Deepak H. Mane

Energy-Harvested Lightweight Cryptosystems

Deepak H. Mane

ABSTRACT

The Internet of Things will include many resource-constrained lightweight wireless sensing devices, hungry for energy, bandwidth and compute cycles. The sheer amount of devices involved will require new solutions to handle issues such as identification and power provisioning. First, to simplify identity management, device identification is moving from symmetric-key solutions to public-key solutions. Second, to avoid the endless swapping of batteries, passively-powered energy harvesting solutions are preferred. In this contribution, we analyze some of the feasible solutions from this challenging design space. We have built an autonomous, energy-harvesting sensor node which includes a micro-controller, RF-unit, and energy harvester. We use it to analyze the computation and communication energy requirements for Elliptic Curve Digital Signature Algorithm (ECDSA) with different security levels.

The implementation of Elliptic Curve Cryptography (ECC) on small microcontrollers is challenging. Most of the earlier literature has considered optimizing the performance of ECC (with respect to cycle count and software footprint) on a given architecture. This thesis addresses a different aspect of the resource-constrained ECC implementation wherein the most suitable architecture parameters are identified for any given application profile. At the high level, an application profile for an ECC-based lightweight device, such as wireless sensor node or RFID tag, is defined by the required security level, signature generation latency and the available energy/power budget. The target architecture parameters of interest include core-voltage, core-frequency, and/or the need for hardware acceleration. We present a methodology to derive and optimize the architecture parameters starting from the application requirements. We demonstrate our methodology on a MSP430F5438A microcontroller, and present the energy/architecture design space for 80-bit and 128-bit security-levels, for prime field curves `secp160r1` and `nistp256`. Our results show that energy cost per authen-

tication is minimized if a microcontroller is operated at the maximum possible frequency. This is because the energy consumed by leakage (i.e., static power dissipation) becomes proportionally less important as the runtime of the application decreases. Hence, in a given energy harvesting method, it is always better to wait as long as possible before initiating ECC computations which are completed at the highest frequency when sufficient energy is available.

This work was supported in part by the NIST Grant : 60NANB10D004 and by the National Science Foundation Grant : 1314598.

Acknowledgments

First, I would like to sincerely thank my adviser, Dr. Patrick Schaumont, for his guidance and support during my graduate research. It has been a privilege working under his guidance and I am extremely grateful for his faith in me as a student. His work ethics, dedication, punctuality, enthusiasm and extensive knowledge has always been an inspiration for me and I believe it will definitely benefit me in my future career and life. I would like to thank Dr. Dong S. Ha and Dr. Leyla Nazhandali for serving on my thesis committee.

This section will not be complete without mentioning my family, who have always supported and encouraged me throughout my life. Their love has always been the strongest source of motivation for me and I simply wouldn't be where I am today without them. Thank you Aai and Pappa for dedicating your lives to create a bright future for your children. My sisters Suvarna and Supriya, deserve a special note of thanks for being my best friends all through my life.

I would also like to thank my beloved friends, who have always supported and cared for me in my good as well as difficult times. Thank you everyone - Pravin, Chinmay, Rohini, Snehal, Mahesh, Dhananjay, Sushrut, Pranil, Praneet, Mrunmayee, Aniket, Anup, Yogesh. Special thanks to Dhiraj, Sarvesh, Vireshwar, Sriram, Krishna who have helped make life here at Blacksburg merrier.

I sincerely appreciate the help from my lab mates, Krishna, Nahid, Aydin, Moen and Mostafa. Without help from Krishna, my research would not have been so smooth.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Lightweight devices including sensor nodes, RFIDs, smart cards etc., are now widely used in many applications for device identification, secure communication, and also to store private information. Sensor nodes and RFIDs that are constrained in terms of the available resources, monitor their surroundings and provide a real-time information regarding a physical phenomenon. The Internet-of-Things, which may turn every such device into an Internet host, is an important opportunity for these lightweight devices. All these devices may become interconnected to provide a better user experience. This trend is now accelerated by advanced technologies such as IPv6, ultra-low-power microprocessors, and novel MEMS sensors. When several devices are interconnected, it is highly probable that malicious/adversarial devices can potentially access sensitive information, thereby compromising the security of the network. Thus, information security plays a crucial role in bringing Internet-of-Things to reality. It is easy to see that cryptographic algorithms are therefore necessary to ensure data secrecy.

There are different types of cryptographic algorithms that address several security and privacy issues in such an interconnected world. These algorithms support services like authentication, confidentiality, non-repudiation, and data integrity. Cryptographic algorithms are classified into two broad categories : Symmetric cryptography and Public key cryptography (PKC). In the context of IoT, PKC is a better choice in comparison to symmetric-key cryptography because of easier key distribution and key handling. Elliptic Curve Cryptography (ECC) is one such public-key technique that has received significant attention due to the high level of security while using smaller key sizes as compared to other PKC techniques such as RSA.

Cryptographic technologies are advancing with novel designs and implementations and there is an extensive body of work on efficient hardware and software implementation of ECC-based [1, 2, 3] signatures. However, the implementation of PKC in constrained environments remains a challenging problem. Lightweight Cryptography(LWC) is one of the research areas that focuses on designing schemes for devices with constrained capabilities in terms of power supply, hardware, software, and connectivity. While device size and energy consumption are the major factors with regards to the hardware implementation; smaller memory footprint, RAM size, and computationally light algorithms are the core focus of the software implementation.

The next challenge with lightweight platforms is the power source. The amount available energy in such constrained environments is limited, and it requires *a holistic approach that considers the energy source as well as the energy consumer.* Several researchers have analyzed the energy cost of public-key cryptography [4, 5, 6] and symmetric-key cryptography [7]. Lightweight cryptosystems warrant optimization techniques that match the available energy budget and performance requirements. This requires selection of optimal architecture parameters such as operating voltage, frequency, and use of efficient cryptographic algorithms. Most of the earlier literature considered optimizing the performance of ECC (with respect to cycle count and software footprint) on a given architecture. These designs use advanced algorithmic transformations and optimizations to accelerate the underlying complex mathematical operations. Furthermore, they also make use of technology-specific features. Software implementations assume certain amounts of flash and RAM memory, or microcontroller features such as a hardware multiplier. Hardware implementations assume a target cell library of specific performance and feature size. Such an assumption of a specific target architecture at the start of the design is typical for contemporary digital design methods. On the other hand, it is much harder for ECC designers to make clear commitments to application constraints such as the available energy budget and the required authentication latency.

In this thesis, we present a different approach to design a lightweight cryptosystem by considering the constrained design requirements, rather than how to obtain the fastest ECC point multiplication. The question addressed in this work is: how can we select architecture parameters such that we meet these design requirements, including energy budget and application throughput? We provide an empirical answer to this question by exploring the energy design-space of PKC in resource constrained environments.

Another challenge of lightweight devices is the battery powered operation, limiting the lifetime of the devices and it requires battery replacement over the time. One possible solution to battery powered devices is to integrate an energy harvester with the device itself which harvests energy from its surroundings [8] in order to power up the system. Energy harvesting considerably simplifies the installation and maintenance of such devices. Without battery replacement or wiring requirements, they can be installed in physically challenging or inaccessible environments - and their lifetime appears to become infinite. The downside of energy harvesting is that it severely limits the energy budget available for WSN operation [9]. For example, vibration-based [10] or piezo-electric based harvesters [11] deliver a few microwatt up to a milliwatt; solar-based harvesters deliver a few tens to hundreds of milliwatt [12].

This thesis looks at a specific type of Wireless Sensor Node (WSN), one which is in capability just above a passive RFID. Figure 1.1 demonstrates the topology of an energy-harvested WSN. A supercapacitor collects the energy from a harvester. The supercap then powers up a microcontroller and a radio. However, this system needs to balance the influx of energy from the harvester with the energy consumed in computing and communicating.

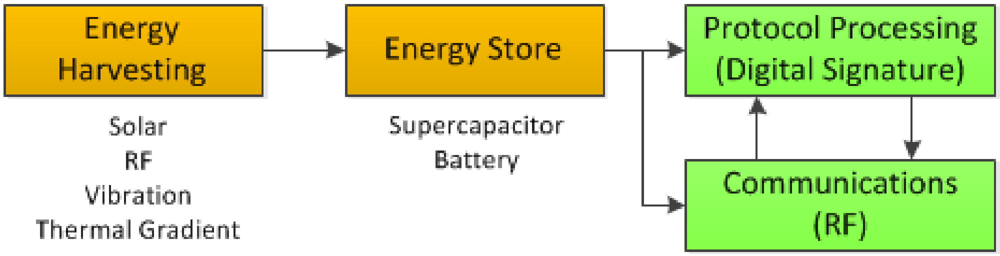

Figure 1.1: Energy harvested Embedded System

The WSN will therefore operate with a certain duty cycle that periodically activates the communication/computation subsystem, and that otherwise powers it off or keeps it in a low-power standby mode.

One of the contributions of this thesis is to understand the energy design-space of PKC, covering computation as well as the communication overhead. This was previously investigated by de Meulenaer for ECDSA [13], and by Wander for ECDSA and RSA [14]. Our efforts differ from these previous work in the following aspects: (a) we present actual measurement data rather than estimates and (b) we investigate the impact of security level on the energy budget. We also note that the importance of the energy design space of PKC was also raised by Struik at DIAC 2013 [15]. We analyze the public-key cryptographic (PKC) primitives on an energy harvested node. In a public-key identification protocol, a verifier sends a random challenge to the WSN and requests a signature for it. Afterward, the verifier checks the signature using the WSN public key. The WSN releases its public-key to the verifier while protecting its secret key.

We study the energy/latency characteristics of an WSN doing ECDSA key generation, signature generation, signature verification and signature transmission over the RF. We present our results for a microcontroller target, a MSP430F5438A from Texas Instruments and evaluate the energy characteristics of signatures at 80-bit and 128-bit security level [16]. The resulting curves then allow us to determine, for a given security level and energy budget, the most appropriate core frequency and voltage level.

1.2 Contribution

This effort brings following contributions to the challenging design space of energy-harvested WSN.

1. We demonstrate a WSN platform that integrates a microcontroller, a radio, an energy-harvester, and an energy-measurement subsystem. We can accurately measure the performance as well as the energy consumption of individual components in this system. The WSN connects to a host workstation that takes the role of server. For example, when implementing an identification protocol, the host workstation acts as the verifier. Our design is based on Commercial Off-The-Shelf (COTS) components leading to a physical proof-of-concept.

2. We introduce a low cost prototype setup for the precise energy measurement of a microcontroller based application.

3. Our prototype implements an efficient authentication between a resource constrained node and a server, using Elliptic Curve Digital Signature Algorithm (ECDSA). We explore the energy/architecture design space for 80 bit and 128 bit security-levels, for prime field curves `secp160r1` and `nistp256`.

4. We examine multiple architecture configurations: multiple frequencies, multiple core voltages, and with/without use of the MSP430's hardware multiplier. For each of these configurations, we perform an in-depth analysis of the energy needs by isolating the energy required for computations from the energy required for communication.

5. Finally we present a methodology to derive and optimize the architecture parameters starting from the application requirements. We introduce an energy model that helps to configure a sensor node automatically so as to adapt to the changing energy needs and thereby guides the design of energy harvester.

1.3 Organization

The following chapters of the thesis are structured as follows. Chapter 2 describes the background related to power and energy in the digital electronics, lightweight platforms and public key cryptography. In chapter 3, we explain the hardware and software setup used for our experiments. We also describe the various operating modes supported by our system. The resulting energy/throughput curves are presented in Chapter 4 and applied in a methodology in Chapter 5. Chapter 6 describes our energy model for sensor node configuration. Finally, Chapter 7 summarizes the contributions of our work and identifies potential future targets.

1.4 Related Articles

Our work is described in the following papers:

- D. Mane, P. Schaumont, ”Energy-Architecture Tuning for ECC-based RFID tags”, RFIDSec’13.
- D. Mane, K. Pabbuleti, P. Schaumont, ”Energy Budget Analysis for Signature Protocols on a Self-powered Wireless Sensor Node”, under review, RFIDSec’14.

Chapter 2

Background

In this chapter we give a brief introduction to power in digital electronics, RFID, WSN, and Public key cryptography.

2.1 Power and Energy in Digital Electronics

Power dissipation in modern digital electronics has two major components: static power dissipation defined by the static leakage current, and dynamic power consumption, defined by the circuit activity. The static power dissipation depends, in first order, on the operating voltage of the circuit and the size of the circuit. On the other hand, dynamic power dissipation depends, in first order, on the operating frequency of the circuit, the size of the circuit, and the square of the operating voltage.

We analyze what happens to the energy dissipation for a fixed workload, such as signature generation, under varying operating conditions. In the following, K and C are technology constants, α is an application-dependent activity factor, T_{cycle} is the clock cycle period, f_{cycle} is the operating frequency, and n is the workload cycle budget. The energy dissipation per workload has a static and a dynamic component which are given by,

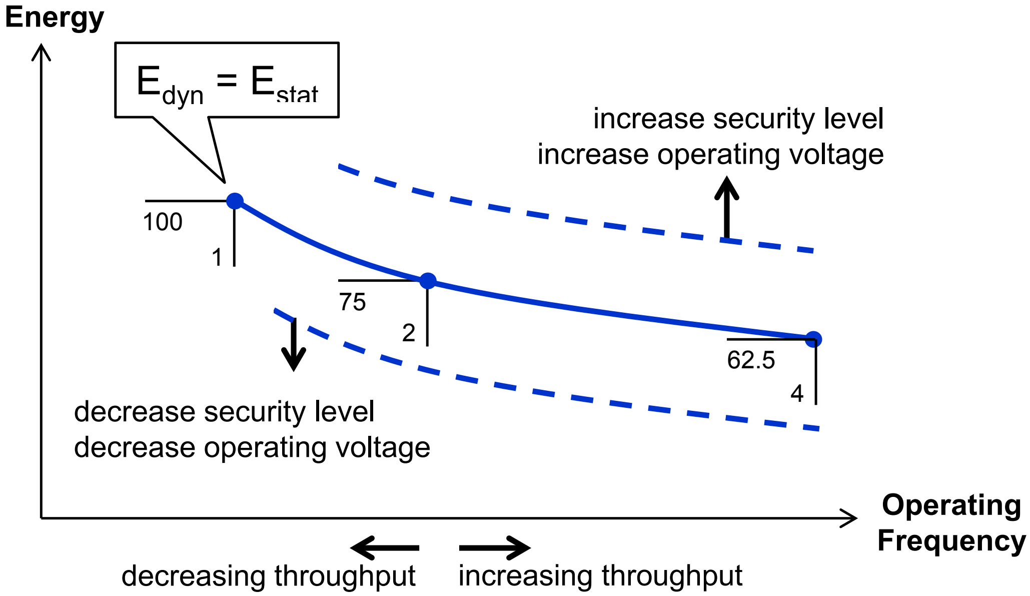

Figure 2.1: Expected Energy dissipation as a function of frequency

$$E_{dyn} = P_{dyn}.T_{alg} = \alpha.C.V^2.f_{cycle}.T_{cycle}.n \tag{2.1}$$

$$E_{stat} = P_{static}.T_{alg} = K.V.T_{cycle}.n \tag{2.2}$$

The total energy dissipation thus equals

$$E_{tot} = E_{dyn} + E_{static} = n.[\alpha.C.V^2 + K.V.T_{cycle}] \tag{2.3}$$

This formulation leads to the following assessment. If the clock frequency increases, the total energy per workload will decrease: the dynamic energy remains constant, while the static energy decreases. Furthermore, if the operating voltage decreases, the total energy per workload will decrease as well. Finally, if the security level of the design increases (from 80 bit to 128 bit, for example), the cycle budget n will increase, and the total energy per workload will increase as well. This analysis is captured by Figure 2.1. The leftmost point on the curve represents a design where the static and dynamic parts of the energy per workload are equal. As the operating frequency increases, the total energy decreases as well. We note that this figure is a theoretical model: it ignores overhead for clock generation, and for transitions between power modes.

2.2 Constrained RFID, Wireless Sensor Nodes

Lightweight platforms are constrained in terms of the available computational resources, power consumption, memory size, form factor, cost etc. Such constraints make security (i.e., integrity, confidentiality and availability) implementation challenging. This requires design and implementation of optimized cryptographic algorithms that meet the resource requirements of the lightweight devices. Lightweight cryptography is an area that deals with designing such schemes efficiently.

Depending on the power source, RFID designs are either energy-constrained or power constrained [17]. They also have to optimize application throughput (the time taken to complete a single signature) with respect to the available energy budget. We note that the requirements on energy and power depend on the type of RFID.

- Active RFID are powered from a battery source, and they have to minimize the energy consumed per signature as this will maximize the battery lifetime.
- Passive RFID are powered through an RF source, and they have to minimize the time required per signature while matching the available power budget.
- Passive RFID, powered through an energy harvesting mechanism that includes an energy store, have to minimize the energy used per signature as well, since this allows uninterrupted RFID operation. Furthermore, the energy needed for the desired application throughput has to match the average energy influx in the harvester.

Along similar lines, different nodes interact with each other in a wireless sensor network and it becomes necessary to secure their communication against eavesdropping or malicious manipulation. WSN node includes a computational unit, radio transceiver with antenna and an energy source. Size and cost of such sensor node is an important factor for their wide deployment in the real world. Further, size and cost result in corresponding constraints on the resources like computational power, memory, energy and communication bandwidth. It is necessary to optimize the energy consumption of WSN nodes as it determines their lifetime. WSN nodes can be deployed in various environments that warrants the following characteristics; a) maximum lifetime b) self-configuration, and c) robustness.

2.3 Public Key Cryptography

Table 2.1: Recent ECC implementations for RFID

Ref	Field/Curve	Target	Time	Operation	Resource
		Hardware or Software	(seconds)		
[18]	$GF(2^{163})$	HW, 0.13μm 100Khz	0.244	Point Mult	12,506 GE
[19]	$GF(2^{163})$	HW, 0.18μm 106Khz	0.279	Point Mult	11,904 GE
[4]	`secp160r1`	SW, MSP430F1611 8MHz	8.54	ECDSA sign	13,520 bytes code
[1]	`secp160r1`	SW, MSP430F1611 8MHz	1.1	Point Mult	NA
[2]	`nistp192`	SW, MSP430F2131 6.7MHz	1.6	Point Mult	16,060 bytes code
[20]	`secp160r1`	SW, MSP430F5529 25MHz	0.068	ECDSA sign	24,000 bytes code

There are appealing advantages to use public-key cryptography in lightweight applications like RFID, and WSN that require authentication. Indeed, PKC-based authentication significantly simplifies the distribution of cryptographic keys, resulting in a more scalable solution. Elliptic curve cryptography(ECC) is one such popular PKC technique in embedded context because of the high security level while using small key sizes. The challenge of using ECC [21] in such constrained environment is how to deal with the high computational cost associated with ECC algorithms relative to the capabilities of the lightweight platforms [22, 23, 24, 25]. Table 2.1 shows some of the more recent achievements. It is fair to state that a security level of 80 bit (ECC curves of at least 160 bit) is within reach, with sub-second latency, in small footprint applications (i.e. 15 KGate, 100 KHz hardware implementations; or 16 bit, 8-MHz software implementations).

Chapter 3

Implementation

This section explains the system architecture of our design which includes a brief description of the different hardware blocks, the driver application and its operational modes. We first explain our energy measurement setup and the procedure to measure the average current consumed by any software subroutine.

3.1 High Resolution Energy Measurement

Figure 3.1 depicts the block diagram of energy measurement setup used in our experiments. The average current consumed by a microcontroller during a particular interval of time is measured by integrating the immediate current. The current is measured by means of the voltage drop over a shunt resistor on the microcontroller `Vcc` line. To sample the voltage drop, we use OpenADC [26, 27] with a Spartan FPGA [28], in place of a traditional high-speed oscilloscope setup. OpenADC is a custom ADC board with a 10-bit A/D converter that supports differential inputs and an adjustable reference voltage. The clock frequency of the A/D converter can be independently chosen of the microcontroller clock. The FPGA takes care of the sample accumulation and sample counting as shown in Figure 3.2. The

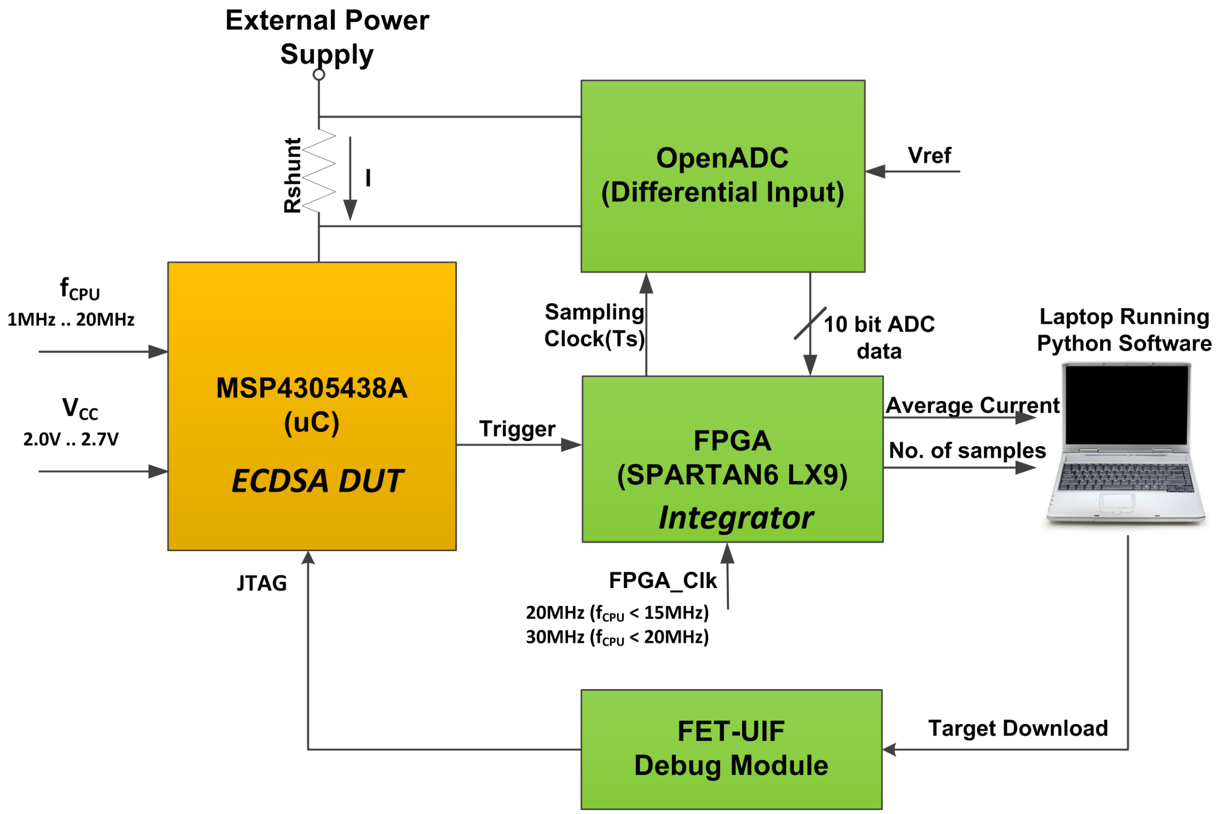

Figure 3.1: Energy Measurement Setup Block Diagram

integration period is defined by means of trigger signals created by the microcontroller.

3.1.1 Average current measurement

At a desired time, the microcontroller triggers the FPGA to start sampling OpenADC data (Figure 3.3). Once the trigger is asserted, the FPGA accumulates ADC samples until the trigger line is re-asserted. The accumulator represents the average current consumed by a function being executed on a microcontroller. It also measures the number of samples collected during trigger window to derive the execution time of that function, denoted by T_{Alg}. At the end of trigger event, accumulated value and number of samples are written to the internal FIFO which is read by a python script running on laptop (PC) over UART interface. A Python script uses this data and calculates the average energy consumed by an

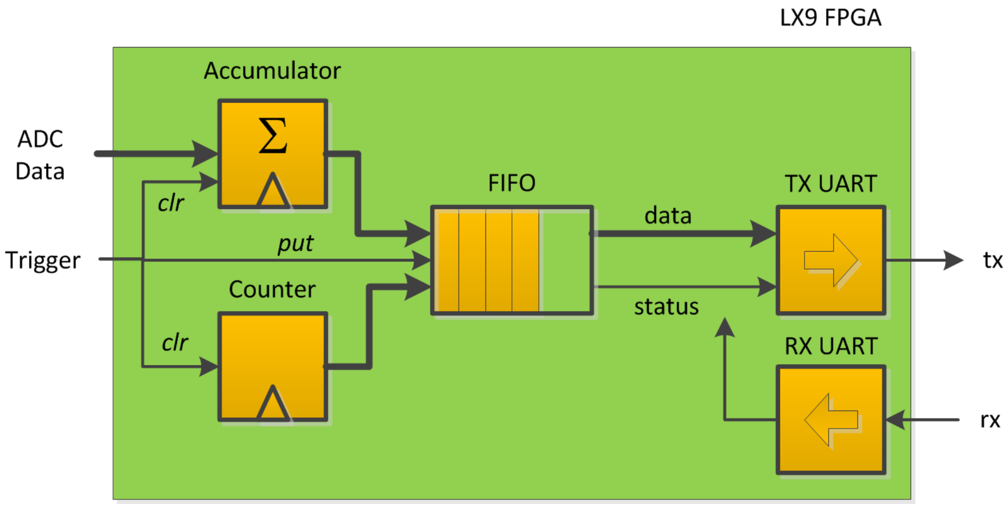

Figure 3.2: Average current integration within FPGA

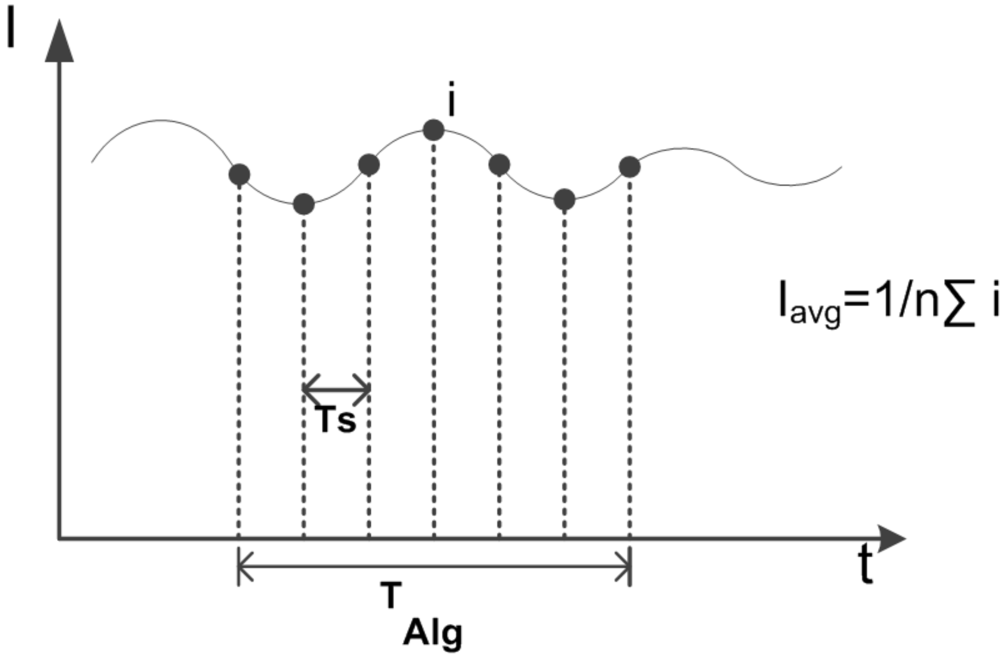

Figure 3.3: Average current measurement

MSP430 function as follows :

$$\begin{aligned}\text{Energy} &= V_{cc}.I_{avg}.T_{Alg} \\ &= V_{cc}.\frac{Accm.V_{ref}}{2^n.N_s.R}.N_s.T_s \\ &= V_{cc}.\frac{Accm.V_{ref}}{2^n.R}.T_s \qquad (3.1)\end{aligned}$$

$$\text{Cycle count} = N_s.T_s.F_{cpu} \qquad (3.2)$$

where V_{cc} is supply voltage of the microcontroller, $Accm$ is FPGA accumulator value, V_{ref} is ADC reference voltage, N_s is number of samples collected during trigger window, T_s is sampling period, 2^n is resolution of ADC where n is 10 bit, R is a shunt resistor value and F_{cpu} is microcontroller frequency. Our Energy formula assumes that the voltage supply at the microcontroller input is constant, in other words, that the voltage drop over the shunt resistor is negligible with respect to V_{cc}.

The above formula shows that the resolution of energy measurements increases with low reference voltage of ADC, high sampling frequency and with high resolution of ADC. We use sampling rate of 20MHz for operating frequencies below 15MHz and 30MHz for operating frequencies above 15MHz. In our experiments, ADC reference voltage is 0.5V and shunt resistor is taken to be 100Ω. This setup can be used to measure the energy consumption of any device provided that it has the facility to insert a resistor in series with power supply. Also, the device should be able to generate a trigger signal to activate the FPGA accumulator.

3.2 System Architecture

Figure 3.4 shows the block diagram of our experimental setup. It consists of two major blocks, a self-powered wireless sensor node and a central control unit(server). The server authenticates different sensor nodes by verifying the node's signature. The sensor nodes is powered from an integrated energy harvester. The energy measurement unit described in the previous section is used to precisely calculate the computation and communication energy.

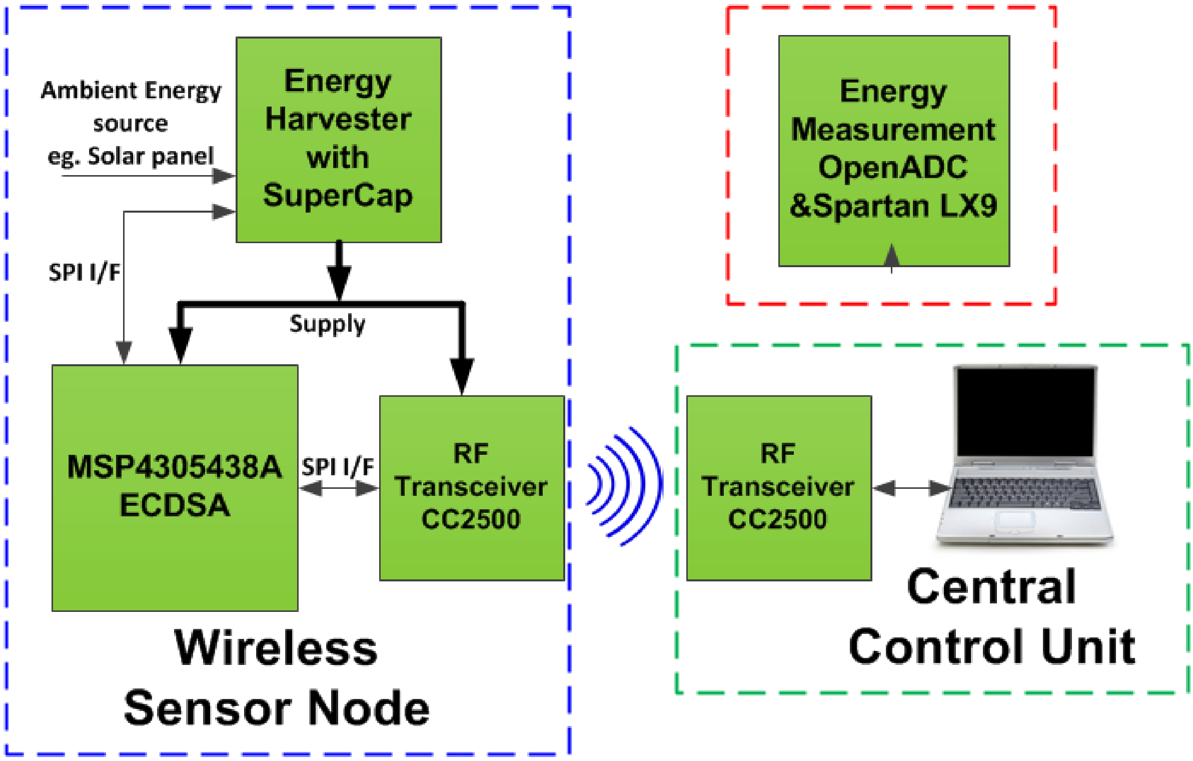

Figure 3.4: Wireless Sensor Node Block Diagram

3.2.1 Microcontroller

We use MSP430F5438A [29] as our prototyping platform. The MSP430F5438A is an ultra-low power Reduced Instruction Set Computer (RISC) from Texas Instruments, optimized for low-resource applications [30]. The architecture combines five different low power modes suitable for low power battery operation. The MSP430F5438A features a 16-bit CPU, 256KB flash, 16KB SRAM, up to 25 MHz CPU clock and 16 working registers with 12 available as general purpose registers. It also supports a 32 bit hardware multiplier.

Figure 3.6 shows the internal energy management architecture of the MSP430F5438A. In

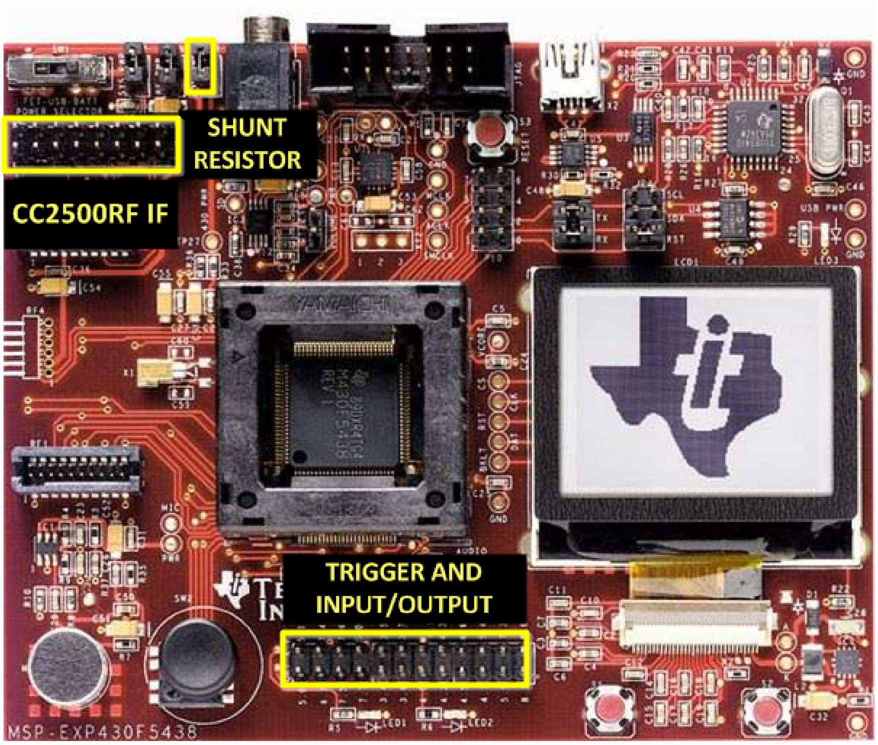

Figure 3.5: Texas Instruments Microcontroller MSP430F5438A

http://www.ti.com/tool/msp-exp430f5438, Used under fair use, 2014

general, `VCore` supplies the CPU, memories (flash and RAM), and the digital modules, while `DVcc` supplies the I/Os and all analog modules. The internal core voltage of the MSP430F5438A, `VCore`, needs to be adjusted as a function of the desired operating frequency. The `VCore` output is programmable in four steps, to provide only as much power as is needed for the speed that has been selected for the CPU. We configure `VCore` voltage by writing register bits `PMMCOREV[1:0]`. Table 3.1 shows recommended `PMMCOREV` settings and minimum external power supply voltage for different frequency ranges. We make use of these programmable `VCore` levels to optimize the energy efficiency of ECDSA on the MSP430.

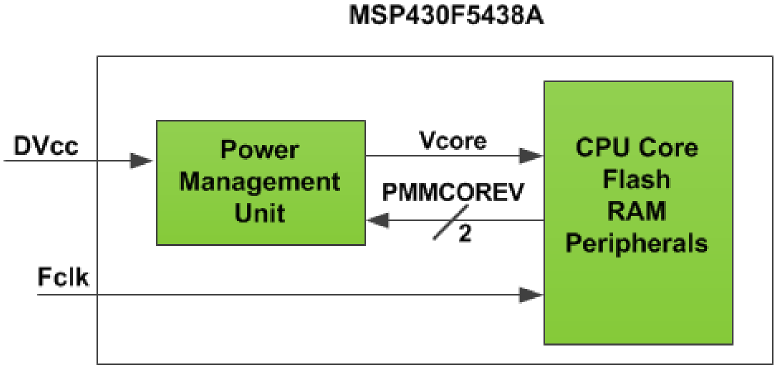

Figure 3.6: Voltage and Frequency Scaling

Table 3.1: Recommended PMMCOREV and DV_{CC} settings for Selected f_{sys}

f_{sys}(max) (MHz)	Minimum DVCC	Minimum PMMCOREV[1:0]
8	1.8V	00
12	2.0V	01
20	2.2V	10
25	2.4V	11

3.2.2 RF Transceiver

CC2500 is a RF transceiver from Texas Instruments designed for low power wireless communication [31]. It operates in 2.4GHz ISM band and supports various programmable modulation schemes and power levels. It has a SPI interface for configuration and data transfer. It can operate in polling-based and interrupt-based data transfers. It supports two low power modes, Power down mode and Wake-On-Radio mode, suitable for low power applications. In Power down mode, all the chip peripherals including radio frontend and digital circuitry are off, consuming only 2μA current. But this mode does not support interrupt based re-

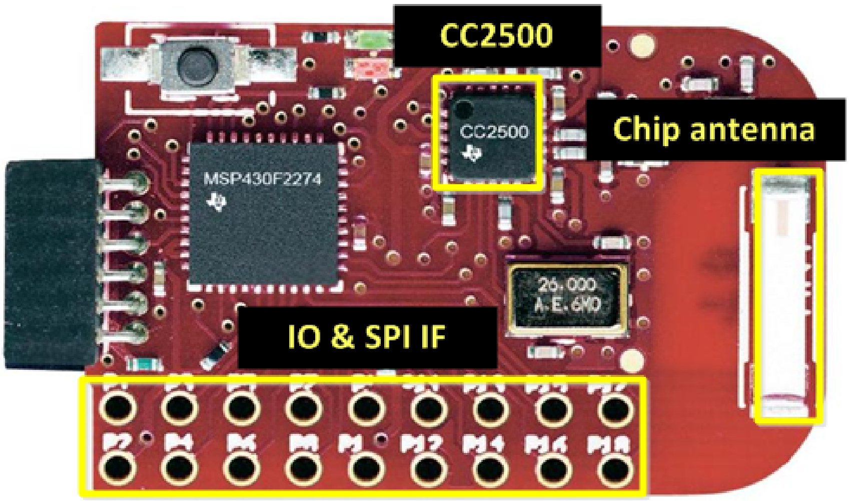

Figure 3.7: RF CC2500

Used under fair use, 2014

ception as the RF front end is turned off. In Wake-On-Radio mode, the RF receiver wakes up periodically and checks for any available packets. It automatically goes back to sleep if no packet is available. It consumes more power in this mode than in the powerdown mode and depends on the wakeup frequency and stay-awake time. We use CC2520EM [32], a RF solution optimized for low power applications.

3.2.3 Energy harvester

The sensor node has an integrated energy harvester that scavenges energy from the ambient resources and stores the energy in low leakage supercapacitor. We use AN1010, an optimized energy harvester from Anagear that supports efficient energy management [33]. It features a photovoltaic cell and has an interface to other energy sources like thermal gradient, vibration etc. It supports dual programmable output supplies and consumes very less current in the sleep mode. It implements efficient energy storage capability using autonomous charging circuitry and a supercapacitor. The size of the harvester is dependent on the target application and available ambient resources. Different register settings, level of supercapacitor voltage

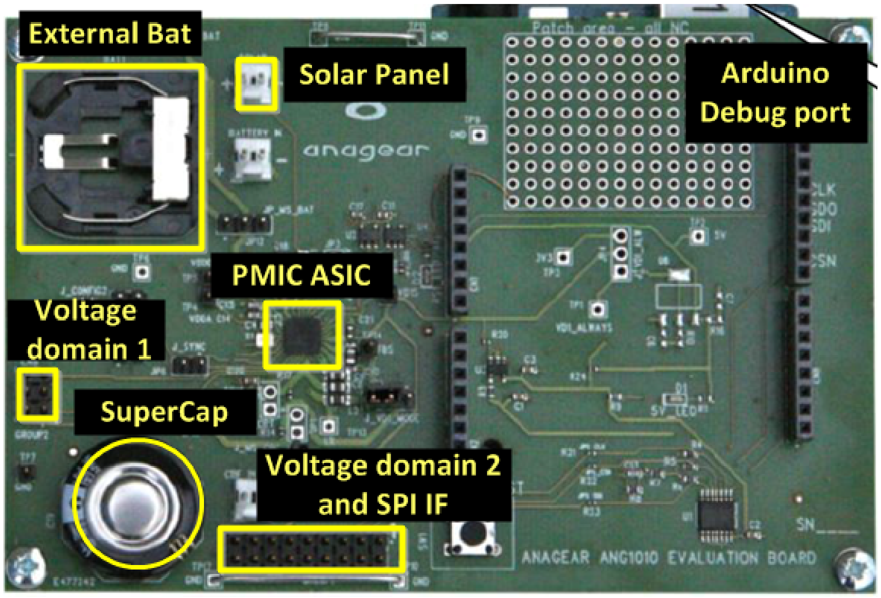

Figure 3.8: Anagear Energy Harvester

http://www.anagear.com/content/ANG1010, Used under fair use, 2014

and status of the energy harvester are checked over SPI interface.

3.2.4 Server

The WSN connects to a server PC with an integrated RF transceiver. The server receives packets from the sensor node, and implements the second half of the signature verification protocols.

3.2.5 Energy Measurement Unit

This unit precisely calculates the computation and communication energy for an authentication. These energy readings are used to efficiently implement an energy model for a given application as discussed in chapter 5.

3.3 Target Protocol: ECDSA in `secp160r1` and `nistp256`

The driving application for our measurements is ECDSA key generation, signature generation, signature verification, and signature transmission. ECDSA is a well-known signature mechanism based on Elliptic curve cryptography [21]. Rather than developing our own implementation from scratch, we use the RELIC library with support for the MSP430 and 32-bit hardware multiplier [34]. We have implemented ECDSA using two different prime-field curves, `secp160r1` and `nistp256`, which have a security level of 80 bit and 128 bit respectively. In ECDSA, signing costs one point multiplication, while verification costs two. The scalar multiplication is done with a left-to-right window-3 NAF multiplication, and using Jacobian Projective Coordinates. The field operations are basic Comba multiplication and squaring, with Montgomery reduction. SHA-1 is used for hashing and as a pseudo-random generator. We used two implementation variants for each of the curves: one which uses a 32-bit hardware multiplier (using RELIC's `msp-asm` backend), and a second one which emulates multiplication in software (using RELIC's `easy` backend).

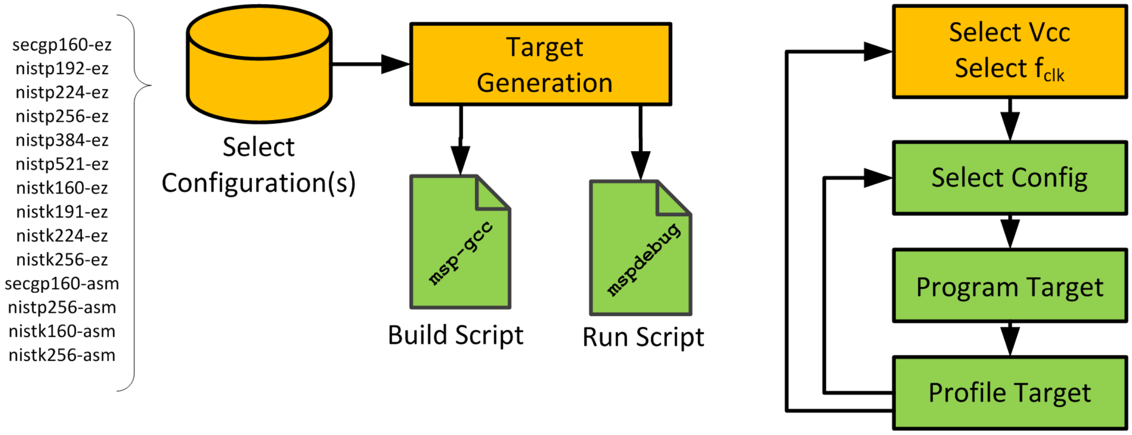

Figure 3.9: Test Software Generation

Our standard testbench goes through ECDSA key generation, ECDSA signature generation of a fixed digest, and ECDSA signature verification, signature transmission over RF. The execution time, as well as the energy, is measured for each of these steps separately. Our performance and energy numbers only cover the computations, and they don't include initialization overhead. Figure 3.9 shows the basic software flow that selects different configurations based on the different prime-field curves, supporting different security levels.

3.4 Operational Modes

The way various nodes communicate with each other may vary widely depending on the application. In our experiments, we implement the following modes of operation which are gated by the amount of available energy and then analyze the performance of WSN.

1. **Asynchronous TX/RX Mode:** In this mode ,the server initiates a communication and sends a request to WSN; the node replies back with appropriate data. The WSN waits for a service request from the server.

2. **Periodic TX Mode:** In a second mode of operation, the WSN starts the protocol by sending the sensed data to the server. The server then authenticates the node by verifying the nodes signature and takes further action.

In both the modes, sensor node first checks the amount of energy available with the harvester and accordingly takes further action. It performs signature generation and transmission if sufficient energy is available, otherwise the system enters power down mode. Energy controlled operation of a node makes it autonomous and energy efficient. It makes the sensor node deployment easy and scalable.

Chapter 4

Experimental Results

In this section, we present experimental results of our energy measurements under different architecture configurations. Each authentication consists of two parts of the energy, namely computation and communication energy. Computation energy is attributed to signature generation whereas communication energy is attributed to signature transmission over RF. We measure the energy consumption of ECDSA key generation, signature generation and signature verification for different operating voltages and frequency settings. We then measure communication energy needed to transmit a signature over the RF to the server. We also measure the signature generation and transmission time to find the throughput of the system, i.e., number of authentications performed per second. We use the `gcc 4.6.3` cross-compiler for MSP430 family of microcontrollers. First, we explain the results based on computational efforts for the signature generation and then we discuss communication overhead and total energy consumption for an authentication.

4.1 Computation results analysis

Table 4.1 shows cycle counts for different ECDSA operations, and Table 4.2 shows the footprint of the corresponding implementations. We examine different architecture with varying operating voltage, frequency and with/without use of MSP430's hardware multiplier. Following graphs show the energy/throughput characteristics for signature generation.

Table 4.1: Cycle count of ECDSA operations on the MSP430F5438A

Operation	`secp160r1`		`nistp256`	
	w/o HWM	with HWM	w/o HWM	with HWM
Key Generation	19,343,970	1,796,499	124,427,469	5,225,820
Signing	19,141,737	2,372,103	124,942,312	6,408,792
Verification	57,621,281	57,48,345	346,290,644	15,584,937

Table 4.2: Code size of the implementation of ECDSA on the MSP430F5438A

	`secp160r1`		`nistp256`	
	w/o HWM	with HWM	w/o HWM	with HWM
Flash Bytes	27,134	28,168	28,138	32,234
RAM Bytes	1,074	1,074	1,542	1,542

Figure 4.1 and 4.2 show the energy consumption for 80-bit(`secp160r1` curve) security level, without and with hardware multiplier, respectively. We analyze the effect of different operating voltage on energy consumption. In our experiments, we found that if a microcontroller is operated at 2.0V instead of 2.7V, it saves almost 1.4 times energy consumption. Reducing the operating voltage reduces the dynamic power consumption because the circuit will have smaller voltage swings during switching. Also the static power consumption reduces because the leakage current reduces. Further, the use of the hardware multiplier results in almost

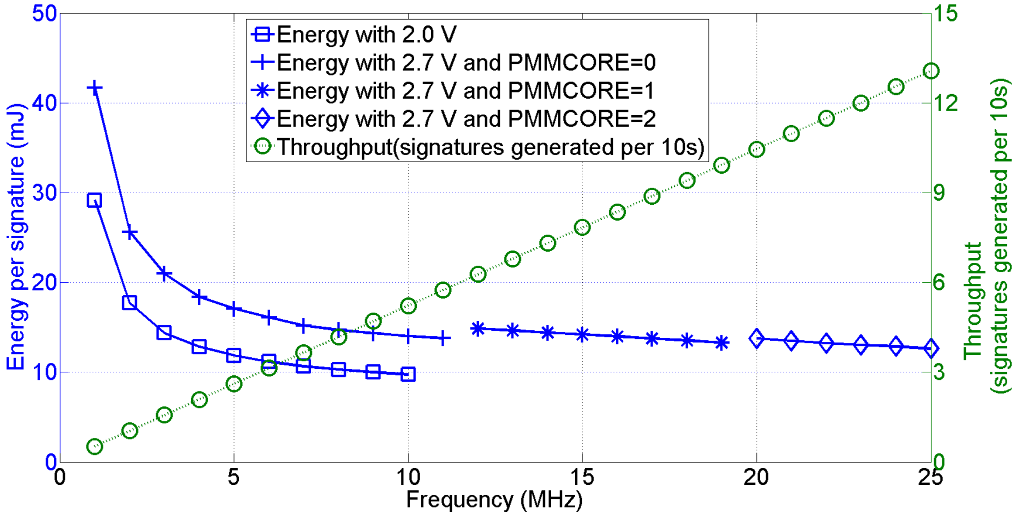

Figure 4.1: Energy consumption for `secp160r1` without hardware multiplier

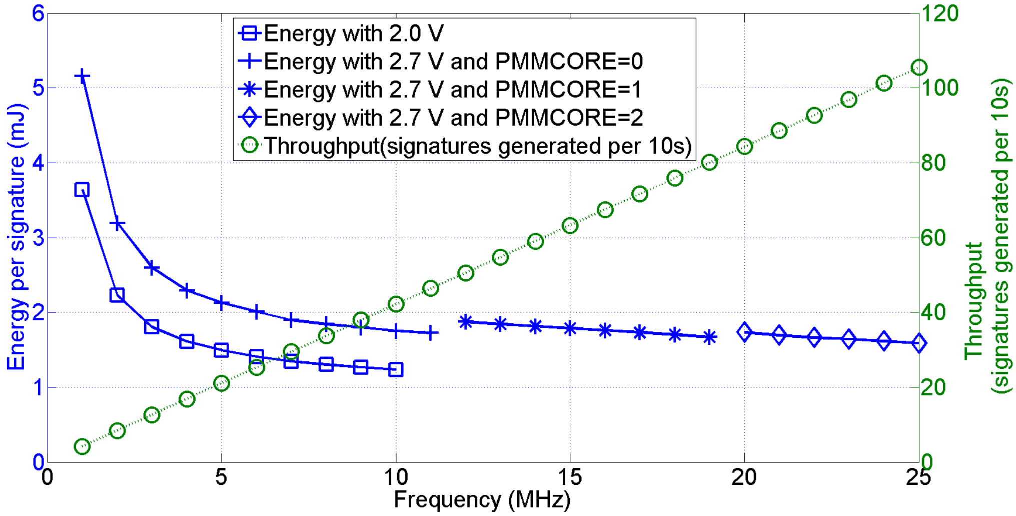

Figure 4.2: Energy consumption for `secp160r1` with hardware multiplier

8 times energy reduction. This is because the hardware multiplier accelerates the signing operation almost by 8 times.

Figure 4.3 and 4.4 show the energy consumption for 128-bit(`nistp256` curve) security level, without and with hardware multiplier respectively.

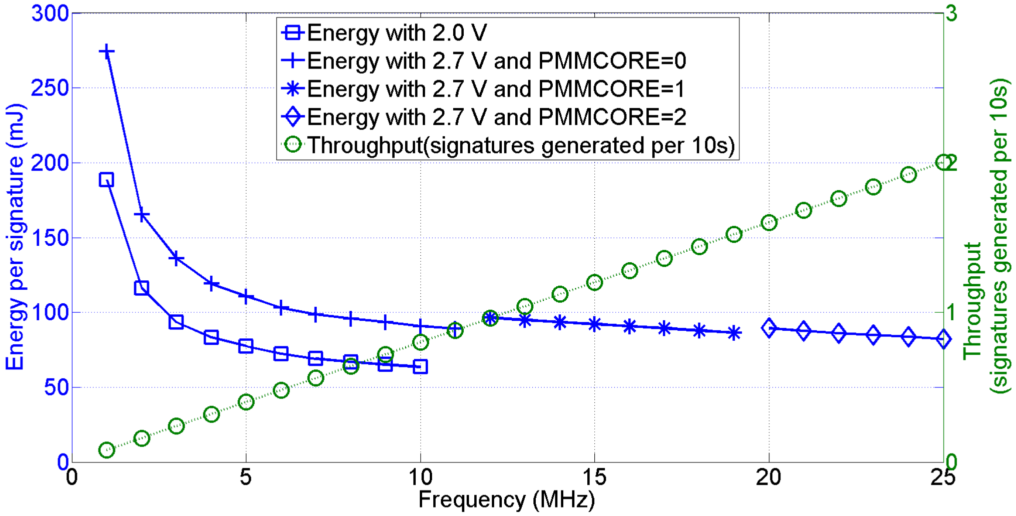

Figure 4.3: Energy consumption for `nistp256` without hardware multiplier

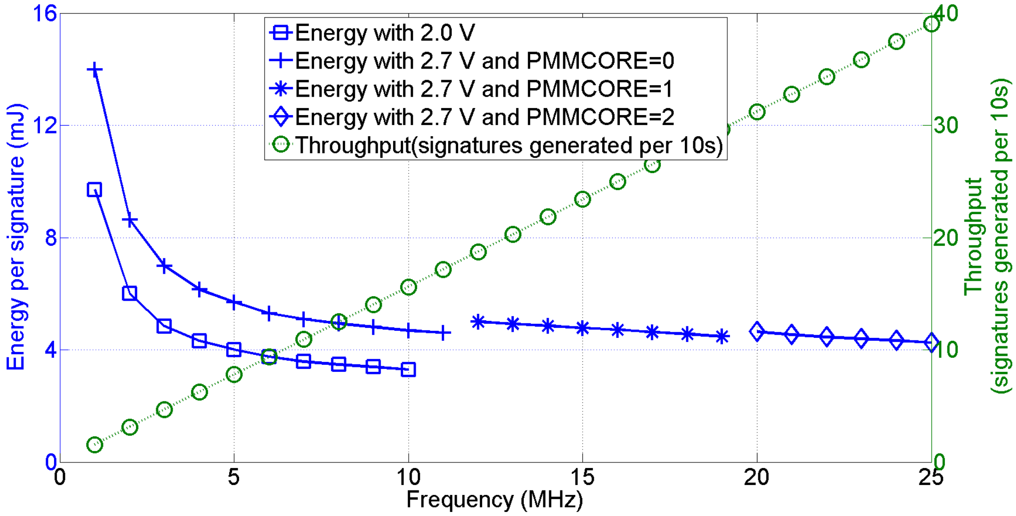

Figure 4.4: Energy consumption for `nistp256` with hardware multiplier

The curves for operation at 2.7V is discontinuous at 12MHz and 20MHz. The discontinuities are caused by reprogramming of the power management system of the CPU.

Figure 4.5 shows energy improvement factors for the different architecture configurations. It can be observed that moving towards the origin results in the most optimized design where energy consumption is minimal. Reducing the operating supply voltage from 2.7V to

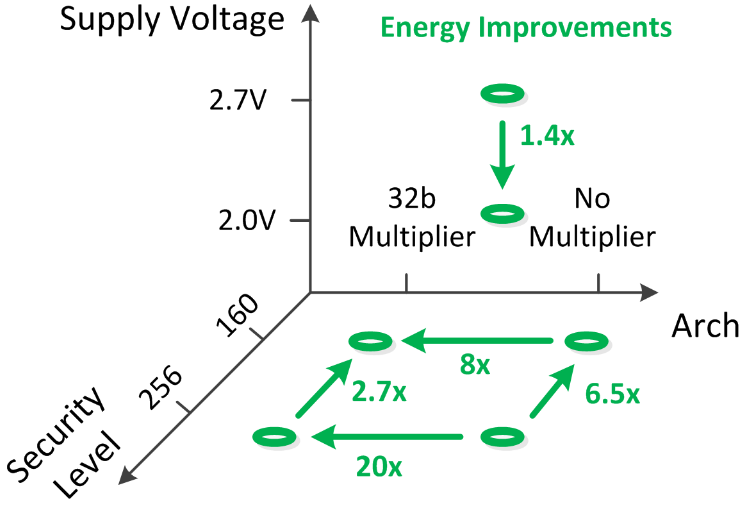

Figure 4.5: Energy Profile for ECDSA `secp160r1` and `nistp256` on TI `MSP430F5438A`

2.0V results in a gain of 1.4 for both security levels. The computational complexity of used algorithm results in different improvement factors since energy consumption is dependent upon runtime of an algorithm. The acceleration achieved with hardware multiplier is also dependent on the used security level which gives different energy gain factors. Overall the most significant impact comes from architecture specialization.

4.2 Communication results analysis

We measure the energy needed to transmit a signature over the RF to the server. This energy depends on the length of the signature; larger the signature size, more is the energy needed for transmission.

Table 4.3 shows the signature length, RF transmission packet length and corresponding transmission energy. One transmission packet consists of total 64 bytes of data. It can be noted that transmission energy is not affected by use of MSP430's inbuilt hardware multiplier, since it only accelerates the computation of a signature keeping the signature size same. As seen from the table, although the signature length is different for both security

Table 4.3: Signature, packet size and Communication energy

	`secp160r1`	`nistp256`
Signature Size(bytes)	40	64
No. of RF Packets	1	1
Transmission Energy(mJ)	0.137	0.137

levels, signatures can be packed in a single RF packet consuming same transmission energy. However if the signature size increases we need to send multiple RF packets, thus increasing the transmission energy.

4.3 Total energy per authentication

Figure 4.6 shows total energy per signature measured at different operating frequencies of a microcontroller. Signature schemes with security level of 128 bits need more energy than those with security level of 80 bits because of the increased computational complexity and longer signature size. As the frequency increases, the time needed for computation decreases and the energy needed for computation decreases. But, the communication energy remains the same irrespective of the frequency of a microcontroller. In case of ECC, because the computational energy is the major contributor for the total energy, the total energy decreases as the frequency of operation increases.

Figure 4.7 compares the energy consumption of 80 bit and 128 bit security level at 10MHz. ECDSA is based on point multiplication over finite fields which is computationally expensive. Hence, computational energy for ECDSA is higher and it scales up cubically as we increase the required security level from 80 to 128. As the ECDSA based signatures are smaller in size, they need less communication energy. However it was observed that other protocols such as

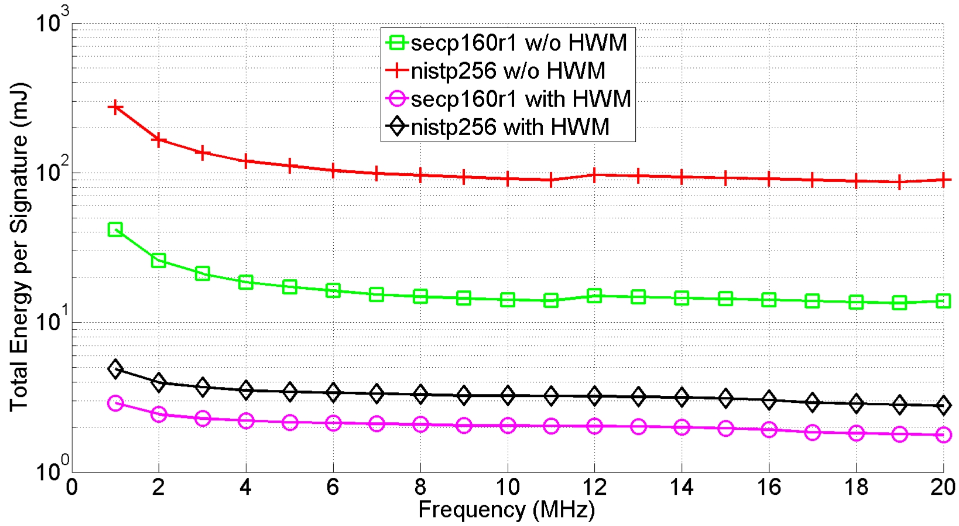

Figure 4.6: Total Energy Consumption per Signature

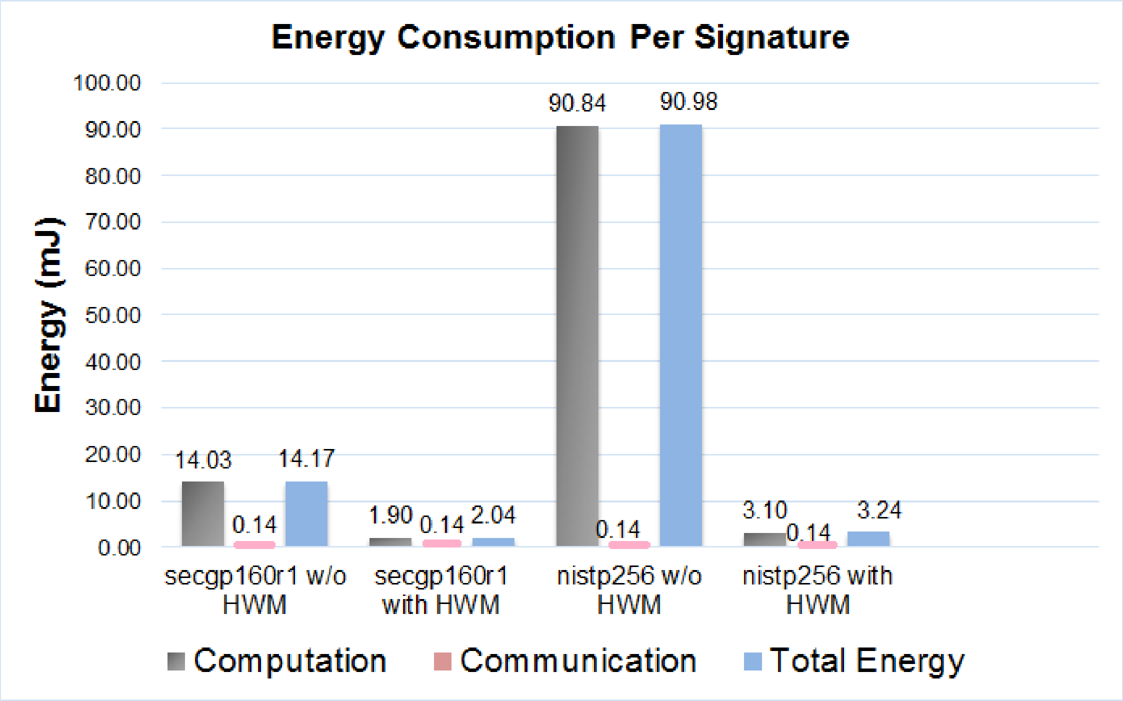

Figure 4.7: Comparison of energy for different configurations at 10MHz, 2.7V

Lamport-Diffie one-time hash-based signature scheme (LD-OTS) and the Winternitz one-time hash-based signature scheme (W-OTS) resulted in more communication energy than the computation. The reason for it is, these signatures are comparatively easy to compute but are longer in the length as compared to the ECDSA based signatures. Therefore, depending on choice of the signature protocol a system may be either computationally constrained or communication constrained.

Chapter 5

Methodology for Architecture-Energy Tuning

In this chapter, we show how the energy measurement method can be applied to meet the design requirements. In the following examples, we demonstrate how our energy/throughput curves can be used for system dimensioning. We consider two cases; (1) start from an energy constraint and derive the achievable performance in terms of the latency to complete one signature, (2) we start from a desired application performance, and we derive the required energy (and thus the required energy store). Since there are multiple energy/throughput curves (at different security levels, and at different architecture configurations), we propose that this evaluation is done concurrently over the all available curves, in order to analyze the design space. In the example discussed below, however, we will focus on a single security level and a microcontroller with a hardware multiplier.

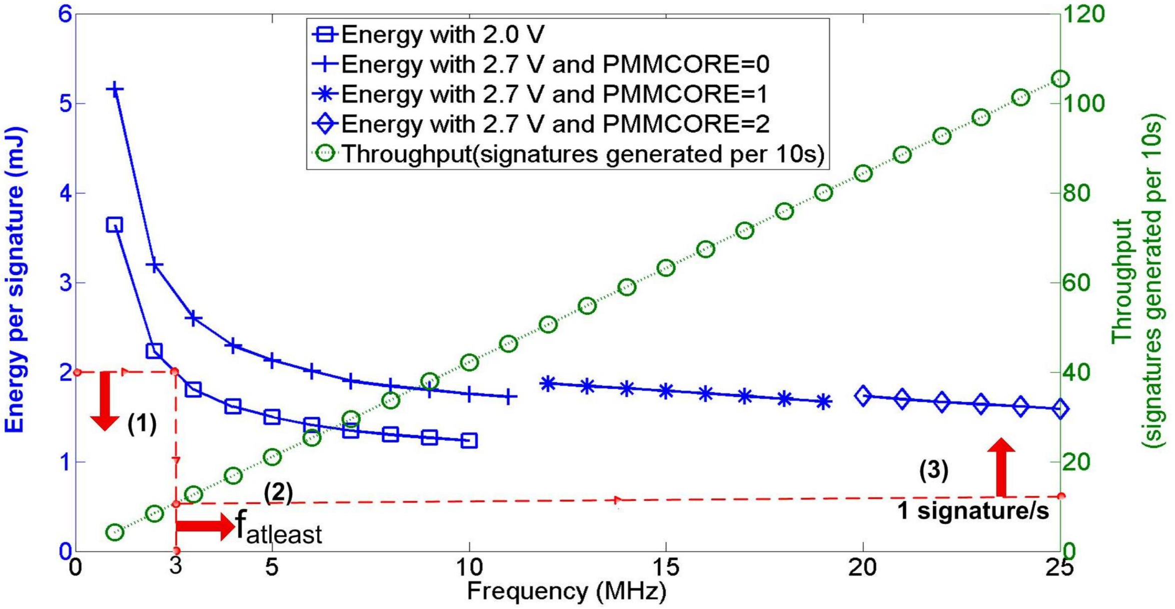

Figure 5.1: Architecture tuning for energy constrained system

5.1 Energy constrained system

Figure 5.1 shows the energy curve for the 80-bit security level. We assume an energy budget of 2 mJ per signature, and we assume a 2V operating level. This limit sets a *minimum* operating frequency for the microcontroller. The 2 mJ energy level requires a 3MHz operating frequency, which enables a signature to complete in a one second. If we increase the core voltage to 2.7V, the minimum operating frequency at 2mJ per signature will increase to 9MHz, and the signature will complete in 0.25 seconds.

5.2 Throughput constrained system

In second example, we to start from the required signature throughput. The example shown in Figure 5.2 needs to complete a signature in 1/6 of a second. We use our graphs to decide the operating frequency and to find required energy per signature. First, we note that the system needs to operate at least at 14MHz. At that frequency, only the 2.7V core voltage

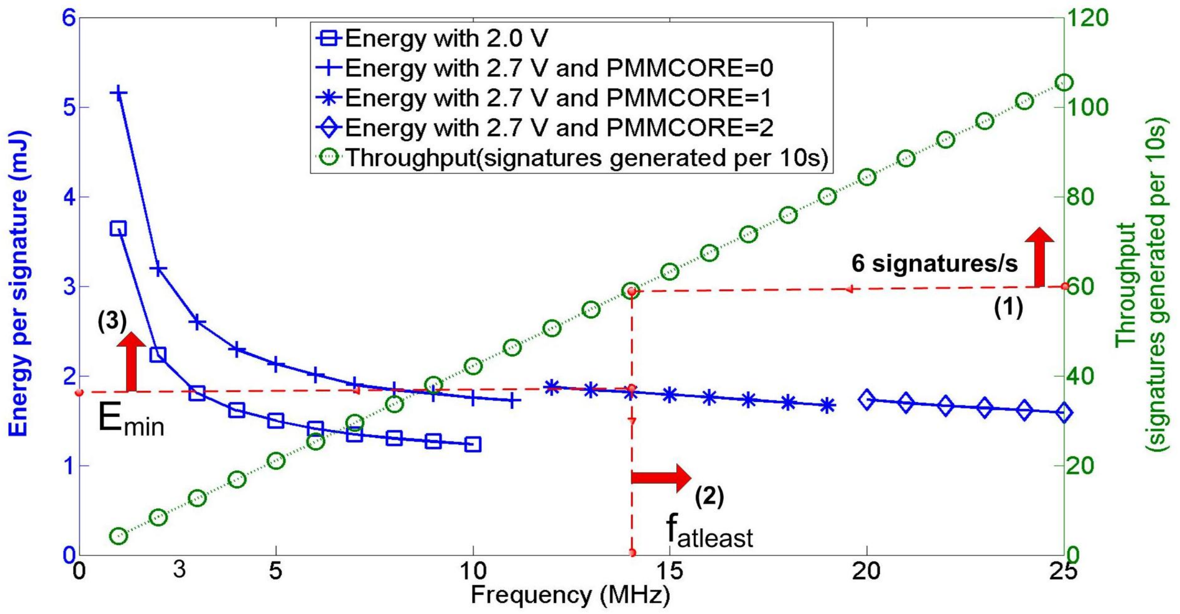

Figure 5.2: Architecture tuning for throughput constrained system

mode is available. Under this setting, the corresponding energy per signature is 1.9mJ.

Chapter 6

Energy Model

As the battery replacement is a major bottleneck for sensor networks, we propose integration of an energy harvester with the sensor node itself. As the amount and availability of harvested energy is limited, it becomes necessary to efficiently store and utilize the harvested energy. We introduce a model that estimates the total energy needed for an authentication that guides the design of an energy harvester.

6.1 Required Supercapacitor voltage calculation

In order to calculate the required energy for one authentication, we need to precisely measure both computation as well as communication energy. The required supercapacitor voltage is calculated using (6.1). We consider a safety margin of twice the required energy. As solar panel harvests energy, supercapacitor voltage increases and MSP430 periodically monitors this level to control its operational mode.

Energy stored in capacitor > Computation + Communication Energy

$$\frac{C.V^2}{2} = 2.(E_c + E_{rf}) + E_{ov}$$

$$\frac{C.V^2}{2} = \frac{2.(E_c + E_{rf}) + 125}{1000}$$

$$V = \sqrt{\frac{2.(E_c + E_{rf}) + 125}{500.C}} \tag{6.1}$$

where C is supercapacitor value in `Farad`, V is supercapacitor voltage in `Volts`, E_c is computational energy in `mJ`, E_{rf} is communication energy in `mJ`, E_{ov} is energy harvester overhead which is 125`mJ`.

6.2 Energy Model for decision making

If there is sufficient charge accumulated on the supercapacitor, authentication is performed and signature is transmitted over the RF to the server. Otherwise microcontroller goes to sleep mode waiting for required supercapacitor voltage. Figure 6.1 summarizes different steps involved in the energy modeling. It shows the energy model of a system operating at 2.7V and 10MHz, performing ECDSA signatures, and either configured in periodic transmit mode or in asynchronous transmit-receive mode.

We can derive a relation between solar panel rating, energy needed for one signature and the duty cycle as;

$$V_{solar}.I_{solar}.t = P_{sleep}.t + E_{sig}.Dutycycle.t$$

$$V_{solar}.I_{solar} = P_{sleep} + E_{sig}.Dutycycle \tag{6.2}$$

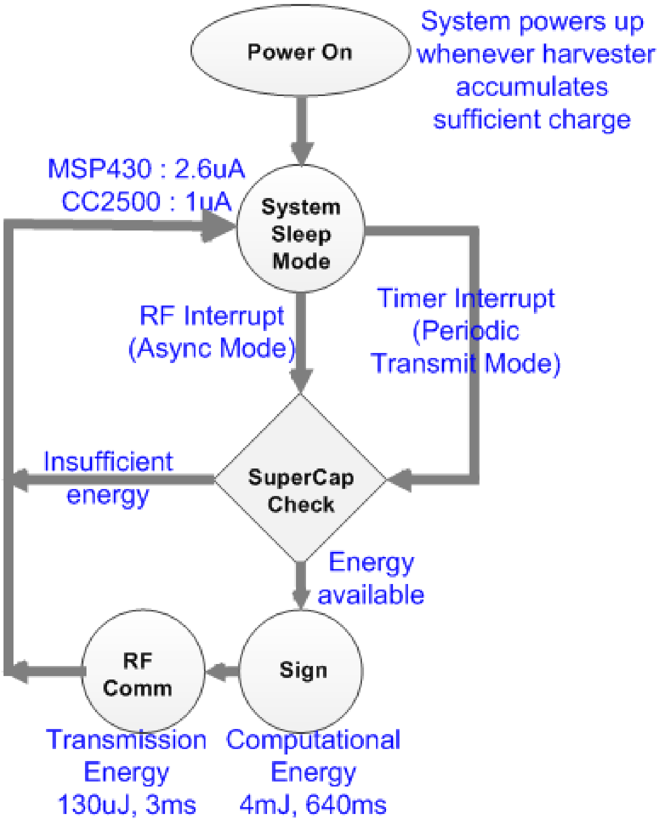
Power On
System powers up whenever harvester accumulates sufficient charge
MSP430 : 2.6uA
CC2500 : 1uA
System Sleep Mode
RF Interrupt (Async Mode)
Timer Interrupt (Periodic Transmit Mode)
Insufficient energy
SuperCap Check
Energy available
RF Comm
Sign
Transmission Energy 130uJ, 3ms
Computational Energy 4mJ, 640ms

Figure 6.1: Energy Model for decision making

where V_{solar} is the rated voltage of solar panel, I_{solar} is the rated current of solar panel, P_{sleep} is the sleep mode power of the system, E_{sig} is the energy needed for one signature and $Dutycycle$ denotes how often the signature is generated.

Using 6.2, we can design a system depending on the available amount of energy, required security level and duty cycle. For example, if the security level and duty cycle are fixed, we can determine the rating of the solar panel needed. We can determine what security algorithms can be run at what duty cycle with given energy budget. Such modeling needs the amount of energy needed for one computation and transmission, given application energy budget and timing requirements.

The above equation can be further extended to determine how long the system can run on the supercapacitor's charge when no solar energy is available as given below;

$$\begin{aligned} E_{supercap} &= P_{sleep}.t + (E_{sig}).(Dutycycle).t \\ \\ t &= E_{supercap}/(P_{sleep} + E_{sig}.Dutycycle) \qquad (6.3) \end{aligned}$$

where $E_{supercap}$ is the amount of energy the supercapacitor can hold.

Chapter 7

Conclusion

We demonstrated the importance of energy-architecture tuning for the lightweight platforms. The complex energy-provisioning environment of these applications requires a holistic approach that not only considers performance optimization, but also the energy and/or power needs. We implemented a self-powered wireless sensor node that is capable of providing efficient authentication between resource constrained node and a server, using digital signature scheme. We analyzed and compared the resource overhead in computation and communication for ECDSA for the security levels of 80 bit and 128 bit.

Our experimental results showed the impact of algorithm complexity and several architecture optimization techniques including voltage scaling, frequency scaling and use of a hardware multiplier, on the total energy consumption. For an MSP430 without a hardware multiplier, our prototype needs roughly six times as much energy per signature at the 128 bit security level in comparison to the 80 bit security level. When a hardware multiplier can be used, the difference is roughly two times. A second observation is that architecture specialization matters. Under constant security level, a hardware multiplier reduces the computational energy consumption by almost 8 times. Voltage scaling results in an additional gain factor of 2 in energy.

In addition, we also found that WSN implementing ECDSA consumes more computational energy than communication overhead. Depending on the choice of a signature protocol, a lightweight system can be either computationally constrained or communication constrained. A signature scheme with faster computation and shorter signature size would always result in minimum energy consumption. Furthermore, our results showed that running a microcontroller at highest possible speed minimizes the total energy consumption. Hence, in an energy-harvested sensor node, it is best to hold off on activities until the energy store has sufficient energy to support at least one complete iteration of the signature protocol.

Bibliography

[1] E. Wenger and M. Werner, "Evaluating 16-bit processors for elliptic curve cryptography," in *CARDIS*, pp. 166–181, 2011.

[2] C. Pendl, M. Pelnar, and M. Hutter, "Elliptic curve cryptography on the wisp uhf rfid tag," in *RFIDSec*, pp. 32–47, 2011.

[3] E. Wenger, M. Feldhofer, and N. Felber, "Low-resource hardware design of an elliptic curve processor for contactless devices.," in *WISA* (Y. Chung and M. Yung, eds.), vol. 6513 of *Lecture Notes in Computer Science*, pp. 92–106, Springer, 2010.

[4] A. Liu and P. Ning, "Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proceedings of the 7th international conference on Information processing in sensor networks*, IPSN '08, (Washington, DC, USA), pp. 245–256, IEEE Computer Society, 2008.

[5] D. H. Mane and P. Schaumont, "Energy-architecture tuning for ecc-based rfid tags," in *RFIDSec*, pp. 147–160, 2013.

[6] V. Cervenka, D. Komosny, L. Malina, and L. Mraz, "Energy efficient public key cryptography in wireless sensor networks," in *Innovations and Advances in Computer, Information, Systems Sciences, and Engineering* (K. Elleithy and T. Sobh, eds.), vol. 152 of *Lecture Notes in Electrical Engineering*, pp. 497–509, Springer New York, 2013.

[7] L. Batina, A. Das, B. Ege, E. B. Kavun, N. Mentens, C. Paar, I. Verbauwhede, and T. Yalçin, "Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures," in *RFIDSec*, pp. 103–112, 2013.

[8] J. A. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics.," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, 2005.

[9] P. Mitcheson, E. Yeatman, G. Rao, A. Holmes, and T. Green, "Energy harvesting from human and machine motion for wireless electronic devices," *Proceedings of the IEEE*, vol. 96, pp. 1457–1486, Sept 2008.

[10] E. Lai, A. Redfern, and P. K. Wright, "Vibration powered battery-assisted passive rfid tag," in *EUC Workshops*, pp. 1058–1068, 2005.

[11] N. Kong, T. Cochran, D. Ha, H. Lin, and D. Inman, "A self-powered power management circuit for energy harvested by a piezoelectric cantilever," in *Applied Power Electronics Conference and Exposition (APEC), 2010 Twenty-Fifth Annual IEEE*, APEC2010, 2010.

[12] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, (Piscataway, NJ, USA), IEEE Press, 2005.

[13] G. de Meulenaer, F. Gosset, O.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing,*, pp. 580–585, Oct 2008.

[14] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pp. 324–328, March 2005.

[15] R.Struik, "Aead ciphers for highly constrained networks," in *DIAC*, 2013. `http://2013.diac.cr.yp.to/slides/struik.pdf`.

[16] National Institute of Standards and Technology, "FIPS 186-3: Digital Signature Standard (DSS)," 2009.

[17] M. Buettner, B. Greenstein, and D. Wetherall., "Dewdrop: An Energy-Aware Runtime for Computational RFID," in *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, (Boston, MA, USA), 2011.

[18] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede, "Elliptic-curve-based security processor for rfid," *Computers, IEEE Transactions on*, vol. 57, no. 11, pp. 1514–1527, 2008.

[19] D. M. Hein, J. Wolkerstorfer, and N. Felber, "Ecc is ready for rfid - a proof in silicon," in *Selected Areas in Cryptography*, pp. 401–413, 2008.

[20] C. P. L. Gouvêa and J. López, "High speed implementation of authenticated encryption for the msp430x microcontroller," in *LATINCRYPT*, pp. 288–304, 2012.

[21] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[22] H.-J. Chae, D. J. Yeager, J. R. Smith, and K. Fu, "Maximalist cryptography and computation on the WISP UHF RFID tag," in *Proceedings of the Conference on RFID Security*, July 2007.

[23] G. Hinterwälder, C. Paar, and W. P. Burleson, "Privacy preserving payments on computational rfid devices with application in intelligent transportation systems," in *Proceedings of the 8th international conference on Radio Frequency Identification: security and privacy issues*, RFIDSec'12, (Berlin, Heidelberg), pp. 109–122, Springer-Verlag, 2013.

[24] B. Ransford, S. Clark, M. Salajegheh, and K. Fu, "Getting things done on computational rfids with energy-aware checkpointing and voltage-aware scheduling," in *Proceedings of*

the 2008 conference on Power aware computing and systems, HotPower'08, (Berkeley, CA, USA), pp. 5–5, USENIX Association, 2008.

[25] H. Wang and Q. Li, "Efficient implementation of public key cryptosystems on mote sensors (short paper)," in *Proceedings of the 8th international conference on Information and Communications Security*, ICICS'06, (Berlin, Heidelberg), pp. 519–528, Springer-Verlag, 2006.

[26] "Colin O'Flynn, OPENADC," 2012. `http://newae.com/tiki-index.php?page=OpenADC`.

[27] C. O'Flynn, "Power analysis for cheapskates," 2012. `https://media.blackhat.com/ad-12/O%27Flynn/bh-ad-12-for-cheapskates-o%27flynn-WP.pdf`.

[28] "Xilinx Spartan-6 FPGA LX9 MicroBoard." `http://www.em.avnet.com/en-us/design/drc/Pages/Xilinx-Spartan-6-FPGA-LX9-MicroBoard.aspx`.

[29] "Texas Instruments MSP430F5438A, Mixed Signal Microcontroller." `http://www.ti.com/lit/ds/symlink/msp430f5438a.pdf`.

[30] "Texas Instruments MSP430x5xx and MSP430x6xx Family User's Guide 2013." `http://www.ti.com/lit/ug/slau208m/slau208m.pdf`.

[31] "Texas Instruments Low Power 2.4GHz RF Transceiver." `http://www.ti.com/lit/ds/swrs040c/swrs040c.pdf`.

[32] "Texas Instruments RF CC2520EMK Evaluation Module Kit," `http://www.ti.com/tool/cc2520emk#descriptionArea`.

[33] "Anagear Power Management." `http://www.anagear.com/content/ANG1010`.

[34] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab, "Tinypbc: Pairings for authenticated identity-based non-interactive key

distribution in sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, 2011.