Journal of Universal Computer Science, vol. 19, no. 16 (2013), 2368-2384
submitted: 15/7/13, accepted: 27/9/13, appeared: 1/10/13 © *J.UCS*

Multiplication and Squaring with Shifting Primes on OpenRISC Processors with Hardware Multiplier

Leandro Marin
(Department of Applied Mathematics
Computer Sciences Faculty, University of Murcia
Reg. Campus of Int. Excellence Campus MareNostrum
Murcia, Spain
leandro@um.es)

Antonio J. Jara
(Research Institute for Oriented ICT (INTICO)
Computer Sciences Faculty, University of Murcia
Reg. Campus of Int. Excellence Campus MareNostrum
Murcia, Spain
jara@um.es)
(Research Institute of Information Systems,
University of Applied Sciences Western Switzerland (HES-SO)
Sierre, Switzerland
antonio.jara@hevs.ch)

Antonio F. Skarmeta
(Research Institute for Oriented ICT (INTICO)
Computer Sciences Faculty, University of Murcia
Reg. Campus of Int. Excellence Campus MareNostrum
Murcia, Spain
skarmeta@um.es)

Abstract: Cryptographic primitives are the key component in the security protocols to support the authentication, key management and secure communication establishment. For that reason, this work presents the optimization of the Elliptic Curve Cryptography through the usage of Shifting Primes for constrained devices. Specifically, this presents the optimization for the chipsets JN51XX from NXP/Jennic, which are based on OpenRISC architecture and offer a class-2 constrained device. In details, Shifting Primes features have allowed to optimize the multiplication and squaring through a double accumulator and shifting reduction. This work is ancillary to the previous works about optimization of Shifting Primes for class-1 constrained devices. The optimization of the Elliptic Curve Cryptography for the class-2 constrained devices brings several opportunities for realistic scenarios, where the security interoperability between a gateway (class-2 device) and end-nodes (class 1 devices) is a major requirement.

Key Words: Security GT, Internet of Things, Elliptic Curve Cryptography, Shifting Primes, OpenRISC
Category: GT, B.4.1, E.3, G.1.1, B.2.1

1 Introduction

The Internet of Things (IoT) [Atzori et al., 2010, Zhang et al., 2012] is one of the main drivers for the evolution of the Internet towards the Future Internet.

Nowadays, sensors, actuators and devices (so-called things), are connected to the Internet through gateways and platforms such as Supervisory Control and Data Acquisition platforms (SCADAs), panels, and brokers. These gateways and platforms break the end-to-end connection with the Internet. For that reason, this initial approach is defined as an Intranet of Things [Zorzi et al., 2010].

The Intranet of Things is being extended to smart things, such as lights, watches, clinical devices, proximity sensors, etc. [Kortuem et al., 2010] with a higher scalability, pervasiveness, and integration into the Internet Core. This extension is leading to reach a real IoT, where things are first class citizens in the Internet, and they do not need to relay any more on a gateway, middleware, proxy, or broker.

IoT requires both an architecture and products that allow for the extension of Internet technologies, in order to reach a Future Internet of Things, Services and People.

IoT drives towards integrating everything into the Internet Core. This integration is motivated by the market wish to have all processes remotely accessible through an uniform medium while at the same time understanding that re-engineering an infrastructure to allow this for each application independently would be prohibitively costly and time-consuming. Moreover, the current evolution from uniform mass markets, to personalized ones [Chen and Prokopi, 2013], where the customization and user-specified adaptation is a requirement, makes the sort of uniform infrastructure found in the Internet, imperative. This allows many components to be re-used, and services to be shared, with correspondingly huge economies of scale and shortened implementation times.

IoT fills the gap between the needs arising from the evolution of the market, information, users, and things, by moving all of them to a common framework, the Internet. This is different from the current approach in such applications, where they are based on stand-alone and monolithic solutions designed for a narrow or -*stovepiped*- application domain. Users now require more flexibility and freedom. Offering a common framework allows choice among the available manufacturers, suppliers, service providers, delivery options, and payment services. While this obviates the need for standalone or proprietary solutions, it also requires a high level of integration.

IoT allows communication among very heterogeneous devices connected via a very wide range of networks through the Internet infrastructure. IoT devices and resources are any kind of device connected to Internet, from existing devices, such as servers, laptops, and personal computers, to emerging devices such as smart phones [Costa et al., 2010], smart meters, sensors, identification readers, and appliances.

In addition to the physical devices, IoT is also enriched with the cybernetic resources and Web-based technologies [Pedrinaci and Domingue, 2010]. For that purpose, IoT is enabled with interfaces based on Web Services such as RESTFul architecture and the novel protocol for Constrained devices Applications Protocol (CoAP) [Shelby et al., 2013]. These interfaces enable the seamless integration of the IoT resources with information systems, management systems, and the humans. Reaching thereby a universal and ubiquitous integration among human networks (i.e., society), appliance networks, sensor networks, machine networks, and, in definitive, everything networks.

IoT offers several advantages and new capabilities for a wide range of application areas. For example, nowadays IoT is finding applications for the development of *Ambient Intelligent* [Kofod-Petersen and Cassens, 2010], *Smart Cities*, starting with *Smart Grid* [Ling et al., 2012], *Smart Lighting* and transport with new services such as *Smart Parking* and the *Bicycle Sharing System* from Barcelona (Spain) [FroehlichJon et al., 2008] for building sustainable and efficiently smart ecosystems.

The application of the IoT is not limited to high scale deployments such as the locations in Smart Cities, elsewhere it can also be considered for epayment [Zoran et al., 2007], consumer electronics, vehicular communications, industrial control, building automation, logistic, retail, marketing, and healthcare [Glascock and Kutzik, 2006]. Other applications can include voice and video [Vaidya et al., 2011], [Zhou et al., 2010], [Denko et al., 2009].

But, the exposition of constrained resources to Internet, through a direct connectivity presents a challenge in terms of security.

The majority of the IoT applications need to take into considerations the support of mechanisms to carry out the authentication, authorization, access control, and key management. In addition, due to the reduced capabilities from the constrained devices enabled with Internet connectivity, a higher protection of the edge networks needs to be considered with respect to the global network and the implementation problems should be considered as a major issue, see [Keoh et al., 2013].

This work presents an optimization of the cryptographic primitives based on Elliptic Curve Cryptography (ECC) for the JN51XX chipset family.

2 Security for the Internet of Things

Security is a wide concept which covers everything from authenticity (ensuring that the end-user is who is claimed to be), authority (ensuring that the end-user is allowed to perform the requested action), integrity (the data received is exactly the same data transmitted), and confidentiality (communication is not understandable for intermediary users, even when an intruder is in the network). These concepts are satisfied through a set of protocols, algorithms and cryptographic primitives [Chen et al., 2008].

The IoT security has been one of the most discussed and yet pending issues, even after of the existence of protocols for IPv6 network security such as IPSec, and for datagrams (i.e., UDP or CoAP) such as DTLS. Security for the IoT is not excessively extended and deployed because of the difficulties in configuring (IPSec) for end users and the lack of scalable certificate management for DTLS. Consequently, the majority of the Internet traffic continues being transmitted in plain text, i.e., unprotected.

For that reason, one of the initial actions in order to carry out an effective deployment of autonomous and unassisted IoT deployments that satisfies the scalability and self-management requirements from the IoT is the development of protocols for authentication and key management.

Specifically, on the one hand, the protocol for the authentication and key management at the network layer such as the Protocol for Carrying Authentication for Network Access (PANA) [Marin-Lopez et al., 2012] is being considered by the research institutions and also industrial alliances, such as the ZigBee Alliance for their ZigBee IP stack [Sturek, 2009].

On the other hand, the IPSec set of protocols (i.e., Internet Key Exchange (IKE) and Encapsulation Security Protocol (ESP)), and another protocols at the medium access layer such as 802.1x, are also being considered. All of these share the usage of the Extensible Authentication Protocol (EAP) to transport the security credentials.

Therefore, the challenge is not limited to the protocol, else the EAP scheme needs to be optimized in terms of a proper support of the required cryptographic primitives by the constrained device, i.e., symmetric cryptography algorithm to protect the packet, hash function to ensure the integrity and authenticate of the packet [Zia and Zomaya, 2011, Lee et al., 2012], and finally asymmetric cryptographic algorithm to carry out the key exchange and initial authentication .

Some initial works for the IoT have been proposed for IPSec such as IPSec for Contiki OS [Raza et al., 2012a], where several pending problems have been found, since for example a low version of the symmetric cryptography with 32-bit keys is used, such as AES-CBC-32, which are very weak. In addition, this relies on pre-shared keys for IPsec, which is not very scalable. Therefore, it does not solve the scalability and self-management requirements.

In order to satisfy these requirements, a Key Management Protocol (KMP) can be considered, that allows keys to be refreshed periodically (therefore maintaining acceptable security levels). Specifically, an automatic key exchange mechanism is required; thereby, each node can keep track of the security associations (SA) that specify how a particular IP flow should be treated in terms of security.

The most extended KMP is IKE. A very simple approach of IKE has been defined in [Raza et al., 2012b], which does not satisfy all the requirements and functionality for a full SA establishment.

Other issues from IPSec is that the overhead caused by a IPsec packet (the extra bytes on the IP header) can force the packet to be fragmented (the link layer payload that includes the extra IPsec bytes becomes bigger than the maximum size of a 802.15.4 packet), thus an extra packet must be sent to the link layer and the energy/network overhead will become bigger. In addition, this overhead problem is worse with the ESP mode of IPSec, since the internal headers of IPv6 and UDP are encrypted and consequently cannot be compressed.

In addition to IPSec, the majority of works from the CORE Working Group in IETF are focused on the integration of security through the transport layer security solutions such as DTLS for CoAP. DTLS is the default security for CoAP.

A pre-shared key mode (PSK) is also considered by CoAP, with the aforementioned problems regarding the lack of scalability for this pre-establishment of the security credentials.

CoAP also offers a very interesting approach based on RawPublicKey, i.e., a solution based on the use of an asymmetric key pair, but without an X.509 certificate metadata. This approach is highly relevant since it can manage the identity issues mentioned in the introduction section, in order to verify the authenticity of the device and its link with the manufacturer. For example, the Certification Authority (CA) of the public key can also indicate the list of identities of the nodes, with which it can communicate. It can thereby indicate the entities which are trustworthy in the initial verification and bootstrapping phase.

CoAP also considers certificates, i.e., X.509 certificate that binds it to its Authority Name and is signed by some common trust root, e.g., the manufacturer.

In order to optimize DTLS for smart objects, DTLS 1.2 offers the schemes to re-use the cryptographic hardware support by the majority of the IEEE 802.15.4 transceivers [Rescorla and Modadugu, 2012]. In particular, it is based on AES CCM for the symmetric cryptography. In addition, considers the usage of Elliptic Curve Cryptography (ECC) for the asymmetric cryptography. Thereby, making it more suitable for these constrained devices.

Nowadays, DTLS is being considered by the Smart Energy profile for ZigBee alliance (SE 2.0), and it is also being considered as an adaptation of DTLS 1.2 in the IPSO Alliance based on the subset allowed by RFC6347.

In addition to the solutions presented, there is security support over the current Internet architecture based on IPv6 in the network layer and UDP/TCP for the transport layer, where the security is based on IPSec for IPv6 and DTLS/TLS for UDP/TCP respectively. Also two solutions from the IETF to support the ID/Locator split have been defined. The first, HIP, has been developed by the Host Identity Protocol (HIP) Working Group, a group mainly focused on improving security of the Future Internet, and the HIP Diet EXchange (HIP DEX) [Nie et al., 2011], which has been optimized for constrained environments such as the Internet of Things. HIP offers in a single mechanism the capabilities for authentication and establishment of the communication.

Therefore, the goals to solve for the security support are, first, to optimize cryptographic primitives for the described protocols. Specially, ECC for the asymmetric cryptography.

Thereby in the future works it will be feasible to analyze and evaluate the impact of IP security protocol (IPSec) for constrained devices.

3 Asymmetric cryptography for constrained devices

Asymmetric cryptography is applied to the different methods and mechanisms developed by the community such as the mentioned IPSec IKE. Asymmetric cryptography has been considered mandatory in order to satisfy scalability of security and the goal to build highly scalable and autonomous solutions. Specifically, the optimization of the cryptographic primitives for constrained devices such as the 16-bits microprocessor MSP430 from Texas Instrument (commonly used in IoT devices such as 6LoWPAN, active RFID and DASH7) presented in [Marin et al., 2011, Marin et al., 2013b].

That work solved the mathematical optimization of cryptographic primitives for asymmetric cryptography based on a special pseudo-Mersenne primes, which we have denominated *Shifting Primes*. These primes can be used for ECC primitives with 160-bit keys in a highly optimal way. Specifically, this allows us to carry out ECC scalar multiplication within 5.42 million clock cycles over MSP430 devices without a hardware multiplier. This result reduces, for a microprocessor working at 8Mhz, the time required for the basic ECC protocols to the boundary of 1s (the key generation and the Diffie-Hellman protocol is basically only one scalar multiplication). This is even less time than offered by the TinyECC implementation for devices with fast multiplication hardware instruction [Liu and Ning, 2008].

This work presents the usage of the *Shifting Primes* for the 32-bits microprocessor JN5139 from NXP/Jennic based on OpenRISC architecture, such as described in [Beuran et al., 2012]. This microprocessor presents higher capabilities and consequently offers higher performance for this operation. This microprocessor has a multiplication instruction that can be used for the implementation, and this produces big differences between JN5139 and MSP430.

The following sections present how the properties of the *Shifting Primes* can be exploited to optimize the multiplication and squaring. Specifically, Section

4 presents how a double accumulator and shifting reduction has been defined thanks to the properties of the *Shifting Primes* to simplify the multiplication and squaring. The details of both optimizations are described in details in the Section 5 and Section 6 respectively.

4 The Double Accumulator and Shifting Reduction

The key point in the following algorithms is the fact that the Jennic/NXP JN51XX family of chipsets, based on the OpenRISC architecture, support 32-bit registers. This OpenRISC architecture supports an instruction to multiply two registers, although it is limited to store the result in a single 32-bit register. This means that only are stored the 32 least significant bits of the result.

In order to implement big number multiplication, it is required to split the operand in 16-bit operands and use the multiplication in the microprocessor to get 32-bit results. These partial results will be added to obtain the final result.

The set of registers that will be used to add the partial results will be called the accumulator. This accumulator will be used also to make the reductions modulo p whenever necessary.

Before going any further, let introduce some notation:

1. $C_i = 2^{16i}$ for $i \in \{0, \cdots, 9\}$, they are the 10 pointers for the shifting of 16 bits inside a 160 bits number.

2. $B_i = 2^{32i}$ for $i \in \{0, \cdots, 9\}$, they are the 10 pointers for the shifting of 32 bits inside a 320 bits number.

3. $D_i = 2^{32i+16}$ for $i \in \{0, \cdots, 9\}$, they are the 10 pointers for the shifting of 32 bits inside a 320 bits number for not aligned operations.

4. p is the shifting prime $p = 2uC_9 - 1$. Specifically, the examples presented in this work will be based on the shifting prime with value equal to $u = 0x6400 = 51200$.

5. Given a 32-bit number W, It will be defined as $\mathsf{hi}_k(W)$ the k most significant bits of W and $\mathsf{lo}_k(W)$ the k lowest significant bits. This notations satisfies the equality $W = \mathsf{hi}_k(W)2^k + \mathsf{lo}_{32-k}(W)$. If it is not specified any subscript, then $k = 16$. Then it is written $W = \mathsf{hi}(W)2^{16} + \mathsf{lo}(W)$ where $\mathsf{hi}(W)$ and $\mathsf{lo}(W)$ are 16-bit numbers.

Following this notation, a 160-bit number x can be written $x = x_0C_0+x_1C_1+ \cdots + x_9C_9$ with $x_i \in \{0, \cdots, 2^{16} - 1\}$. Since, it is required to multiply two 16-bit numbers a and b in different position, for example aC_i and bC_j. The result ab (which is a 32-bit number) will be in position C_{i+j}, but this can be not 32-bit aligned. It will be aligned only in case $i + j$ is an even number. This produces a problem of misalignment of the half of the multiplications. For that reason, it is required to use the B_i for the aligned results and the D_i for the not aligned ones. All these assumptions are summarized in the following proposition (with some useful properties of the symbols introduced):

Proposition 1. *These values satisfy the following properties for* $i, j \in \{0, \cdots, 9\}$ *and any 32-bit number* W*:*

1. $C_iC_j = \begin{cases} B_{(i+j)/2} & \text{if } i \text{ and } j \text{ have the same parity} \\ D_{(i+j-1)/2} & \text{otherwise} \end{cases}$
2. $B_i \equiv 2uD_{4+i}$ *modulo* p
3. $D_i \equiv 2uB_{5+i}$ *modulo* p
4. $WB_i = \mathsf{hi}(W)D_i + \mathsf{lo}(W)B_i$
5. $WD_i = \mathsf{hi}(W)B_{i+1} + \mathsf{lo}(W)D_i$

Proof. 1. $C_iC_j = 2^{16i+16j} = \begin{cases} 2^{32(i+j)/2} = B_{(i+j)/2} & \text{if } i \equiv j(2) \\ 2^{32(i+j-1)/2+16} = D_{(i+j-1)/2} & \text{otherwise} \end{cases}$

2. Since, a shifting prime i used, in particular $p = 2u2^{32\cdot4+16} - 1$, this has the modular equivalence $1 \equiv 2u2^{32\cdot4+16}$. Then $B_i = 2^{32i} \cdot 1 \equiv 2u2^{32i}2^{32\cdot4+16} = 2u2^{32(i+4)+16} = 2uD_{i+4}$.

3. Using the same equivalence, $D_i = 2^{32i+16} \cdot 1 \equiv 2^{32i+16}2^{32\cdot4+16} = 2^{32(i+4)+32} = B_{i+5}$.

4. $WB_i = \mathsf{hi}(W)2^{16}2^{32i} + \mathsf{lo}(W)2^{32i} = \mathsf{hi}(W)2^{32i+16} + \mathsf{lo}(W)2^{32i} = \mathsf{hi}(W)D_i + \mathsf{lo}(W)B_i$.

5. $WD_i = \mathsf{hi}(W)2^{16}2^{32i+16} + \mathsf{lo}(W)2^{32i+16} = \mathsf{hi}(W)2^{32i+32} + \mathsf{lo}(W)2^{32i+16} = \mathsf{hi}(W)B_{i+1} + \mathsf{lo}(W)D_i$.

In order to make the implementation more efficient, the accumulator is stored in a set of registers (to avoid operations with memory). The multiplication or the square of 160-bit numbers is a 320-bit number, therefore this requires 10 registers.

This is also required that at least one operand is stored in registers, since they need to be stored as 16-bits half words, then this requires another 10 registers; although it will be only used the least significant half of the registers.

The problem with the accumulator is that half of the partial results are not aligned, therefore it needs to be shifted 16-bits before adding with the align partial results. This continuous shifting requires a lot of computations, and instead of it, this work proposed an optimization solution based on the use of a double accumulator, one that accumulates the aligned partial results, and another one for the other partial results.

The problem is that this requires 10 extra registers, and this makes a total requirement of 30 registers, that is too much for the Jennic/NXP JN51XX Open-RISC processor, since this is limited to 32 registers, and some of them cannot be used such as the stack pointer, the constant register r_0, and the link register. Finally, it is also required some temporal registers for the partial results before addition.

It is required to compute modular multiplications, therefore it can use the shifting prime to have modular equivalent results that can be written using less registers. This is also relevant because in the Montgomery representation, a 160-bit shifting of the final result is required, therefore it needs to store all the information in the most significant words.

The trick is given by properties (2) and (3) in Proposition 1. The transfer of the content from the least significant words to he highest ones multiplying by $2u$, since $B_i \equiv 2uD_{4+i}(p)$, $D_i \equiv 2uB_{5+i}(p)$, and consequently the final result is also equivalent. To exploit this property, the prime p must be a shifting prime, since the accumulator information can be concentrated in the registers of any of both accumulators. Therefore, let reuse the other registers. The unique pending problem is that multiply by $2u$ increases the size in 16 bits, therefore the corrections have to be made in both accumulators. For example, let W be a 32-bit number, then

$$WB_i = \mathsf{hi}(W)D_i + \mathsf{lo}(W)B_i = 2u\mathsf{hi}(W)B_{5+i} + 2u\mathsf{lo}(W)D_{4+i}$$

$$WD_i = \mathsf{hi}(W)B_{i+1} + \mathsf{lo}(W)D_i = 2u\mathsf{hi}(W)D_{5+i} + 2u\mathsf{lo}(W)B_{5+i}.$$

To sum up, this technique let implement the double accumulator with much less than 20 registers. Thereby, in total is required the at least 10 for the accumulator, 10 for the parameter and some others for partial computations.

5 Multiplication

The multiplication algorithm is explained in detail in [Marin et al., 2013a]. This section explains it briefly with the notation introduced in the previous section. This will allow to compare the multiplication algorithm with the squaring that will be presented in the following section.

Let $x = \sum_{i=0}^{9} x_i C_i$ and $y = \sum_{j=0}^{9} y_j C_j$ be the operands, it needs to be computed the Montgomery multiplication of x and y, this is $xy2^{-160}$, or what it is equivalent xy written on 10 words, but concentrated in the most significant words using the relations induced by the shifting prime that has been described in the previous section.

Since, a double accumulator is used, it is just required to merge them and carry out the required shifting at the end.

The accumulator A starts with the value $A_0 = 0$, and it is increased in several steps. Therefore, the accumulator is evolving in the following way A_1, A_2, etc. At the end, the accumulator will be xy. The value increased is

$$A_{i+1} - A_i = x_i C_i y = \sum_{j=0}^{9} x_i y_j C_i C_j =$$

Two different situations need to be managed, On the one hand, when i is even, the value is:

$$\begin{array}{lllll} =x_i y_0 B_{i/2} & +x_i y_2 B_{i/2+1} & +x_i y_4 B_{i/2+2} & +x_i y_6 B_{i/2+3} & +x_i y_8 B_{i/2+4} \\ +x_i y_1 D_{i/2} & +x_i y_3 D_{i/2+1} & +x_i y_5 D_{i/2+2} & +x_i y_7 D_{i/2+3} & +x_i y_9 D_{i/2+4} \end{array}$$

On the other hand, when i is odd, the value is:

$$\begin{array}{lllll} =x_i y_0 D_{\frac{i-1}{2}} & +x_i y_2 D_{\frac{i-1}{2}+1} & +x_i y_4 D_{\frac{i-1}{2}+2} & +x_i y_6 D_{\frac{i-1}{2}+3} & +x_i y_8 D_{\frac{i-1}{2}+4} \\ +x_i y_1 B_{\frac{i-1}{2}} & +x_i y_3 B_{\frac{i-1}{2}+1} & +x_i y_5 B_{\frac{i-1}{2}+2} & +x_i y_7 B_{\frac{i-1}{2}+3} & +x_i y_9 B_{\frac{i-1}{2}+4} \end{array}$$

The odd and even cases follows an almost similar calculus, the main difference is the interchange of the roles for the D's and B's variables. This calculus has been designed in order to simplify the implementation and make feasible the re-use of code in order to optimize memory footprint.

The values in the least significant words are moved to the most significant words in every step using the following equations:

$$WB_i = \mathsf{hi}(W)D_i + \mathsf{lo}(W)B_i = 2u\mathsf{hi}(W)B_{5+i} + 2u\mathsf{lo}(W)D_{4+i}$$
$$WD_i = \mathsf{hi}(W)B_{i+1} + \mathsf{lo}(W)D_i = 2u\mathsf{hi}(W)D_{5+i} + 2u\mathsf{lo}(W)B_{5+i}.$$

This equations let store the accumulator with only 10 registers. Since, in the last step of the algorithm, both parts of the accumulator are combined. The information about the performance and the detailed implementation is presented in [Marin et al., 2013a].

6 Squaring

The optimization of ECC for constrained devices is focused on the multiplication and squaring operations. The previous section has explained how to optimize the registers usage for the multiplication, this section explains how to use the shifting primes to implement $x^2(p)$ for $p = 2uC_9 - 1$ and $x = \sum_{i=0}^{9} x_i C_i$.

In order to implement the squaring, it could be re-used the standard multiplication with $x = y$. But, in that case, it is requiring all the computation time of a standard multiplication, when the squaring can be optimized and reached a lower computation time. Therefore, as the majority of the optimization techniques for the implementation of cryptographic primitives is a trade-off between memory footprint and computational time required for the operation set.

In the case of the squaring, the extra code is highly worth, when it is required to maximize the speed for the ECC protocols. In addition, the usage of the shifting primes offer a set of properties that allows even to optimize much more the squaring implementation.

When x^2 is computed, it presents repeated multiplications $x_i x_j$ and $x_j x_i$, these multiplications produce $2x_i x_j$ when $i \neq j$, i.e., the same result. Therefore, it can be reduced the number of multiplications, and it can be considered an accumulator that stores the values $x_i x_j$ and multiply by 2 the value at the end.

The problem appears with the values x_i^2. These values are not multiplied by 2 and therefore, they require a special treatment. The shifting prime solves this following the next equation:

$$\frac{1}{2}(x_i C_i)^2 = \frac{1}{2} x_i^2 B_i = \frac{1}{2}\left(\mathsf{hi}_{15}(x_i^2)2^{17} + \mathsf{lo}_{17}(x_i^2)\right) B_i =$$

$$\mathsf{hi}_{15}(x_i^2)2^{16} B_i + \frac{1}{2} 2u\mathsf{lo}_{17}(x_i^2) D_{4+i} = \mathsf{hi}_{15}(x_i^2) D_i + u\mathsf{lo}_{17}(x_i^2) D_{4+i}$$

The constant 2 that multiplies u in the Proposition 1(2) let add $\frac{1}{2}(x_i C_i)^2$ in the accumulator without a division by 2. This operation is done for the values $i \in \{0, \cdots, 4\}$. The last values $x_i^2 C_i$ for $i \in \{5, \cdots, 9\}$ will be added at the end, after to carry out the multiplication of the accumulator by 2.

The temporal value for the accumulator is defined as:

$T_i = \frac{1}{2}\left(x^2 - \left(\sum_{j=i}^{9} x_j C_j\right)^2\right)$ for $i = 0, 1, 2, 3, 4, 5$. This temporal value is almost the accumulator.

$$T_{i+1} - T_i = \frac{1}{2}\left(\left(\sum_{j=i+1}^{9} x_j C_j\right)^2 - \left(\sum_{j=i+1}^{9} x_j C_j\right)^2\right) =$$

$$\frac{1}{2} x_i C_i \left(x_i C_i + 2\sum_{j=i+1}^{9} x_j C_j\right)$$

$$\frac{1}{2}(x_i C_i)^2 + \sum_{j=i+1}^{9} x_i x_j C_i C_j = \frac{1}{2} x_i^2 B_i + \sum_{j=i+1}^{9} x_i x_j C_i C_j =$$

$$\mathsf{hi}_{15}(x_i^2)D_i + u\mathsf{lo}_{17}(x_i^2)D_{4+i} + \sum_{j=i+1}^{9} x_i x_j C_i C_j$$

It is required to compute these values for $i = 1, 2, 3, 4, 5$, note that for $T_0 = 0$

$$\begin{aligned}
T_1 - T_0 =& (x_0x_1 + \mathsf{hi}_{15}(x_0^2))D_0 + x_0x_3D_1 + x_0x_5D_2 + x_0x_7D_3 & +u\mathsf{lo}_{17}(x_0^2)D_4+\\
& & x_0x_9D_4+\\
& x_0x_2B_1 + x_0x_4B_2 + x_0x_6B_3 & +x_0x_8B_4\\
T_2 - T_1 =& (x_1x_2 + \mathsf{hi}_{15}(x_1^2))D_1 + x_1x_4D_2 + x_1x_6D_3 + x_1x_8D_4 & +u\mathsf{lo}_{17}(x_1^2)D_5+\\
& x_1x_3B_2 + x_1x_5B_3 + x_1x_7B_4 & +x_1x_9B_5\\
T_3 - T_2 =& (x_2x_3 + \mathsf{hi}_{15}(x_2^2))D_2 + x_2x_5D_3 + x_2x_7D_4 + x_2x_9D_5 & +u\mathsf{lo}_{17}(x_2^2)D_6+\\
& x_2x_4B_3 + x_2x_6B_4 + x_2x_8B_5 & \\
T_4 - T_3 =& (x_3x_4 + \mathsf{hi}_{15}(x_3^2))D_3 + x_3x_6D_4 + x_3x_8D_5+ & u\mathsf{lo}_{17}(x_3^2)D_7+\\
& x_3x_5B_4 + x_3x_7B_5 + x_3x_9B_6 & \\
T_5 - T_4 =& (x_4x_5 + \mathsf{hi}_{15}(x_4^2))D_4 + x_4x_7D_5 + x_4x_9D_6+ & u\mathsf{lo}_{17}(x_4^2)D_8+\\
& x_4x_6B_5 + x_4x_8B_6 &
\end{aligned}$$

The multiplications $u\mathsf{lo}_{17}(x_i^2)$ are always below 2^{32} because $u < 2^{15}$, but it can present a carry in the addition $x_0x_9 + u\mathsf{lo}_{17}(x_0^2)$, therefore in order to avoid that carry, the followed technique is to delay the addition of x_0x_9.

Regarding the computation of $\frac{1}{2}\left(x^2 - \sum_{i=5}^{9} x_i^2 B_i\right)$, it is required to include the following terms:

$$x_5x_7B_6 + x_5x_9B_7 + x_6x_8B_7 + x_7x_9B_8+$$

$$x_5x_6D_5 + x_5x_8D_6 + x_6x_7D_6 + x_6x_9D_7 + x_7x_8D_7 + x_8x_9D_8$$

These values can be included with the other ones that have been used to compute the T_i.

Then. the new terms are underlined and removed the $x_0x_9D_4$ from the first addition. These terms will be included in Δ_8' such as follows:

$$
\begin{array}{llllll}
\Delta_0 = & (x_0x_1 + \mathsf{hi}_{15}(x_0^2))D_0 & +x_0x_3D_1 & +x_0x_5D_2 & +x_0x_7D_3 & +u\mathsf{lo}_{17}(x_0^2)D_4 \\
\Delta_1 = & & x_0x_2B_1 & +x_0x_4B_2 & +x_0x_6B_3 & +x_0x_8B_4 \\
\Delta_2 = & (x_1x_2 + \mathsf{hi}_{15}(x_1^2))D_1 & +x_1x_4D_2 & +x_1x_6D_3 & +x_1x_8D_4 & +u\mathsf{lo}_{17}(x_1^2)D_5 \\
\Delta_3 = & & x_1x_3B_2 & +x_1x_5B_3 & +x_1x_7B_4 & +x_1x_9B_5 \\
\Delta_4 = & (x_2x_3 + \mathsf{hi}_{15}(x_2^2))D_2 & +x_2x_5D_3 & +x_2x_7D_4 & +x_2x_9D_5 & +u\mathsf{lo}_{17}(x_2^2)D_6 \\
\Delta_5 = & & x_2x_4B_3 & +x_2x_6B_4 & +x_2x_8B_5 & +\underline{x_5x_7B_6} \\
\Delta_6 = & (x_3x_4 + \mathsf{hi}_{15}(x_3^2))D_3 & +x_3x_6D_4 & +x_3x_8D_5 & +\underline{x_5x_8D_6} & +u\mathsf{lo}_{17}(x_3^2)D_7 \\
\Delta_7 = & & x_3x_5B_4 & +x_3x_7B_5 & +x_3x_9B_6 & +\underline{x_5x_9B_7} \\
\Delta_8 = & (x_4x_5 + \mathsf{hi}_{15}(x_4^2))D_4 & +x_4x_7D_5 & +x_4x_9D_6 & +\underline{x_6x_9D_7} & +u\mathsf{lo}_{17}(x_4^2)D_8 \\
\Delta_8' = & x_0x_9D_4 & +\underline{x_5x_6D_5} & +\underline{x_6x_7D_6} & +\underline{x_7x_8D_7} & +\underline{x_8x_9D_8} \\
\Delta_9 = & & x_4x_6B_5 & +x_4x_8B_6 & +\underline{x_6x_8B_7} & +\underline{x_7x_9B_8}
\end{array}
$$

The final computation is $x^2 \equiv 2(\Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 + \Delta_4 + \Delta_5 + \Delta_6 + \Delta_7 + \Delta_8 + \Delta_8' + \Delta_9) + \sum_{i=5}^{9} x_i^2 B_i$. The order in the additions is important, because this allows to free the registers that have the values x_i when they are already used. The order in which these registers are made available (i.e., free) is:

register	freed after adding
x_1	Δ_3
x_2	Δ_5
x_3	Δ_7
x_0	Δ_8'
x_4	Δ_9

The registers that have the values $x_5, \cdots, x_9$ can be used to store the final result, since they can be re-used to store their square value, and for adding on it two times the accumulated value given by the Δ's (after adding the D-values over the B-values).

During the accumulation process, it is required to move the least significant words to the highest significant ones, in order to have everything stored and calculated in terms of $B_5, ..., B_9$ and $D_5, ..., D_9$ (the value on D_9 is used only for carriers).

Therefore, it is required five registers for D's and other five for B's. The most significant register for D's will be used only for carriers. For each step, it is moved a register to the most significant part as follows:

First, it is already calculated the accumulated value αD_i, then:

$\alpha D_i = \mathsf{hi}(\alpha)B_{i+1} + \mathsf{lo}(\alpha)D_i = \mathsf{hi}(\alpha)B_{i+1} + u\mathsf{lo}(\alpha)B_{i+5}$. The value $\mathsf{hi}(\alpha)B_{i+1}$ can be added with the next $x_r x_s B_{i+1}$ because $x_r x_s + \mathsf{hi}(\alpha) < 2^{32}$. This process requires only one extra multiplication by u instead of two, that was required in the multiplication process explained in more details in the work presented in [Marin et al., 2013a].

7 Evaluation

The evaluation of the presented optimizations is carried out following a bottom-up approach. Specifically, on the one hand, the squaring optimization, and on the other hand, this is evaluated from the scalar multiplication, to the modular multiplication. Note, that the modular multiplication is the required operation for the cryptographic process of key generation, encryption, and decryption.

Regarding the squaring optimization, the computation cost in terms of number of scalar multiplications for a modular product xy is 119. In the computation of the squaring, i.e., x^2 is 70. Therefore, this presents a reduction of 59% in the number of single scalar multiplications. The time reduction is equal to the 65%, this time reduction is also influenced by the other operations involved, this reduction is limited by the other operations that are carried, which are not using the square operation.

In order to extrapolate the impact of the squaring optimization for the modular multiplication; let $\mathbf{M}$, which denotes the time required for a modular multiplication, and $\mathbf{S}$ the time required for squaring. In the literature, it is usual to consider that $\mathbf{S} \simeq 0.8\mathbf{M}$. But, the proposed algorithm offers a $\mathbf{S} \simeq 0.65\mathbf{M}$. This reduction has a high impact in the time required for the modular multiplication.

The time required for the modular multiplication also depends on the representation used. In particular, this work has used the Jacobian coordinates on Weierstrass curves, point addition (with $Z2 = 1$) can be done with $7\mathbf{M} + 4\mathbf{S}$, this means a difference $\frac{7+0.65*4}{7+4} \simeq 0.87$. Doubling can be done with $3\mathbf{M} + 5\mathbf{S}$, this means a reduction $\frac{3+0.65*5}{3+5} \simeq 0.78$. The doubling algorithm is used more often than point addition, then it is reached a final time using the special square that is 80% the time required using the same function for multiplication and squaring.

The results depend on the technique used for the scalar multiplication. There is a wide variety of formulas for different point representations and also different curves that can be used for cryptography. See for example the information found in [1] for a detailed compilation.

We can compare these results with other implementations given in the literature, for example [Chatzigiannakis et al., 2011] we can see an implementation that requires 11.121 sec for JN5139 for key generation. Even without this special

[1] Hyper Elliptic - http://www.hyperelliptic.org/EFD/g1p

square function, our implementation requires 140.37 ms (see [Marin et al., 2013a]). With this extra reduction, the time is reduced another 20 percent. The authors in [Chatzigiannakis et al., 2011] claim that their implementations are not optimized for speed, this can explain these big differences.

8 Conclusions and Future work

This work has presented the optimization cryptographic primitives for asymmetric key cryptography based on Elliptic Curve Cryptography, this has demonstrated that asymmetric key cryptography is feasible for constrained devices.

The ongoing work is focused on the definition of hybrid scenarios with the defined optimization based on Shifting Primes for scenarios with chipsets such as the MSP430 for the end-devices, and for chipsets such as the presented JN51XX for the gateway and border routers.

These hybrid scenarios will make use of high level algorithms such as IPSec and DTLS, where Elliptic Curve Cryptography is used for the establishment of the security association or session.

The future work is focused on solutions to carry out IoT/M2M trust verification, through a mechanism such as capabilities-based access control. Consequently, novel scenarios based on temporal access to resources can be defined. For example, a house proprietor with an access control solution (e.g. a smart door lock) is able to offer temporal access to his neighbor so as to go everyday at anytime from 15:00 to 18:00 in order to feed the pets and irrigate the plants.

The mechanisms required to offer secure solutions that make usage of the IoT capabilities feasible during usual human activities and behaviors, where devices and physical resources are involved, needs to be enhanced. As a result, these new mechanisms and solutions will facilitate the introduction of the IoT as part of the Internet-powered society.

Acknowledgment

This research has been conducted partially by the Intelligent Systems and Networks group of the University of Murcia, Espinardo, Spain. This research group has been awarded for its excellence as a research group in the frames of the Spanish "Plan de Ciencia y Tecnología de la Región de Murcia" from the "Fundación Séneca" (04552/GERM/06). The authors would like to thank the European Project "Universal Integration of the IoT through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability (IoT6)" from the FP7 with the grant agreement no: 288445, the Spanish Ministry for Industry, Tourism and Infrastructure, and the Ministry for Education, Social Politics and Sport for sponsoring the research activities under the grants "Architecture for Intelligent Respiratory Evaluation (AIRE)" (TSI-020302-2010-95), FPU program (AP2009-3981), the "Fundación Séneca" and the Spanish ministry for Science and Technology.

References

[Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, Vol. 54(no. 15):pp. 2787–2805.

[Beuran et al., 2012] Beuran, R., Nakata, J., Yasuo, T., and Shinoda, Y. (2012). Emulation testbed for ieee 802.15. 4 networked systems. *IEICE Transactions on Communications*, 95(9):2892–2905.

[Chatzigiannakis et al., 2011] Chatzigiannakis, I., Pyrgelis, A., Spirakis, P. G., and Stamatiou, Y. C. (2011). Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices. *CoRR*, abs/1107.1626.

[Chen et al., 2008] Chen, L., Dawson, E., Lai, X., Mambo, M., Miyaji, A., Mu, Y., Pointcheval, D., Preneel, B., Smart, N., Susilo, W., et al. (2008). Cryptography in computer system security. *Journal of Universal Computer Science*, 14(3):314–317.

[Chen and Prokopi, 2013] Chen, L. B. and Prokopi, M. (2013). Enabling Resource-Aware Ubiquitous Applications for Personal Cloud with a Pairing Device Framework. *Journal of Internet Services and Information Security (JISIS)*, 3(1/2):83–100.

[Costa et al., 2010] Costa, G., Lazouski, A., Martinelli, F., Matteucci, I., Issarny, V., Saadi, R., Dragoni, N., and Massacci, F. (2010). Security-by-contract-with-trust for mobile devices. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 1(4):75–91.

[Denko et al., 2009] Denko, M. K., Sun, T., Woungang, I., Rodrigues, J. J. P. C., and Chao, H.-C. (2009). A trust management scheme for enhancing security in pervasive wireless networks. In *GLOBECOM*, pages 1–6. IEEE.

[FroehlichJon et al., 2008] FroehlichJon, J., Neumann, J., and Oliver, N. (2008). Measuring the pulse of the city through shared bicycle programs. *Proc. of UrbanSense08*, pages pp. 16–20.

[Glascock and Kutzik, 2006] Glascock, A. P. and Kutzik, D. M. (2006). The impact of behavioral monitoring technology on the provision of health care in the home. *Journal of Universal Computer Science*, 12(1):59–79.

[Keoh et al., 2013] Keoh, S., Kumar, S., and Shelby, Z. (2013). Profiling of dtls for coap-based iot applications. `http://datatracker.ietf.org/doc/draft-keoh-dtls-profile-iot`.

[Kofod-Petersen and Cassens, 2010] Kofod-Petersen, A. and Cassens, J. (2010). Proxies for privacy in ambient systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 1(4):62–74.

[Kortuem et al., 2010] Kortuem, G., Kawsar, F., Fitton, D., and Sundramoorthy, V. (2010). Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, Vol. 14(no. 1):pp. 44–51.

[Lee et al., 2012] Lee, S. M., Kim, D. S., and Park, J. S. (2012). A survey and taxonomy of lightweight intrusion detection systems. *Journal of Internet Services and Information Security (JISIS)*, 2(1/2):119–131.

[Ling et al., 2012] Ling, A. P. A., Kokichi, S., and Masao, M. (2012). Enhancing smart grid system processes via philosophy of security-case study based on information security systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 3(3):94–112.

[Liu and Ning, 2008] Liu, A. and Ning, P. (2008). Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, pages pp. 245–256. IEEE.

[Marin et al., 2013a] Marin, L., Jara, A. J., and Gómez-Skarmeta, A. F. (2013a). Shifting primes on openrisc processors with hardware multiplier. In Mustofa, K., Neuhold, E. J., Tjoa, A. M., Weippl, E., and You, I., editors, *ICT-EurAsia*, volume 7804 of *Lecture Notes in Computer Science*, pages 540–549. Springer.

[Marin et al., 2013b] Marin, L., Jara, A. J., and Skarmeta, A. (2013b). Shifting primes on openrisc processors with hardware multiplier. In *Information and Communication Technology, Lecture Notes in Computer Science (LNCS)*, pages pp. 540–549. Springer.

[Marin et al., 2011] Marin, L., Jara, A. J., and Skarmeta, A. F. (2011). Shifting primes: Extension of pseudo-mersenne primes to optimize ecc for msp430-based future internet of things devices. In *International Cross-Domain Conference on Availability, Reliability and Security - Multidisciplinary Research and Practice for Business, Enterprise and Health Information Systems*. Springer, LNCS.

[Marin-Lopez et al., 2012] Marin-Lopez, R., Pereniguez-Garcia, F., Gomez-Skarmeta, A. F., and Ohba, Y. (2012). Network access security for the internet: protocol for carrying authentication for network access. *IEEE Communications Magazine*, 50(3):pp. 84–92.

[Nie et al., 2011] Nie, P., Vähä-Herttua, J., Aura, T., and Gurtov, A. (2011). Performance analysis of hip diet exchange for wsn security establishment. In *Proceedings of the 7th ACM symposium on QoS and security for wireless and mobile networks*, pages pp. 51–56. ACM.

[Pedrinaci and Domingue, 2010] Pedrinaci, C. and Domingue, J. (2010). Toward the next wave of services: linked services for the web of data. *Journal of Universal Computer Science*, 16(13):1694–1719.

[Raza et al., 2012a] Raza, S., Duquennoy, S., Höglund, J., Roedig, U., and Voigt, T. (2012a). Secure communication for the internet of thingsa comparison of link-layer security and ipsec for 6lowpan. *Security and Communication Networks*.

[Raza et al., 2012b] Raza, S., Voigt, T., and Jutvik, V. (2012b). Lightweight ikev2: A key management solution for both the compressed ipsec and the ieee 802.15. 4 security. In *Proceedings of the IETF Workshop on Smart Object Security*.

[Rescorla and Modadugu, 2012] Rescorla, E. and Modadugu, N. (2012). Rfc 6347: Datagram transport layer security version 1.2. *IETF*.

[Shelby et al., 2013] Shelby, Z., Hartke, K., and Bormann, C. (2013). Constrained application protocol (coap). `http://tools.ietf.org/html/draft-ietf-core-coap-14`.

[Sturek, 2009] Sturek, D. (2009). Zigbee ip stack overview. ZigBee Alliance.

[Vaidya et al., 2011] Vaidya, B., Denko, M. K., and Rodrigues, J. J. P. C. (2011). Security mechanism for voice over multipath mobile *ad hoc* networks. *Wireless Communications and Mobile Computing*, 11(2):196–210.

[Zhang et al., 2012] Zhang, D., Ning, H., Xu, K. S., Lin, F., and Yang, L. T. (2012). Internet of things j. ucs special issue. *Journal of Universal Computer Science*, 18(9):1069–1071.

[Zhou et al., 2010] Zhou, L., Chen, M., Yu, Z., Rodrigues, J. J. P. C., and Chao, H.-C. (2010). Cross-layer wireless video adaptation: Tradeoff between distortion and delay. *Computer Communications*, 33(14):1615–1622.

[Zia and Zomaya, 2011] Zia, T. A. and Zomaya, A. (2011). A lightweight security framework for wireless sensor networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 2(3):53–73.

[Zoran et al., 2007] Zoran, Mari, O., and Dragan, G. (2007). Internet payment system: A new payment system for internet transactions. *Journal of Universal Computer Science*, 13(4):479–503.

[Zorzi et al., 2010] Zorzi, M., Gluhak, A., Lange, S., and Bassi, A. (2010). From today's intranet of things to a future internet of things: a wireless-and mobility-related view. *IEEE Wireless Communications*, Vol. 17(no. 6):pp. 44–51.