Security-Driven Exploration of Cryptography in DSP Cores

Catherine H. Gebotys
Department of Electrical and
Computer Engineering,
University of Waterloo
Waterloo,Ont N2L3G1 Canada
cgebotys@optimal.vlsi.uwaterloo.ca

ABSTRACT

With the popularity of wireless communication devices a new important dimension of embedded systems design has arisen, that of security. This paper presents for the first time design exploration for secure implementation of cryptographic applications on a complex DSP processor core. A new metric for security, the implementation security index, is introduced for measuring resistance to power attacks. Elliptic curve cryptographic algorithms are used to demonstrate and quantize security, energy, performance and code size tradeoffs. Modification of power traces is performed to maximize security against power attacks which has significant savings in energy dissipation compared to an existing mathematical approach. This research is important for industry since efficient yet secure cryptography is crucial for wireless communication embedded system devices.

Categories and Subject Descriptors

E.3 [**Data**]: Data Encryption – *public key cryptosystems, standards.*

General Terms

Security, Design, Performance.

Keywords

Power Analysis Attack, Low Energy, DSP, Methodology

1. INTRODUCTION

Unlike current design methodologies for embedded systems which concentrate on high performance, low cost, low power and low energy, security is increasingly becoming important. Design for security involves secure protocol implementation and power analysis in addition to algorithm design. In fact power dissipation has a large impact on security as well as cost, and reliability of an embedded system. Not only must cryptographic algorithms be high performance and low energy, but more importantly they must be secure or safe from side channel attacks. A side channel attack involves obtaining useful information from the cryptographic application which may lead to the revelation of the secret key. Useful information includes the amount of time it takes to perform various operations or the variation of power dissipation during key computations. In the later case this is known as a power attack[1]. The power attack is more difficult and challenging to avoid. As an example, an attacker who has obtained the secret key is able to eavesdrop on a confidential wireless communication between two parties. Consider Alice, the victim, who wishes to encrypt a conversation with Bob. She first has to set up a session key. Alice computes the session key as: xP in elliptic curve cryptography (or $y^x \ mod \ n$ in RSA technology) where x is the secret key and this computation is performed various times for different P's (or y's). The attacker may know P (or y) and in addition may obtain the computations times or may be able to monitor power dissipation. With this additional information the attacker may eventually be able to compute the secret key, x [1]. When the attacker has obtained the secret key, communication between Alice and Bob is not secure. Alternatively, obtaining the secret key in other applications, such as smart cards, allows one to illegally use phone or digital TV services. In power attacks, the dynamic power of the device is measured over time (for example a smart card which has been inserted into a fake banking machine or the attacker has temporarily obtained the portable device with the embedded secret key). In elliptic curve cryptography (ECC), the analysis of power may reveal when a point doubling occurs (or calculation of $2P$), and when two points are being added (such as $2P+P$) in the computation of xP, thus revealing the secret key. In SOC platforms, core processors often run at different voltages, and use separate power pins, thus secure implementations of cryptography are important to discourage any power attacks. Furthermore these complex chips are often composed of network processors in addition to many DSP cores, thus cryptography on DSP cores is an important lower cost alternative to encryption cores. This paper for the first time presents a new metric for quantizing security and design exploration for ECC on a DSP processor core. Results are based upon real measured dynamic power. Additionally tradeoffs in code size, performance, and energy dissipation for security are explored.

ISSS'02, October 2–4, 2002, Kyoto, Japan.

Previous literature has discussed performance and energy dissipation at length[10,12], however only recently with the widespread use of wireless communications has interest in the security of embedded system designs increased. Typically embedded systems involve the design of low power, low cost, high performance algorithms supporting wireless communication systems with audio and video types of data. However keys used to encrypt various types of data must be secure. For example cryptography provides techniques for authentication (verification that the receiver of your transmitted data is who you want it to be and not an imposter) and for generating sessions keys. Currently research in power attacks of smart cards, have utilized general purpose processors with low clock frequencies (~100MHz) [1]. Typically smart card applications are not time critical and energy dissipation is not a major concern since power is attained from the card reader (or ATM machine, etc)[5]. Power attacks of more sophisticated processors with parallel instruction execution have not been reported in the literature. The measurement of power while a processor is executing an application (or a power trace) has been used in power-attacks of cryptographic devices, such as smart cards[1]. In particular the analysis of the variation of power, and computations on a number of power traces can be used to detect data and algorithmic dependencies[1]. This research studied the correlation of power variation with data values being manipulated and instruction sequencing. In the former case, known as differential power attacks(DPA), encryption applications were analyzed[1]. In DPA, including the extension for elliptic curve cryptography[15], it is assumed that for each power trace the same instruction is executing at the same instance of time. Thus one defense against DPA is to insert random time shifts to decorrrelate the output power trace. A defense against another power analysis attack, known as simple power attack (SPA), was to introduce random sequencing of instructions again to reduce correlations. Elliptic curve cryptography is believed to be more complex yet efficient[7,6] compared to RSA technology and is used in many smart cards and other devices. For example a 1024 bit key in RSA is equivalent to 160 bit key in ECC[5]. Researchers addressing smart card application have suggested security against power attacks be achieved through 70% increase in computational cost. This is achieved through using different forms of the curve, such as the Jacobi form, where mathematically the sum and the double of a point use the same formula[3,8]. The study of DSP processors in cryptography has been limited, however their resistance to power attacks has not been addressed. Recent low power research work in embedded systems design is directed towards minimizing energy dissipation[9,10] and power modeling[11,12]. Researchers have also investigated power traces to obtain exact timing information of applications running on processors [2]. Modifying power traces to save energy dissipation, known as dynamic power management, has been investigated by a number of researchers[9,10]. However modifying power traces of applications for implementation-security on DSP processor cores has not previously been studied.

This paper will present a new metric for security against power attacks, the implementation security index (*ISI*), and design exploration of performance, energy dissipation, code size in addition to security. It will be demonstrated on a complex VLIW DSP processor core, the Star*core (SC140), developed by Motorola and Lucent[4]. An elliptic curve cryptographic application is designed to be resistant to power-attacks yet low energy dissipation, high performance, and small code size objectives are supported. Security from power-attacks is verified with real current measurements of the DSP hardware VLSI core in a chip. The design and implementation of this cryptographic algorithm will be discussed with emphasis on the implementation security index.

2. METHODOLOGY

This section will introduce the application, elliptic curve cryptography on prime fields and introduce the methodology used to develop a security index for measuring resistance to power attacks. Prime field ECC was chosen over binary field ECC since it was more suitable for DSP processor implementation. Prime field cryptography involves a significant number of integer multiplies which can be performed very efficiently on DSP cores, however binary field cryptography relies more heavily on exclusive or operations. Since cryptographic key sizes are much larger than processor word sizes (32 or 40 bit registers in SC140's datapath) multi-precision algorithms are used to perform the arithmetic. In addition to a chosen key length, there are many different fields, projective coordinates, and types of elliptic curves that can be implemented. For added security portable devices should be able to support numerous curves and fields. However it is important for the designer to be able to choose which sets to implement on an embedded device, to tradeoff performance, code size, energy dissipation, and security against power attacks. The application, point multiplication, will be introduced in this section, followed by a discussion of implementation methods.

For this research, point multiplication was explored using different design implementations of Jacobi projective coordinates[8] on the standard Weierstrass equation of the curve and the Jacobi form of the elliptic curve[8,3]. Point multiplication (or computation of xP where P and xP are both points on the elliptic curve and x is the secret key) is used in session key generation, signature generation, signature verification, etc. In elliptic curve cryptography, there is no multiplication operation, hence the only method for computing xP is by doubling and adding. In general this is why elliptic curves are so secure since the problem of finding *x* given *P* and *xP* is a very hard problem. The algorithm takes *P* and computes *2P*, then *4P*, etc.. each time using a point doubling routine (*double*). The point summation routine (*sum*) adds *kP* to the accumulated sum according to the bits of *x*. For example if *x=$0b* ($ indicates hex) then the non-adjacent form of *x = 1 0 –1 0 –1* so *xP =2* 2*(2*2P -P)-P*. The direct implementation of the Jacobi projective coordinates on the standard curve has a very low value of implemented security. This is illustrated in the power trace shown at the bottom of figure1, where S is the *sum* routine and D is the *double* routine. The power trace at the top performs point multiplication on the same key but is much more secure since one *sum* visually looks like two *double*s.

The modification of *sum* and *double* routines for security against power attacks was explored by inserting redundant operations into the *double* and *sum* routines so that the order of operations was identical. Additionally the higher level algorithms were designed so that the routines in between the *sum* and *double* were also identical (ie. so one could not distinguish *sum* followed by *double* or *double* followed by *sum,* or *double* followed by *double*). Three different implementations were analyzed for the general Jacobi projective coordinates on the standard Weierstrass equation. Additionally the implementation and power traces of the Jacobi Curve (*JC*) was performed for a comparison of security (since the Jacobi form uses the same routine for both *sum* and *double*). Codesize, performance, energy dissipation and power traces were recorded for each implementation. The power traces for four keys were analyzed with various statistical techniques. Finally a metric for quantizing the security of an applications implementation was developed. We call this metric the *ISI* for implementation security index.

3. EXPERIMENTAL SECTION

The power traces for four keys were obtained for the three different design implementations of the Weierstrass curve plus the Jacobi curve (*JC*). The three different implementations were called *2F, IL,* and *WR.* In *2F* the *sum* routine was split into two routines which were called at the higher level allowing insertion of miscellaneous code inbetween calls to *sum*s or *double*s. In *IL* the code was further optimized and fully inlined to again improve performance at the expense of codesize. Finally in *WR* the code was further optimized replacing redundant operations with more secure implementations and detailed assembly modification to ensure exact timing of the *sum* routines and *double* routine. All power traces were obtained by executing the cryptographic algorithms on the SC140 at 100MHz (for illustration purposes), using a pattern generator and high speed oscilloscope to capture the power traces. In the power trace plots the y-axis represents the current variation and the x-axis represents the time. Matlab was used for signal analysis of the power waveforms.

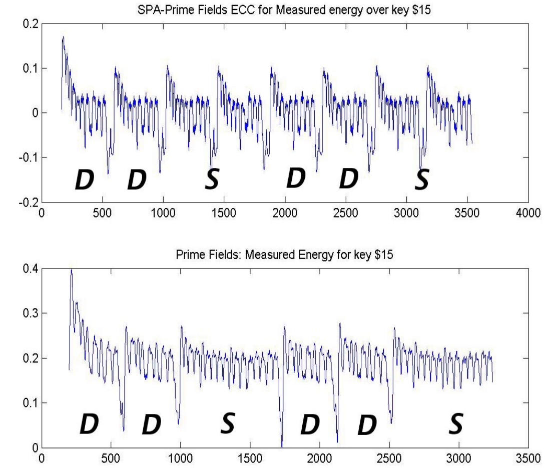

Figure 1. Comparison of power traces of original cryptographic code at bottom with power-attack resistant cryptographic code for the same key.

The *sum* and *double* routines from each power trace was extracted and the variances and the mean plus or minus two times the standard deviation were computed and plotted in figures 2,3 for *2F* and *JC.* The average variances (*3.12E-4* for *2F* and *1.911E-4* for *JC*) provide an initial indication of security. However it is difficult to use it to determine which part of the code needs to be modified to increase security. Note also in these figures that the differences of the sum and double routines shown in the bottom plot of the figures are high where the power traces have the highest slopes. The *IL* code increased code size by 6 times for only a 11% improvement in performance.

The implementation security index was next computed, where $ISI = \left|\frac{1}{t}\right|, t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{(s_1)^2}{n_1} - \frac{(s_2)^2}{n_2}}}$, where the means, standard deviations and number of power subtrace samples are $(\bar{x}_1, \bar{x}_2), (s_1, s_2), (n_1, n_2)$. Here a subtrace sample is a *sum* followed by a *double* (or *double* followed by a *sum*) representing group subscript 1 (or group 2). This metric combines the variance and means, and identifies regions of the power trace where the means differ and variances are low. In these cases the *ISI* is low indicating that this region of the power trace could be used to differentiate a *sum* from a *double*. The *ISI* can be used by the embedded systems designer to identify regions of the power trace (or routine)

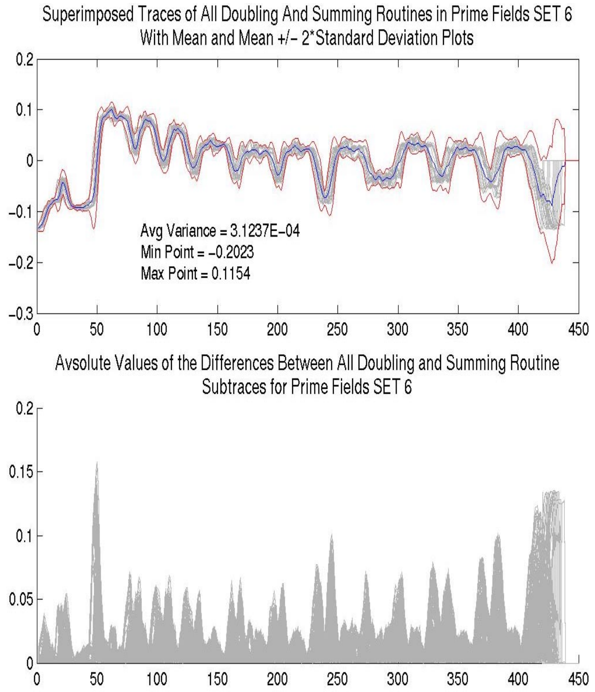

Figure 2. Means, Variances, Differences of *2F*.

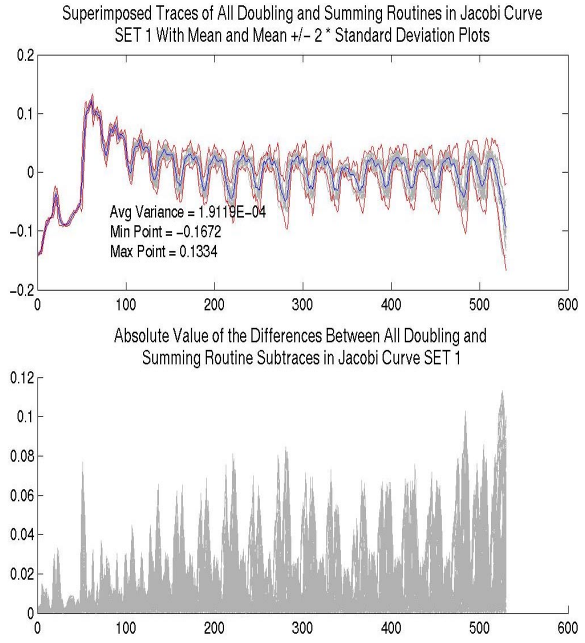

Figure 3. Means, variances, and differences of *JC*.

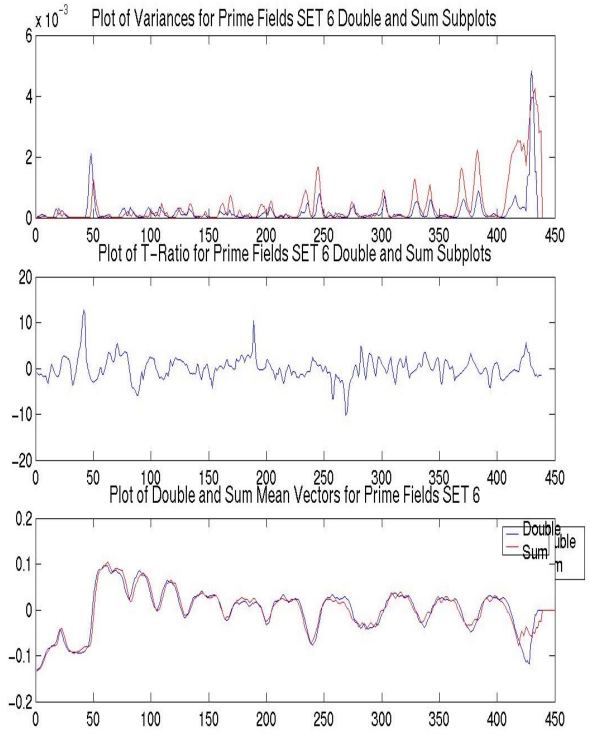

Figure 4. Variance, *t*, means for *2F* power traces.

which have low security. The top plot of figure 1 was code *2F* which although looks secure, is not. For example the middle plot of figure 4 is *t* for *2F* which has a peak near 190 on the x-axis (indicating low security). This peak occurs where a multiplication of a 192bit number by 8 (**8*) is performed in the *double* routine. The corresponding multiplication in the *sum* routine is a multiplication of two full 192bit numbers. Thus the *2F* code was not secure (it has an average *ISI* of *0.295*) and was modified to eliminate this security problem. The modified code called *WR*, removed the **8* and replaced it with a multiply of two full 192bit numbers (whose result is put in a temporary variable), and introduced shifts to accomplish the **8* functionality. The added shifts were also added to the *sum* routines. The *t* for the resulting code, *WR*, is shown in figure 5 and it has improved security with an average *ISI* of *0.49* (indicating higher security than *2F*). The peak in *t* close to 200 in figure 5 (and corresponding power dip close to 260 in figure 4) is due to starting the point multiplication application with z-coordinate of the EC point equal to one (and is eliminated by using a pseudorandom number instead [8]). Figure 6 illustrates *JC* code which has a *t* peak in between the *double* and *sum* routines, thus indicating a security leak in the high level code, which was subsequently fixed. The final *JC* code had a *ISI* of *0.44*. The bar chart in figure 7 from left to right shows the energy dissipation (mJ) of *WR, JC* and clock cycles for *WR, JC* for each key (ie. $b, $10, and a random192bit key). In figure 7 the cycles of the 192bit key have been divided by 10,000 and energy of the 192bit key is divided by 10. The cycles for key $b and $10 have also been divided by 100 in the figure as well for scaling purposes. The average of *ISI* vector (from data in figures 5 and modified *JC* code) of *WR* and *JC* is respectively 0.49 and 0.44. The energy per bit (energy dissipation of 192bit key divided by 192) of the *WR* and *JC* implementations is 10.37 and 14.96 respectively. The code size of *WR* is 9618 bytes compared to *JC* which only requires 7338 bytes (since the *double* and *sum* are the same routine).

4. DISCUSSIONS AND CONCLUSIONS

The methodology presented in this paper, has shown that *ISI* can provide important information for cryptographic applications[14] being implemented by embedded system designers. Previous methods suggested, such as simple power attacks, or differencing can be improved by exploring variances and *ISI*. This design exploration has been used to develop code which has improved security yet lower energy dissipation and higher performance compared to the Jacobi curve implementation. Unlike previous power analysis, such as DPA, ISI has been shown to support a complex processor architecture where each clock cycle involves multiple actives busses and the point multiplication algorithm which includes small timing shifts.

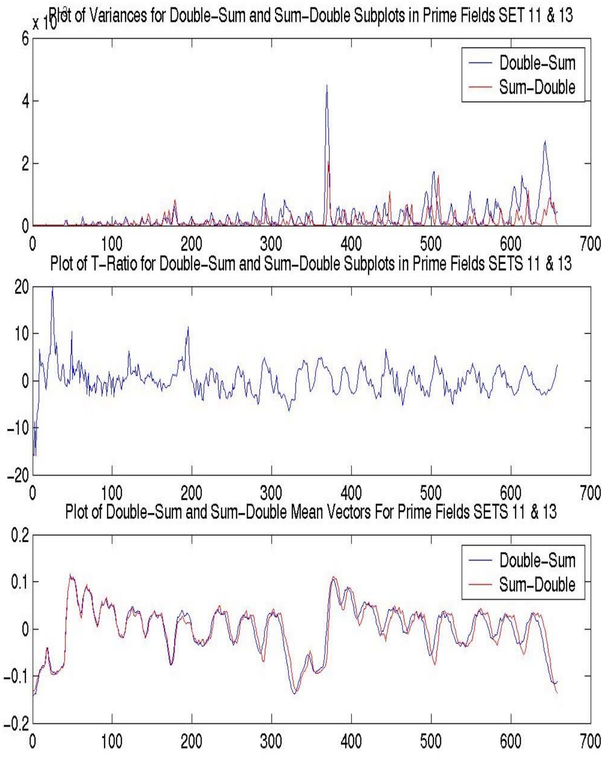

Figure 5. Variances, t, means of *WR* power traces.

Unlike previous research on power management and software techniques to reduce energy dissipation[9,10], this research has examined modification of power and energy dissipation for security. This research provides implementations with lower energy and performance overheads than implementations of researched SPA-resistant algorithms, such as the Jacobi curve, *JC*[3]. The lower energy dissipation will be important for secure implementations in portable devices. Design exploration of verified elliptic curve point multiplication routines running on a complex VLIW DSP processor core is presented. Unlike previous research, a new metric, the implementation security index, *ISI*, has been introduced for quantizing security of implementations. Real power traces have been measured, and security from power-attacks verified with real hardware VLSI chip power measurements. This methodology for the design of secure software for the SC140 DSP processor can in general be applied to other processors. Results show that *WR* code improves energy dissipation, performance, and implementation security index by 1.44 times, 1.44 times, and 1.11 times respectively compared to our implementation of previously research routines, *JC*, with a 31% increase in code size. For the first time a security metric, *ISI*, has been introduced and shown to be a significant aid to embedded system designers to ensure that implementations of applications are secure.

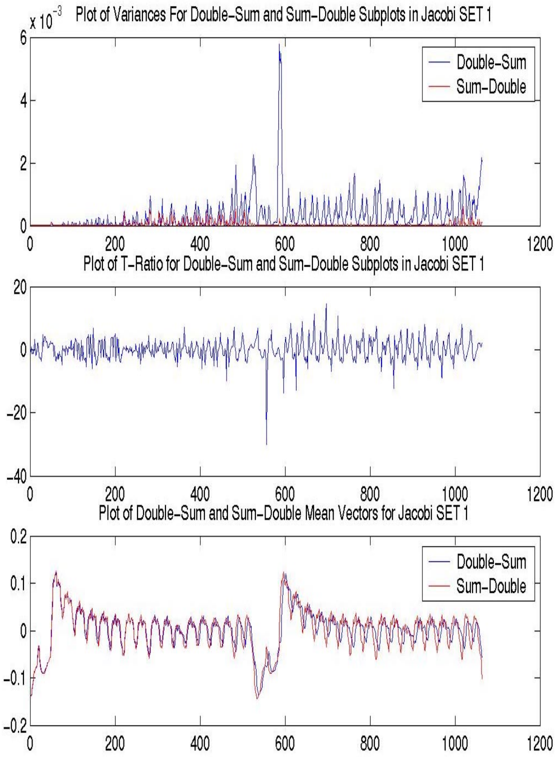

Figure 6. *JC* showing variances, *t*, and means of subtraces of power.

In general this new metric and exploration is applicable to any embedded software which computes with a secret key. This metric can be used for design exploration of security in addition to performance, code size and energy dissipation. This research is crucial for supporting a methodology for designing software that is not only optimized for performance, power and cost, but also for implementation security. For the first time this research introduces a metric for developing secure implementations targeting DSP processors in wireless embedded systems.

5. ACKNOWLEDGMENTS

The author would like to thank NSERC, Motorola, and CITO for their support through funding of this research. The author also would like to thank R.Muresan for his excellent acquisition of the power traces using [13] and A.Sathiananthan for coding the encryption algorithms and running them on the Star*Core board. Additionally the author would like to thank R.Gebotys and T.Tam (Motorola Canada Ltd.) for their helpful discussions (as well as L.Cheng for the inversion algorithm coding).

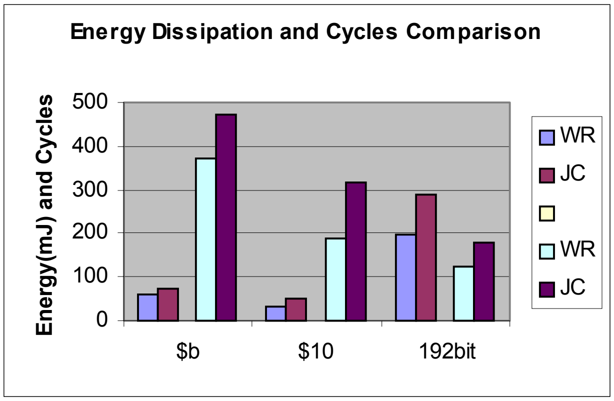

Figure 7. Energy dissipation and cycle comparison of *WR* with *JC.*

6. REFERENCES

[1] P.Kocher, “Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems”, LNCS, 1998.

[2] F.Wolf, J.Kruse, R.Ernst “Segment-wise timing and power measurement in software emulation”, Designers Forum DATE 2000.

[3] P.Liardet, N.Smart “Preventing SPA/DPA in ECC systems using the Jacobi Form”, LNCS 2162, May 2001, pp391-401.

[4] “Star*Core 140 DSP Core Reference Manual”, Motorola/Lucent, Sept 1999.

[5] W.Rankl, W.Effing, Smart Card Handbook, 2nd edtn., Wiley, 2000.

[6] M. Rosing, Implementing Elliptic Curve Cryptography, Manning Publishing, 1999.

[7] D.Hankerson, J.Hernandez, A.Menezes “Software Implementation of Elliptic Curve Cryptography over Binary Fields”, White Paper, www.certicom.com, 2000

[8] Chudnovsky,D.V., G.V.Chudnovsky “Sequences of Numbers generated by addition in formal groups and new primality and factorization tests”, Applied Mathematics, Vol.7, pp385-434,1986.

[9] Y.H.Lu,L.Benini,G.DeMicheli “Requester-Aware Power Reduction” Int’l Symp on Sys. Level Synth., 2000, p18-23.

[10] L.Benini, etal. “Battery-Driven Dynamic Power Management of Portable Systems”, Int’l Symp on Sys Level Synth, 2000, p25-30.

[11] T.Simunic,L.Benini,G.DeMicheli “Source Code Optimization and Profiling of Energy Consumption in Embedded Systems”, Int’l Symp on Sys Level Synth, 2000 p193-198.

[12] V.Tiwari,S.Malik,A.Wolfe “Power Analysis of Embedded Software: A first step towards software power minimization”, IEEE Trans. On VLSI, Vol.2No.4, 1994.

[13] R.Muresan, C.Gebotys, Current consumption dynamics at instruction and program level for a VLIW DSP Processor, Int’l Symp on Syst Level Synth, Oct 2001, pp.130-135.

[14] C.Gebotys, R.Gebotys , Secure Elliptic Curve Implementations: An analysis of resistance to power-attacks in a DSP processor, to appear Workshop on Cryptogr. Hardware and Embd Sys, CHES’02, LNCS 2002.

[15] J.Coron, Resistance against differential power analysis for elliptic curve cryptosystems, CHES 1999.