An Efficient DPA Countermeasure With Randomized Montgomery Operations for DF-ECC Processor

Jen-Wei Lee, Ju-Hung Hsiao, Hsie-Chia Chang, and Chen-Yi Lee

***Abstract*—Nowadays, differential power-analysis (DPA) attacks are a serious threat for cryptographic systems due to the inherent existence of data-dependent power consumption. *Hiding* power consumption of encryption circuit or applying *key-blinded* techniques can increase the security against DPA attacks, but they result in a large overhead for hardware cost, execution time, and energy dissipation. In this brief, a new DPA countermeasure performing all field operations in a *randomized Montgomery domain* is proposed to eliminate the correlation between target and reference power traces. After implemented in 90-nm CMOS process, our protected 521-bit dual-field elliptic curve (EC) cryptographic processor can perform one EC scalar multiplication in 8.08 ms over $GF(p_{521})$ and 4.65 ms over $GF(2^{409})$, respectively, with 4.3% area and 5.2% power overhead. Experiments from a field-programmable gate array evaluation board demonstrate that the private key of unprotected device will be revealed within 10^3 power traces, whereas the same attacks on our proposal cannot successfully extract the key value even after 10^6 measurements.**

***Index Terms*—Dual fields, elliptic curve (EC) cryptography (ECC), power-analysis attacks, security system.**

I. Introduction

ELLIPTIC curve (EC) cryptography (ECC), described in IEEE P1363 [1] and FIPS PUB 186-3 [2], has been widely applied to provide a confident scheme for information exchange. For the past several years, many previous works [3]–[6] have been published for ECC hardware implementation aiming at the performance improvement. However, even if the ECC is secure at cryptanalysis, the private data of an unprotected hardware device can be extracted by physical attacks due to side-channel leakage. The power-analysis attacks, initially presented by Kocher [7], can reveal the key value by analyzing the power information of a cryptographic implementation such as on an application-specified integrated circuit (ASIC), field-programmable gate array (FPGA), or microprocessor.

During the device processing, simple power-analysis (SPA) attacks can distinguish the key value through visual inspection because of the specifically active circuit with direct hardware scheduling. The *double-and-add-always* method [8], [9] is usually used to avoid the variation of power consumption over time. However, the differential power-analysis (DPA) attacks [10] computing the correlation between target power traces and power model can reveal the key value due to the existence of key-dependent operations in every round of calculation.

Hiding technique with algorithm-independent dedicated circuit is a common approach to protect cryptographic processors from attackers collecting the key-dependent characteristics of power traces. In [11], a wave dynamic differential logic circuit with regular routing algorithm is exploited to equalize the current between rising and falling transitions. However, more than 200% overhead in area, performance, and power consumption is added to the unprotected encryption engines. Switched capacitor [12] is able to isolate the encryption core from the external power supplies, but this approach results in 50% speed loss for replenishing charge every cycle. In order to avoid the throughput degradation, a countermeasure circuit using digital controlled ring oscillators [13] is designed outside of the critical path. The concept is to generate random noise power to dominate the power consumption of arithmetic unit, and then, the correlation peak would not be found even matching the correct key value. However, this demands extra 100% power overhead for the key-dependent processing element.

At the algorithm level, *masking* the processed data independent of power consumption is another approach to avoid the DPA attacks. For the ECC schemes, since the scalar of point calculation is periodic with the point order $\#E$, a *key-blinded* technique can be adopted to change the key value by adding $r \cdot \#E$ for every calculation, where r is a random integer. However, with this method, the throughput overhead is inevitable due to extending the key length. In [9], the point calculation of 521-bit key extended with a 32-bit random value needs 10% more execution time to be carried out than that of the unprotected approach.

In this brief, we propose a new efficient countermeasure to overcome the DPA attacks by computing the overall ECC functions in a *randomized Montgomery domain*. The feature of our approach is to mask the intermediate values in not only the arithmetic but also the temporary register. Thus, it is unnecessary to extend the key length, customize the circuit, and modify the routing algorithm in ASIC or FPGA design flow. Since our proposed design adopts simple logic circuit to counteract DPA attacks, the hardware cost overhead could be significantly reduced, and the maximum operating frequency of the protected design is the same as that of the unprotected design using the conventional Montgomery algorithm. In addition, by reducing the iteration time of the division, which dominates other field operations in the computation time, the speed can be improved further.

The remainder of this brief is outlined as follows. DPA attacks applied on the ECC device are introduced in Section II.

Manuscript received August 28, 2011; revised November 27, 2011; accepted February 25, 2012. Date of publication April 19, 2012; date of current version May 16, 2012. This work was supported by the National Science Council of Taiwan under Grant NSC99-2220-E-009-068. This paper was recommended by Associate Editor Z. Wang.

The authors are with the Department of Electronics Engineering and the Institute of Electronics, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: jenweilee@gmail.com; hcchang@si2lab.org; cylee@si2lab.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2012.2190857

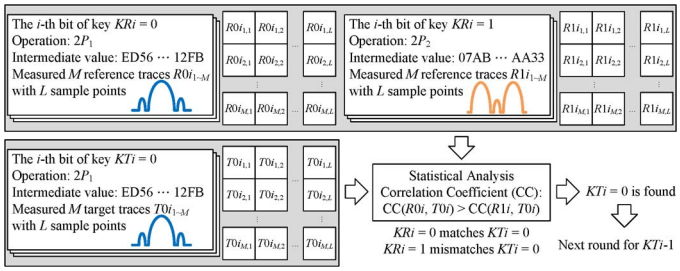


Fig. 1. DPA attacks on an ECC device operating in a specific domain.

The proposed countermeasure method and design architecture are given in Sections III and IV, respectively. Section V shows the FPGA power measurements and ASIC implementation results. Section VI concludes this work.

II. DPA Attacks on ECC Device

For the SPA resistance, the double-and-add-always approach given in Algorithm 1 is adopted to regularly perform the EC scalar multiplication (ECSM) $KP = P + \cdots + P$, where K is the m-bit private key and P is a point on ECs. However, the intermediate values of EC point doubling (ECPD) in Steps 3 and 4 depend on the zero and nonzero bits of the key value. Hence, with a chosen point P, the key value can be distinguished by matching the power trace segment of ECPD calculation.

Fig. 1 shows the scenario of DPA attacks. The power model can be characterized from two different key bit values by measuring the device sample before the statistical analysis, which computes the correlation between the measured target power traces and the power model. If the target key bit matches one of the chosen key bits, the correlation value will be larger than that of the others due to the same operation and processed data. Through this approach, the overall binary key can be extracted after $m - 1$ rounds in linear time.

Algorithm 1 Double-and-add-always ECSM

Input: K and P
Output: KP
1. Let $P_1 \leftarrow P$, $P_2 \leftarrow 2P$
2. **For** i from $m-2$ to 0 **do**
3. **If** $K_i = 1$ **then** $P_1 \leftarrow P_1 + P_2$, $P_2 \leftarrow 2P_2$
4. **else** $P_2 \leftarrow P_1 + P_2$, $P_1 \leftarrow 2P_1$
5. **Return** P_1

III. Proposed Algorithm Against DPA Attacks

The fundamental concept of DPA countermeasure is to break the dependence between intermediate values and power traces. For achieving the point calculation, the well-known Montgomery algorithm is usually adopted to perform the field arithmetic in a specific domain such that $A \equiv a \cdot 2^m (\mathrm{mod}\ p)$, where a is in the integer domain and 2^m is the Montgomery constant with m-bit field length. In this brief, we introduce an approach to resist the DPA attacks at modular algorithm by calculating the operands in a randomized Montgomery domain $A \equiv a \cdot 2^{\lambda} (\mathrm{mod}\ p)$, where the domain value λ equals the Hamming weight (HW) of an n-bit random value r. Note that n is the maximum field length, and the bit values of $\{r_{n-1}, r_{n-2}, \ldots, r_m\}$ are set to zero for preventing λ from exceeding m. Because the proposed method is to randomize intermediate values in basic modular operations, the double-and-add-always ECSM shown in Algorithm 1 against SPA attacks can be applied, and there is no need for external parameter such as point order $\#E$ which is not given in several protocols, including the Diffie–Hellman key exchange [1].

A. RMM

Algorithm 2 shows our proposed randomized Montgomery multiplication (RMM) which contains two operating steps in every iteration to change the intermediate domain value λ', and these steps are determined by the ith bit of random value r. If $r_i = 1$, the domain value of the output operand R decreases by one in Step 4; the domain value remains the same as $r_i = 0$ in Step 5. The functionality can be derived as follows:

1) For first iteration, the intermediate result of R is $(X_0 \cdot Y)(r_0 \cdot 2^{-1})(\mathrm{mod}\ p)$.
2) For the second iteration, R becomes $((X_0 \cdot Y)(r_0 \cdot 2^{-1})(\mathrm{mod}\ p) + X_1(2^{1-HW(r_0)} \cdot Y))(r_1 \cdot 2^{-1})(\mathrm{mod}\ p)$.
3) Until the mth iteration, the final result of R is $(\cdots(((X_0 \cdot Y)(r_0 \cdot 2^{-1})(\mathrm{mod}\ p) + X_1(2^{1-HW(r_0)} \cdot Y))(r_1 \cdot 2^{-1})(\mathrm{mod}\ p) + X_2(2^{2-HW(r_1,r_0)} \cdot Y))(r_2 \cdot 2^{-1})(\mathrm{mod}\ p) + \cdots + X_{m-1}(2^{m-1-HW(r_{m-2},\ldots,r_1,r_0)} \cdot Y))(r_{m-1} \cdot 2^{-1})(\mathrm{mod}\ p) = (X_0 \cdot Y \cdot 2^{-HW(r_{m-1},\ldots,r_0)})(\mathrm{mod}\ p) + (X_1 \cdot Y \cdot 2^{-HW(r_{m-1},\ldots,r_0)+1})(\mathrm{mod}\ p) + \cdots + (X_{m-1} \cdot Y \cdot 2^{-HW(r_{m-1},\ldots,r_0)+m-1})(\mathrm{mod}\ p) = X \cdot Y \cdot 2^{-HW(r_{m-1},\ldots,r_0)}(\mathrm{mod}\ p) = X \cdot Y \cdot 2^{-\lambda}(\mathrm{mod}\ p)$.

Hence, the RMM can be performed in m iterations, the same as those in conventional Montgomery multiplication [6].

Algorithm 2 Randomized Montgomery multiplication

Input: X, Y, p, and r
Output: $R = RMM(X, Y) \equiv X \cdot Y \cdot 2^{-\lambda} (\mathrm{mod}\ p)$
1. Let $V = X$, $R = 0$, $S = Y$
2. **For** i from 0 to $m-1$ **do**
3. $R = R + V_0 \cdot S (\mathrm{mod}\ p)$, $V = V/2$
4. **If** $r_i = 1$ **then** $R = R/2 (\mathrm{mod}\ p)$
5. **else** $S = 2S (\mathrm{mod}\ p)$
6. **Return** R

B. RMD

To achieve the division in Montgomery domain, Kaliski [14] first proposed an iterative algorithm which needs $2m$ iterations of successive reduction, m iterations for degree recovery (reduce intermediate domain value λ' to be m as $\lambda' > m$), and two additional Montgomery multiplications with a final modular reduction $p - R$. The algorithm presented in [14] is formulated from identical equations as follows:

$$\begin{cases} Y \cdot R \equiv -U \cdot 2^{\lambda'} & (\mathrm{mod}\ p) \\ Y \cdot S \equiv V \cdot 2^{\lambda'} & (\mathrm{mod}\ p). \end{cases}$$

Based on Kaliski's method, we derive a new randomized Montgomery division (RMD) which is described in Algorithm 3. To directly achieve the division operation without additional multiplication and final modular reduction, our method is to modify the initial values of (U, V, R, S) to be $(p, Y, 0, X)$ in Step 1 and the RS data path with modular subtraction in Steps 10, 11, 13, and 14. Then, the identities become

$$\begin{cases} X^{-1} \cdot Y \cdot R \equiv U \cdot 2^{\lambda'} & (\mathrm{mod}\ p) \\ X^{-1} \cdot Y \cdot S \equiv V \cdot 2^{\lambda'} & (\mathrm{mod}\ p). \end{cases}$$

Similar to Algorithm 2, the RS data path between the Montgomery domain and integer domain is determined by the ith bit value of r. The domain value of operands R and S increases by one as $r_i = 1$ and remains the same as $r_i = 0$.

For further reducing the degree recovery phase, the RS data path turns into dividing values by two in Steps 5, 8, 11, and 14 to keep the intermediate domain value in $\lambda = HW(r)$ as $i = m$. Thus, the identities in Algorithm 3 are given as follows:

$$\text{If } i < m, \text{then} \begin{cases} X^{-1} \cdot Y \cdot R \equiv U \cdot 2^{\lambda'} & (\mathrm{mod}\ p) \\ X^{-1} \cdot Y \cdot S \equiv V \cdot 2^{\lambda'} & (\mathrm{mod}\ p) \end{cases}$$

$$\text{else} \begin{cases} X^{-1} \cdot Y \cdot R \equiv U \cdot 2^{\lambda'} & (\mathrm{mod}\ p) \\ X^{-1} \cdot Y \cdot S \equiv V \cdot 2^{\lambda'} & (\mathrm{mod}\ p). \end{cases}$$

Before the last iteration, both U and V are one because the initial values of U and V are relatively prime. Then, after finishing the iterative operations in Step 2, the values of (U, V, R, S) become $(1, 0, X \cdot Y^{-1} \cdot 2^{\lambda} (\mathrm{mod}\ p), 0)$. As a result, the proposed randomized division algorithm requires only $2m$ iterations of successive reduction.

Algorithm 3 Randomized Montgomery division

Input: X, Y, p, and r
Output: $R = RMD(X, Y) \equiv X \cdot Y^{-1} \cdot 2^{\lambda} (\mathrm{mod}\ p)$
1. Let $U = p$, $V = Y$, $R = 0$, $S = X$
2. **While** $(V > 0)$ **do**
3. **If** U is even **then** $U = U/2$
4. **If** $r_i = 1$ **then** $S = 2S (\mathrm{mod}\ p)$
5. **else** $R = R/2 (\mathrm{mod}\ p)$
6. **else if** V is even **then** $V = V/2$
7. **If** $r_i = 1$ **then** $R = 2R (\mathrm{mod}\ p)$
8. **else** $S = S/2 (\mathrm{mod}\ p)$
9. **else if** $U > V$ **then** $U = (U - V)/2$
10. **If** $r_i = 1$ **then** $R = R - S (\mathrm{mod}\ p)$, $S = 2S (\mathrm{mod}\ p)$
11. **else** $R = (R - S)/2 (\mathrm{mod}\ p)$
12. **else** $V = (V - U)/2$
13. **If** $r_i = 1$ **then** $S = S - R (\mathrm{mod}\ p)$, $R = 2R (\mathrm{mod}\ p)$
14. **else** $S = (S - R)/2 (\mathrm{mod}\ p)$
15. **If** $i < m$ **then** $i = i + 1$
16. **Return** R

Table I shows the expected operation time and the comparison with related works on modifying the Montgomery division algorithm. With randomization capability, Algorithm 3 will also benefit the hardware design owing to the low latency.

TABLE I
ANALYSIS OF VARIOUS DIVISION ALGORITHMS

	Algorithm 3	TCAS-I'06 [3]	ESSCIRC'10 [9]
Iteration Time	$2m$	$2m$	$3m$
Multiplication	0	$2 \sim 3$	0
Domain	Random 2^{λ}, $0 \le \lambda \le m$	Fixed 2^m	Fixed 2^m

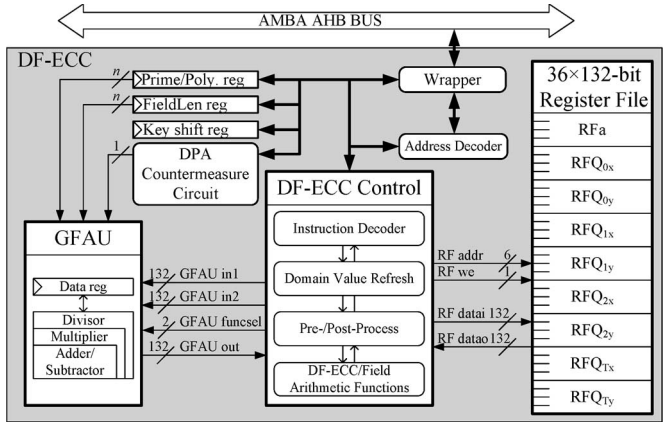


Fig. 2. Overall diagram for the DF-ECC processor.

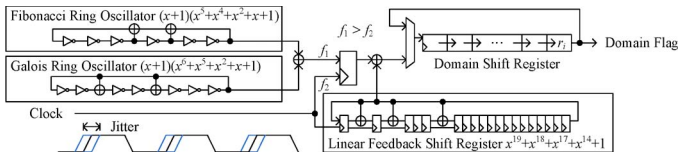


Fig. 3. Domain flag is to randomly assign operating domain for GFAU.

IV. DPA-RESISTANT DF-ECC PROCESSOR

Fig. 2 shows the block diagram of the proposed dual-field ECC (DF-ECC) processor with a standard advanced microcontroller bus architecture advanced high performance bus interface. For the DPA resistance, all field operations over $GF(p)$ and $GF(2^m)$ are performed by the *Galois field arithmetic unit* (GFAU) in a randomized Montgomery domain. The operating domain is determined by the value in domain shift register, which is sourced from a 1-bit random number generator (RNG) and refreshed before the next ECSM calculation. For flexibility, we use an all-digital RNG utilizing the cycle-to-cycle time jitter in free-running oscillators with a synchronous feedback postprocessor [15]. The overall architecture of DPA countermeasure circuit is shown in Fig. 3. To efficiently store the 521-bit operands including EC coefficient and points, a block memory of register file is exploited. Moreover, in order to real-time perform the ECC schemes such as signature and key exchange, the instruction decoder and pre-/postprocess of domain conversion are combined in our DF-ECC processor.

Fig. 4 shows the detailed GFAU architecture. As the iterative operations in Algorithm 3 are performed in one cycle, the critical path is to calculate the results of R or S consisting of the UV comparison with modular operations. The time-critical comparison operation $U > V$ achieved by a subtraction is nearly equal to an addition delay. Since the results of R and S are irrelevant to the results of operands U or V, a fully pipelined stage can be inserted between the UV and RS data paths to moderate the critical path, where the critical path is the path

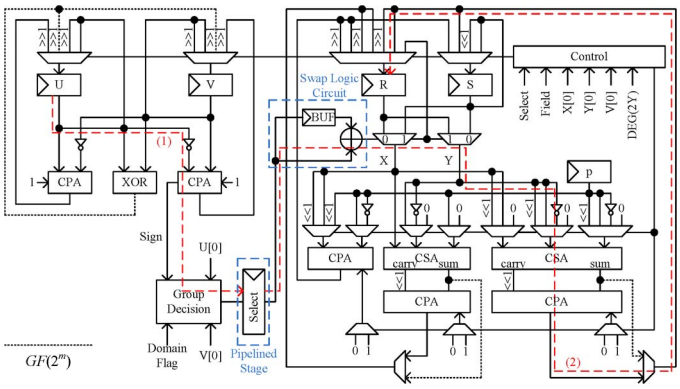


Fig. 4. All dual-field randomized Montgomery operations are integrated into the fully pipelined GFAU with hardware sharing.

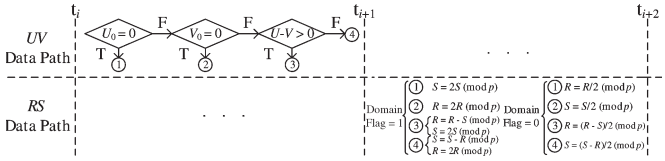


Fig. 5. Fully pipelined scheme for the RMD.

$(1)+(2)/(2)$ in Fig. 4 before/after pipelining. As the UV data path is determined, then the next cycle is to set the values of the operands R and S and simultaneously determine the next case until $V=0$. Although one additional cycle is needed after pipelining, this is negligible as the operation takes hundreds or thousands of cycles. The timing flow of pipelined scheme is shown in Fig. 5. Moreover, to reduce the hardware cost, symmetric modular operations such as $R=(R-S)/2(\mathrm{mod}\ p)$ and $S=(S-R)/2(\mathrm{mod}\ p)$ in Algorithm 3 can be executed by the same computational unit with a swap logic circuit, which is to switch the input operands of RS data path. In Algorithm 3, the RS data path can be classified into two groups: The first group includes Steps 4 and 5 and Steps 10 and 11, and the second one consists of Steps 7 and 8 and Steps 13 and 14. The data flows of R and S are switched as the processing group is different from the group in the previous cycle. Furthermore, since the point calculation is a serial field operation, both of the temporary registers and modular operations can be shared for the operands V, S, and R in Algorithm 2 and Algorithm 3. The modular operations including addition/subtraction and shifting in the iterative loop can be effectively performed by exploiting the carry-save adder (CSA) with a carry-propagation adder. Another benefit is that it can achieve the additive operation for $GF(2^m)$ by circuit integration because the sum of CSAs equals two bitwise XOR operators.

Since the primary inputs of EC coefficient and points are in the integer domain, the domain conversion can be performed by the proposed RMD such that $RMD(a,1)=a2^{\lambda}(\mathrm{mod}\ p)$. On the other hand, to return the point coordinates in the integer domain, the RMM can be exploited as $RMM(a2^{\lambda},1)=a(\mathrm{mod}\ p)$. For calculating one ECSM in affine coordinates, the overhead of domain conversion is three RMD and two RMM operations; both of them can be performed by GFAU to avoid any precomputation from the host system.

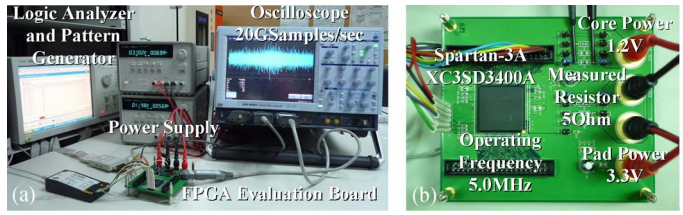


Fig. 6. (a) Environment of power measurement. (b) Current running through the ECC processor recorded by measuring the voltage drop via a resistor in series with the board power pin and FPGA power pin.

TABLE II
FPGA IMPLEMENTATION RESULTS

Design	Area (Slices)	$f_{\max}$ (MHz)	ECSM Method	Algorithm for Field Arithmetic
I	12,112 (51%)	23.0	Double-and-add-always	Montgomery
II	12,757 (53%)	23.0	Double-and-add-always	Randomized Montgomery Operations

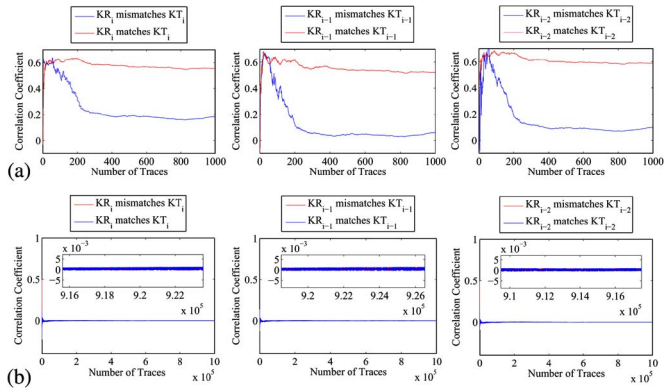
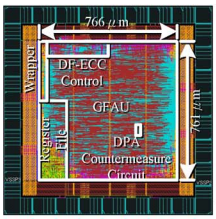


Fig. 7. Correlation coefficients of various target and reference traces obtained from (a) unprotected Design I and (b) protected Design II with enlarged view.

V. FPGA VERIFICATION AND IMPLEMENTATION RESULTS

Based on our proposed architecture, two different 256-bit DF-ECC processors are designed on an FPGA verification platform to evaluate the DPA resistance. The verification environment is shown in Fig. 6, and the performance results are given in Table II.

Fig. 7(a) shows the DPA attacks on the unprotected Design I using the conventional Montgomery algorithm to reveal the first three bits of key value "101." From 10^3 measurements, the correlation coefficients of correct hypothesis $(CC(RH_i,TH_i))$ converge to about 0.6, while those of incorrect hypothesis $(CC(R\bar{H}_i,TH_i))$ converge to values below 0.3, where H is the hypothesis of binary key value. Hence, the key value can be distinguished from a difference of at least 0.3 in correlation coefficients. In contrast, after collecting 10^6 power traces from the protected Design II, which uses randomized Montgomery operations given in Algorithm 2 and Algorithm 3, the correlation coefficients of correct and incorrect hypotheses shown in Fig. 7(b) are close to zero and cannot be scattered. This means that the power model is uncorrelated with the target power traces, and there is no biased information of the key value from the differences in correlation coefficients. Consequently, the statistical analysis of power measurements



	Ours †						ESSCIRC'10 [9] ‡						Ours †		TCAS-II'09 [5] ‡		MWSCAS'09 [6] †	Ours †	TCAS-I'06 [3] †
Technology	90-nm						90-nm						90-nm		0.13-μm		0.18-μm	Virtex 2 Pro	Virtex 2 Pro
Core Area (mm²)	0.58						0.55						0.21		1.44		2.10	2,662 CLB slices	15,755 CLB slices
KGates	168						170						61.3		169		69		
Operating Field	$GF(p)$			$GF(2^m)$			$GF(p)$			$GF(2^m)$			$GF(p)$	$GF(2^m)$	$GF(p)$	$GF(2^m)$	$GF(2^m)$	$GF(p)$	$GF(p)$
	160	256	521	163	283	409	160	256	521	163	283	409	160	163	160	160	163	256	256
f_{max} (MHz)	256	256	250	270	263	263	154	147	132	188	182	166	277	277	121	146	181	43.5	39.5
Op. Time (ms)	0.74	1.89	8.08	0.63	1.95	4.65	1.62	4.4	19.2	1.15	3.33	8.2	0.71	0.61	0.61	0.37	1.89	11.1	3.86
AT product	1	1	1	1	1	1	2.08	2.21	2.25	1.73	1.62	1.67	1	1	1.95 #	1.38 #	3.87 #	1	2.15
Energy (mJ/Op.)	0.02	0.08	0.45	0.02	0.08	0.24	0.11	0.30	1.12	0.08	0.23	0.48	0.01	0.01	0.04	0.03	0.26	-	-
ECSM method	Double-and-add-always						Double-and-add-always						Double-and-add-always		Double-and-add/sub		Double-and-add	Double-and-add-always	Double-and-add
Resistance	SPA and DPA attacks						SPA and DPA attacks						SPA and DPA attacks		-		-	SPA and DPA attacks	-

† Post-layout simulation. ‡ Measurement. # Technology scaled area-time (AT) product = (core area × λ^2) × (op. time × λ), where λ = 90-nm/technology.

Fig. 8. ASIC layout view and implementation results compared with related works.

TABLE III
OVERHEAD FOR DPA RESISTANCE

	Ours	[9]	[11]	[12]	[13]
Design	521 ECC	521 ECC	128 AES	128 AES	S-box
Area	4.3%	10%	210%	7.2%	19%
Op. Time	0	14.0%[I]	288%	100%	0
Energy	5.2%	20.8%[II]	270%	33%	120%*

$$\text{Overhead} = \frac{\text{Result differences between protected and unprotected circuit}}{\text{Results of unprotected circuit}} \times 100\%.$$

I Estimated by cycle count × clock period.

II Estimated by operation time × average power.

* Two ring oscillators and one S-box consume 180μW and 150μW, respectively.

shows that the proposed countermeasure enhances the security against DPA attacks.

Our proposed 521-bit DF-ECC processor was implemented by UMC 90-nm CMOS technology. Moreover, to compare with related works, one 163-bit version of DF-ECC processor and one 256-bit design over $GF(p)$ were implemented to ASIC and FPGA, respectively. The layout and implementation results with comparisons are given in Fig. 8. In terms of area–time product, our DF-ECC processor outperforms other approaches. By reducing the division iteration time and randomizing the intermediate values in field arithmetic without increasing the key size, our work is at least 40% faster than the previous 521-bit design [9] with comparable hardware complexity. Compared with a four-multiplier-based ECC processor without power-analysis protection [5], our highly integrated GFAU architecture achieves competitive speed with 60% less gate counts. In [6], an unprotected design based on $GF(2^{163})$ and a fixed polynomial is optimized for hardware speed. We design the ECC processor with dual-field support and apply the pipelining approach. The throughput achieved is two times higher than that reported in [6].

For the DPA resistance, our approach is to mask the processed data uncorrelated with power traces without changing the logic family and without dominating the power consumption of key-dependent operations. From the comparison given in Table III, our proposed countermeasure is superior to others not only in hardware cost but also in energy dissipation.

VI. CONCLUSION

In this brief, we have introduced a new randomized Montgomery algorithm which is suitable for ECC hardware implementation against DPA attacks. Without modifying the logic circuit, the relationship between target power traces and power model can be broken by performing field arithmetic in an unpredictable domain. A free precomputation scheme has been proposed also to immediately carry out the domain conversion for supporting real-time processing.

The proposed DPA countermeasure approach has been analyzed on an FPGA platform. Attacks on the unprotected designs reveal the private key within 1000 power traces, while the key value of the protected core cannot be extracted after one million power traces. Circuit overhead for randomly determining the operating domain can be integrated into the system without speed degradation. By using a UMC 90-nm technology, our protected 521-bit DF-ECC processor, with 4.3% area and 5.2% average power overhead, can perform one $GF(p_{521})$ ECSM in 8.08 ms and one $GF(2^{409})$ ECSM in 4.65 ms, respectively. We believe that both high performance and efficient DPA countermeasure are achieved in our proposed DF-ECC processor.

REFERENCES

[1] *Standard Specifications or Public-Key Cryptography*, IEEE Std. 1363, Jan. 2000.
[2] *Digital Signature Standard*, FIPS PUB 186-3 Std., Jun. 2009.
[3] C. J. McIvor, M. McLoone, and J. V. McCanny, "Hardware elliptic curve cryptographic processor over $GF(p)$," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 9, pp. 1946–1957, Sep. 2006.
[4] K. Sakiyama, L. Batina, B. Preneel, and I. Verbauwhede, "Multicore curve-based cryptoprocessor with reconfigurable modular arithmetic logic units over $GF(2^n)$," *IEEE Trans. Comput.*, vol. 56, no. 9, pp. 1269–1282, Sep. 2007.
[5] J.-Y. Lai and C.-T. Huang, "A highly efficient cipher processor for dual-field elliptic curve cryptography," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 5, pp. 394–398, May 2009.
[6] J.-H. Hong and W.-C. Wu, "The design of high performance elliptic curve cryptographic," in *Proc. IEEE Int. MWSCAS*, Aug. 2009, pp. 527–530.
[7] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Int. Cryptol. Conf. Adv. Cryptol.*, 1999, pp. 388–397.
[8] P. Longa and A. Miri, "Fast and flexible elliptic curve point arithmetic over prime fields," *IEEE Trans. Comput.*, vol. 57, no. 3, pp. 289–302, Mar. 2008.
[9] J.-W. Lee, Y.-L. Chen, C.-Y. Tseng, H.-C. Chang, and C.-Y. Lee, "A 521-bit dual-field elliptic curve cryptographic processor with power analysis resistance," in *Proc. ESSCIRC*, Sep. 2010, pp. 206–209.
[10] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. CHES*, 2004, vol. 3156, pp. 135–153.
[11] D. Hwang, K. Tiri, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "AES-based security coprocessor IC in 0.18-μm CMOS with resistance to differential power analysis side-channel attacks," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 781–792, Apr. 2006.
[12] C. Tokunaga and D. Blaauw, "Securing encryption systems with a switched capacitor current equalizer," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 23–31, Jan. 2010.
[13] P.-C. Liu, H.-C. Chang, and C.-Y. Lee, "A low overhead DPA countermeasure circuit based on ring oscillators," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 7, pp. 546–550, Jul. 2010.
[14] B. S. Kaliski, "The Montgomery inverse and its applications," *IEEE Trans. Comput.*, vol. 44, no. 8, pp. 1064–1065, Aug. 1995.
[15] J. Golic, "New methods for digital generation and postprocessing of random data," *IEEE Trans. Comput.*, vol. 55, no. 10, pp. 1217–1229, Oct. 2006.