On Power-Analysis Resistant Hardware Implementations of ECC-Based Cryptosystems

Roman Willi
IMES Institut für Mikroelektronik und Embedded Systems
HSR Hochschule für Technik Rapperswil
8640 Rapperswil, Switzerland
roman.willi@hsr.ch

Andreas Curiger
Securosys SA
Technoparkstrasse 1
8005 Zürich, Switzerland
curiger@securosys.ch

Paul Zbinden
IMES Institut für Mikroelektronik und Embedded Systems
HSR Hochschule für Technik Rapperswil
8640 Rapperswil, Switzerland
paul.zbinden@hsr.ch

***Abstract*—Power-analysis (PA) based side-channel attacks are effective methods to attack RSA encryption systems and elliptic-curve cryptography (ECC). In this paper, we describe PA-based side-channel attacks aiming to extract the (randomly chosen) private key for an ECC-based cryptosystem in detail. We assume that for the cryptosystem to be attacked the private key will not be available for more than one basic operation. Hence, statistical methods, commonly applied in differential power analysis attacks to enhance the signal-to-noise ratio (SNR), may not be applied. To reach the required SNR for a successful attack, we have extended the analysis by frequency-selective filtering followed by data fragmentation and correlation. We show that the implementation of a "double-and-add-always" scheme for ECC point multiplication, which according to literature has been considered safe against simple PA, will not resist our analytical attack method. We argue that memory accesses are the root cause for a successful attack, and propose an extension of the double-and-add-always scheme to harden ECC hardware implementations adequately.**

Keywords— Power Analysis, SPA, DPA, side channel, ECC Cryptosystem, ECDSA, FPGA, Balancing power consumption

I. Introduction

Dedicated hardware, like application-specific integrated circuits or field-programmable gate arrays (FPGA), are used to offer adequate protection to cryptographic algorithms by keeping the secret keys inside physically isolated devices. Since the advent of side-channel attacks, however, protocol designers have to consider additional measures for a robust hardware implementation. Among the many different side-channel attacks discovered so far, power-analysis (PA) based methods turned out to be especially effective to attack a variety of well-known crypto algorithms, like the RSA cryptosystem and elliptic-curve cryptography (ECC) based systems.

Power-analysis based attacks take advantage of the physical structure of semiconductor devices. Whenever a logic gate is changing its state from zero to one or vice-versa, the parasitic capacitors of this gate will have to be charged or discharged, respectively. This change in electric charge leads to a flow of electric current and energy consumption and induces electrical and magnetic fields. For instance, if the value 1001 is stored to a register, setting this register to a different value will cause more current to flow than setting it to the same value again.

These differences may be observed in the power consumption of an FPGA. Although tiny differences in energy consumption may not be observable directly, they will show up in the signal statistics. Whenever an electronic circuit is evaluating a cryptographic algorithm, changes caused by individual bits of the involved secret key will be detected by a successful PA attack and ultimately reveal the value of the secret key.

This paper is organized as follows: A description on how PA may successfully be applied to attack ECC-based schemes is given in the next section. Section III discusses the measurement setup, data analysis and the results of our adapted PA attack. A modified double-and-add algorithm with improved symmetry and its robustness against our PA attack will be presented in section IV. Section V will finally conclude this paper.

II. Background Information

A. Side Channel Attacks

Classical examples for side-channels include the execution time of an implementation [1], the power consumption of a chip [2] and its electromagnetic radiation [3]. More exotic examples include acoustics [4], temperature [5] and light emission [6]. Some side-channels can be observed only by means of an invasive attack, where the computing device is opened. Others can be observed in a passive attack, where the device is not damaged [7]. Our PA attack described is not a destructive method, however, requires adjustments of the printed circuit board (PCB).

B. Power Analysis

In order to perform a PA attack, energy consumption of an electronic circuit performing computations with the secret key involved has to be recorded. In the first step of the PA attack, a trace is (or several traces are) going to be recorded. In the context of the PA attack, a trace is the recording of a voltage or current measurement over a certain time span. A trace represents the power consumption of the observed electronic circuit during the time of measurement. To record the traces, we utilize a differential voltage probe[1] and apply it across a shunt resistor. More details can be found in section III. The traces will be used as input for (off-line) post-processing. They reveal specific information about distinct operations performed during

978-1-5090-2817-7/16 $31.00 © 2016 IEEE
DOI 10.1109/DSD.2016.59

computation and, if certain conditions are met, finally reveal the secret key. A good overview of many possible PA attacks and their countermeasures may be found in [8].

C. Countermeasures against Side-Channel Attacks

Countermeasures against side-channel attacks may be split into two classes. The goal of the first class is to eliminate or to minimize the leakage of information. This is achieved by reducing the SNR of the side-channel signals. An example is described in [9]. The second class uses strategies to ensure that the information leaking through side-channels cannot be exploited to recover the key. Examples of this approach are described in [10, 9] and [11], respectively. The goal of our implementation is to reduce the SNR and hence belongs to the first class.

D. Ellipitc-Curve Cryptosystems

Since the independent discovery by Miller [12] and Koblitz [13] that elliptic curves may be applied to cryptography, elliptic-curve cryptography (ECC) has been constantly developed further and international standards have been established, like ECDH for key exchange or ECDSA for digital signatures [14]. The security of ECC algorithms is generally based on the discrete logarithm problem (ECDLP), which seems to be hard to solve. As such, the most appealing feature of ECC-based algorithms is the relatively short key length involved [7, 15]. Nevertheless, ECC-based algorithms require high computational efforts. In many applications, parts of the calculations are therefore outsourced to dedicated hardware, specifically to FPGAs [16]. The most computing-intensive part of ECC is the elliptic-curve point multiplication (ECPM) [17]. ECPM is not a multiplication in the sense of multiplication of say two integers. It is in fact the group operation defined over the finite field of points on a specific elliptic curve and involves the "multiplication" of such a point by a scalar value, which usually is the secret key. It hence comes not for a surprise that ECPM is the operation to be observed for PA [18].

In our work we consider elliptic curves over a prime field $GF(p)$ that are defined by the short Weierstrass equation (Eq. 1): An elliptic curve in affine coordinates is the set of solutions of the equation

$$y^2 = x^3 + ax + b \ (mod\ p) \qquad (1)$$

The parameters a, b and the field p specify the curve. The variables x, y, a, and b are all integers between 0 and $p-1$. ECPM includes point doubling (PDBL) and point addition (PADD) sub-operations. In order to calculate ECPM (i.e., to find point $Q = u.P_0$ with $P, Q \in GF(p)$), we have $u = u_{l-1}2^{l-1} + u_{l-2}2^{l-2} + \cdots + u_1 2 + u_0$, with $u_{l-1} = 1$, i.e. the most significant bit is always set. For our attack, we shall implement in the following ECPM in this way:

E. Double-and-Add-Always Algorithm (Algorithm 1)

The "double-and-add-always" scheme from [10] is depicted in Algorithm 1. According to [10], it protects against basic SPA attacks. However, an extension of the SPA attack will allow recovering u with the recording of just one trace.

[1] Note that the differential voltage probe may be replaced by a magnetic-field probe. As shown in [3], such a magnetic field probe might record the energy consumption of an FPGA contactless.

Algorithm 1 Double-and-add-always (MSB first)

Require:
P_0 : Point on Curve
u : integer $u = u_{l-1,\ldots,0}$ and $u_{l-1} = 1$
Ensure: $Q_0 = u.P_0$
$Q_0 \leftarrow P_0$
for $i \leftarrow l-2$ **down to** 0 **do**
  $Q_0 \leftarrow 2.Q_0$ // point doubling (PDBL)
  $Q_1 \leftarrow Q_0 + P_0$ // point addition (PADD)
  $Q_0 \leftarrow Q_{u_i}$ // storage process (STORP)
end for
return Q_0

A closer look at Algorithm 1 uncovers some asymmetry leading to varying energy consumption. In each round of the for-loop, PDBL and PADD are executed once. In register Q_0, the result of PDBL is stored, whereas in register Q_1, the result of PADD is stored. Depending on the value of u_i, Q_0 gets updated with Q_1 in case $u_i = 1$; else if $u_i = 0$, Q_0 would just hold its value. This minimal asymmetry during storing might be exploited.

III. Attacking Algorithm 1

Our PA attack is a mixture between simple power analysis (SPA) and differential power analysis (DPA). As common with SPA a single power trace is analyzed. However, the original SPA fails at the double-and-add-always Algorithm 1. The very small variations in the power consumption of a single measurement are well below the signal noise. Therefore, a band-pass filter is applied to increase the SNR. In addition, our single power trace variant is supplemented with a statistical (DPA) method.

We shall apply the following strategy to successfully attack the double-and-add-always scheme:

A. Single ECPM current measurement
B. Filtering of the measurement
C. Fragmentation of the measurement into an array of partial currents
D. Creation of a golden reference
E. Correlation of the partial currents and the golden reference
F. Forming of correlation groups

A. Measurement Setup

It must be ensured that the measurement takes place in a low-noise environment. This is particularly important, because during this power attack for ECPM only one single trace can be measured. No series of measurements with subsequent averaging can be made, as is common for improving the SNR, because the secret u may change with every new calculation, [14]. A measurement setup as illustrated in Fig. 1 with a shunt resistor $R_{shunt} = 0.5\,\Omega$ and a decoupling capacitor $C_{L1} = 1.5\,\mu F$ leds to the best results.

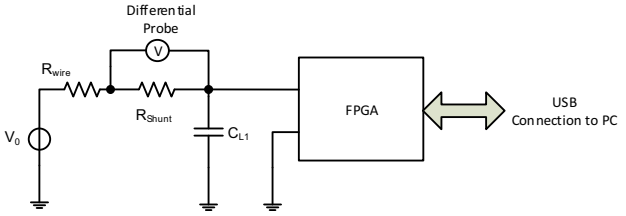

Fig. 1. Block diagram of the measurement setup for measuring the current consumption of an elliptic curve point multiplication on an FPGA.

For our measurements, a Basys3 FPGA Board from Digilent with a Xilinx Artix-7 FPGA (XC7A35T-ICPG236C) was used [19]. Tests were carried out with our proprietary implementation of ECC NIST P-192 ECPM [20]. The overall logic of the FPGA, which is needed for the encryption, is powered with the supply V_0. The wire resistance R_{wire} from the supply V_0 to the FPGA is approximately 0.5 Ω. The decoupling capacitor C_{L1} on the FPGA board is used to support the supply and for smoothing out interference. The point multiplication is composed of the PDBL, which needs 440 clock cycles, and the PADD, which takes 558 clock cycles [21]. The remaining 14 clock cycles are used for storing the result (storage process). To perform one point multiplication, 192 times one PDBL, one PADD and one storage process (STORP) is required. Running at the clock frequency $f_0 = 100\ MHz$, a point multiplication requires a computation time t_1 as follows:

$$t_1 = \frac{1}{f_0} \cdot 192 \cdot (440 + 558 + 14) = 1.94\ ms \qquad (2)$$

One "for loop" (see Algorithm 1) run includes one PDBL, one PADD and the STORP, and requires a calculation time t_2

$$t_2 \approx \frac{1}{f_0}(440 + 558 + 14) = 10.1\ \mu s \qquad (3)$$

Therefore, the calculation repetition frequency f_2 is

$$f_2 = \frac{1}{t_2} = 99\ kHz \qquad (4)$$

For our measurements, a 12 bit 2GS/s oscilloscope HRO66Zi from LeCroy was used. Subsequently, the measured values were processed on the PC.

B. Band-Pass Filter

The measured trace is superimposed by noise and interference. The objective of filtering is to accentuate the information of the STORP and simultaneously to attenuate the signal interference. The band-pass filter attenuates noise components, low-frequency trends as well as unwanted interference of the FPGA and the environment.

The amplitude spectral density, before and after filtering, is represented in Fig. 2. It is apparent that by filtering, the interference in the high frequency range can be greatly suppressed. The filter reduces the noise components and thereby replaces, at least partially, the impossible averaging of measured values.

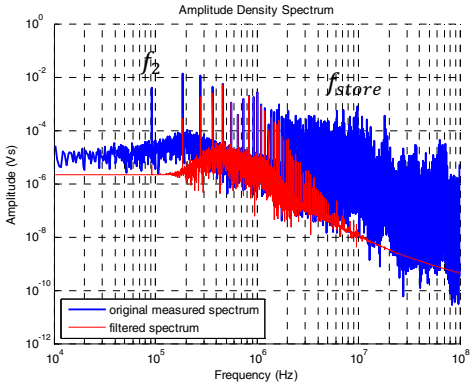

Fig. 2. Measured and filtered amplitude density spectrum

The stopband- and passband frequencies of the filter can be estimated based on the calculation frequency f_2 and the storage frequency f_{store}. As mentioned before, the STORP requires 14 clock cycles resulting in the storage frequency f_{store}

$$f_{store} = \frac{1}{f_0} \cdot 14 = 7.14 MHz \qquad (5)$$

C. Fragmentation

After filtering, the voltage waveform will be fragmented (see Fig. 3). It forms partial voltage waveforms (part-of-trace or subsets) corresponding to one "for loop" cycle of Algorithm 1.

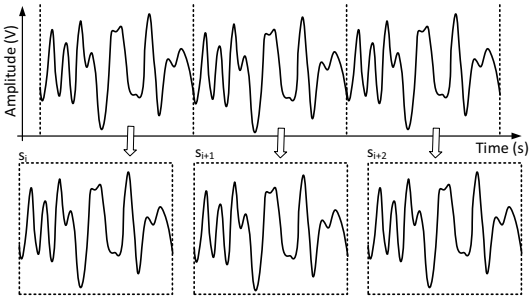

Fig. 3. Fragmentation of the filtered voltage waveform in $l - 2$ subsets s_i

The voltage waveform s_i corresponds to a PDBL A, a PADD B and the STORP C. In the following Fig. 4, this waveform is depicted qualitatively. The STORP C, compared to A and B, runs for a short time only.

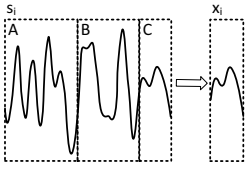

Fig. 4. The voltage waveform s_i corresponds to a PDBL A, a PADD B and the STORP C. Part C is mapped to subset x_i and used for further processing. Parts A and B are discarded.

PDBL and PADD, depending on the value being calculated, have different levels of power consumption. These different levels, however, are not of interest for the subsequent calculations. Only the power consumption of the STORP in part C is of interest. Therefore, part C will be cropped and parts A and B will be discarded. The voltage curve x_i thus contains only a small part of the total calculated length of the "for loop" cycle. This has the advantage that unwanted interference and noise components of the PDBL and the PADD in the subsequent correlation have a lesser impact.

D. Golden Reference

An average is calculated over all subsets x_i, which will further be used as golden reference $\bar{x}$. Assuming that the secret u includes about the same number of 1 as 0, the golden reference $\bar{x}$ forms an average across the influences of all the varying subsets x_i.

E. Cross-Correlation

The golden reference will now be cross-correlated with all subsets x_i. The cross-correlation indicates the degree of similarity in function of the time delay τ, and is calculated by the convolution of the functions x_i and $\bar{x}$ (see Eq. 6):

$$c_i = (\bar{x} \star x_i) \text{ for all } i = l - 2 \text{ down to } 0 \qquad (6)$$

Fig. 5a shows the cross-correlation of the subsets x_i with the golden reference $\bar{x}$. The autocorrelation of the golden reference $\bar{x} \star \bar{x}$ is colored in green:

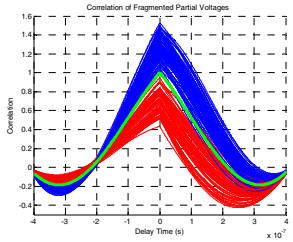

Fig. 5a. Correlation of the fragmented partial voltages with the golden reference

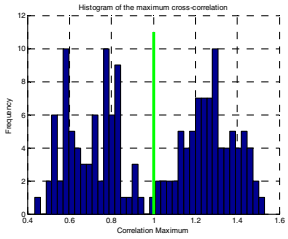

Fig. 5b. Histogram of the maximum cross-correlation

There are two groups of correlations formed, as can be seen in Fig. 5a. In one group the correlation maximum $max(c_i)$ is greater than the correlation maximum of the autocorrelation of the golden reference $max(\bar{x} \star \bar{x})$ (the blue curves in Fig. 5a). In the other group the correlation maximum is smaller (the red curves in Fig. 5a).

F. Correlation Groups

The correlation maxima illustrated in the histogram (see Fig. 5b) shows two approximate Gauss-distributed groups. The maximum of the autocorrelation of the golden reference $max(\bar{x} \star \bar{x})$ forms the boundary of these two groups (indicated by the green mark in Fig. 5b). If the calculated correlation maximum $max(c_i)$ from the subset x_i is greater than the maximum of the autocorrelation of the golden reference $max(\bar{x} \star \bar{x})$, it indicates that the u_i based on this x_i is zero (see Eq. 7).

$$\text{If } max(c_i) \geq max(\bar{x} \star \bar{x}) \text{ then } u_i = 0 \text{ else } u_i = 1 \qquad (7)$$

With this approach, the secret u for the ECC Algorithm 1 can be accurately identified now.

IV. Improving Algorithm 1

In the previous section III we have shown that solely due to the asymmetry in the STORP of Algorithm 1, the secret u can be determined. To prevent this, it is important when storing the result that the same power consumption will occur, whether $u_i = 1$ or $u_i = 0$. In order to achieve this, the following Algorithm 2 is proposed:

Algorithm 2 Double-and-add-always-improved-symmetry (MSB first)

Require:
P_0 : Point on Curve
u : integer $u = u_{l-1,\ldots,0}$ and $u_{l-1} = 1$
Ensure: $Q_0 = u.P_0$
$Q_0 \leftarrow P_0$
for $i \leftarrow l - 2$ **down to** 0 **do**
  $Q_0 \leftarrow 2.Q_0$ // point doubling (PDBL)
  $Q_1 \leftarrow Q_0 + P_0$ // point addition (PADD)
  if $u_i = 1$ **then**
    $Q_0 \leftarrow Q_1$ // storage process (STORP)
  else
    $Q_1 \leftarrow Q_0$ // storage process (STORP)
end for
return Q_0

The STORP $Q_1 \leftarrow Q_0$ at $u_i = 0$ is, from a purely functional standpoint, considered unnecessary. It is only important for the improved symmetry. The recommendation is that during the STORP the result is always stored: Either the result of the PADD $Q_1 \leftarrow Q_0 + P_0$ is stored in Q_0, or the result of the PDBL $Q_0 \leftarrow 2.Q_0$ is stored in Q_1. Since the same registers are addressed in both STORP, the wire length is almost identical, leading to similar parasitic capacitances. On average, the same number of bits will change when storing $Q_0 \leftarrow Q_1$ or $Q_1 \leftarrow Q_0$. If this were not the case, a PA attack could be successful.

Our improved algorithm requires much less resources than, say, "blind basepoint P ", "randomized projective coordinates" or "blind the multiplier u", see [10] and [11]. Furthermore, our modifications remain exclusively at an algorithmic level in contrast to, for instance, the logic level design methodology described in [9].

A. Measurements on Hardware using Algorithm 2

The following measurement and evaluation of the Algorithm 2 shows that by the symmetry improvement the secret u can no longer be determined. The correlation maximum

illustrated in histogram Fig. 6b no longer shows two distinct Gaussian - type groups as in Fig. 5b.

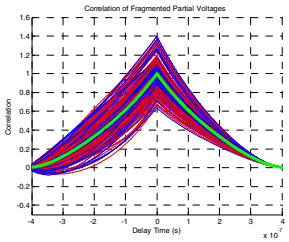

Fig. 6a. Correlation of the fragmented partial voltages with the golden reference

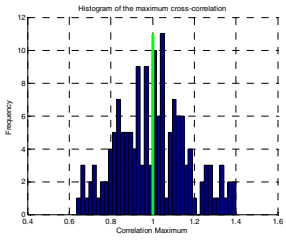

Fig. 6b. Histogram of the maximum cross-correlation

In addition, it can be shown that Algorithm 2 is secure against timing attacks, as described in [1]. This is because each computation step requires the same amount of time and the STORP always occurs at the same Δt. If this were not the case, there would be a shift in the delay time of the correlation maximum on the x-axis of Fig. 6a.

V. Conclusions

In this paper, a PA attack has been presented, which is able to extract the private key of cryptographic ECPM implementations from marginal asymmetric details in the well-known double-and-all-always scheme. By aid of dedicated frequency selective filtering followed by data fragmenting and correlation, the attack is able to increase the SNR of the leakage information such that asymmetric memory access within a XILINX Series-7 FPGA implementation can be detected with a single measurement.

In order to protect ECC implementations against such PA attacks, only minor modifications to the classical algorithm and its implementation need to be introduced. A double-and-add-always algorithm with improved symmetry has been presented, which makes it impossible to exploit statistical data of memory accesses. The algorithm adds very little hardware overhead. It has been shown that after introducing the counter measures, our FPGA implementation of ECC algorithms can no longer be decrypted by timing- and PA attacks as described in this paper.

Acknowledgment

This work was supported by the Commission for Technology and Innovation CTI. The CTI is the federal innovation promotion agency responsible for encouraging science-based innovation in Switzerland.

References

[1] P. C. Kocher, “Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems,” *CRYPTO '96*, vol. LNCS 1109, pp. 104–113, 1996.

[2] P. C. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis,” *CRYPTO' 99*, vol. LNCS 1666, pp. 388–397, 1999.

[3] E. De Mulder, P. Buysschaert, S. B. Ors, P. Delmotte, B. Preneel, G. Vandenbosch, and I. Verbauwhede, “Electromagnetic Analysis Attack on an FPGA Implementation of an Elliptic Curve Cryptosystem,” *EUROCON'05*, vol. 2, pp. 1879–1882, 2005.

[4] D. Genkin, A. Shamir, and E. Tromer, “RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis,” *CRYPTO'14*, vol. LNCS 8616, pp. 444–461, 2014.

[5] M. Hutter and J. M. Schmidt, “The temperature side channel and heating fault attacks,” *CARDIS'13*, vol. LNCS 8419, pp. 219–235, 2014.

[6] J. Di Battista, P. Perdu, J. C. Courrege, B. Rouzeyre, and L. Torres, “Validation of differential light emission analysis on FPGA,” *SCS'09*, vol. 3, pp. 1–5, 2009.

[7] Nigel P. Smart, “Algorithms , Key Sizes and Parameters Report,” *European Union Agency for Network and Information Security*, pp. 0–95, 2014.

[8] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, “Introduction to differential power analysis,” *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011.

[9] K. Tiri and I. Verbauwhede, “A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation,” *Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, pp. 246–251, 2004.

[10] J.-S. Coron, “Resistance against differential power analysis for elliptic curve cryptosystems,” *CHES'99*, vol. 1717, pp. 292–302, 1999.

[11] M. Joye and C. Tymen, “Protections against Differential Analysis for ECC,” *CHES'01*, vol. LNCS 2162, pp. 377–390, 2001.

[12] V. Miller, “Use of Elliptic Curves in Cryptography,” *CRYPTO'85*, vol. LNCS 218, pp. 417–426, 1986.

[13] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[14] D. Johnson, A. Menezes, and S. Vanstone, “The Elliptic Curve Digital Signature Algorithm (ECDSA),” 2001.

[15] L. Julio and R. Dahab, “An Overview of Elliptic Curve Cryptography,” *TECHREPORT Institute of Computing, Sate University of Campinas*, 2000.

[16] K. C. C. Loi, S. An, and S. Ko, “FPGA Implementation of Low Latency Scalable Elliptic Curve Cryptosystem Processor in GF (2 m),” *ISCAS'14*, pp. 822–825, 2014.

[17] J. Vliegen, N. Mentens, J. Genoe, A. Braeken, S. Kubera, A. Touhafi, and I. Verbauwhede, “A compact FPGA-based architecture for elliptic curve cryptography over prime fields,” in *ASAP'10*, 2010, vol. 21, pp. 313–316.

[18] T. Izu, B. Möller, and T. Takagi, “Fast elliptic curve multiplications resistant against side channel attacks,” *INDOCRYPT'02*, vol. LNCS 2551, no. 1, pp. 296–313, 2005.

[19] Digilent Inc., “Basys3 FPGA Board Reference Manual,” *502-183*, 2014. [Online]. Available: https://reference.digilentinc.com/_media/basys3:basys3_rm.pdf.

[20] NIST, “Digital Signature Standard (Dss),” *Federal Information Processing Standards Publication 186-4*, 2013.

[21] D. Amiet, A. Curiger, and P. Zbinden, “Flexible FPGA-Based Architectures for Curve Point Multiplication over GF (p),” *Euromicro Conference on Digital System Design — DSD 2016*, 2016.