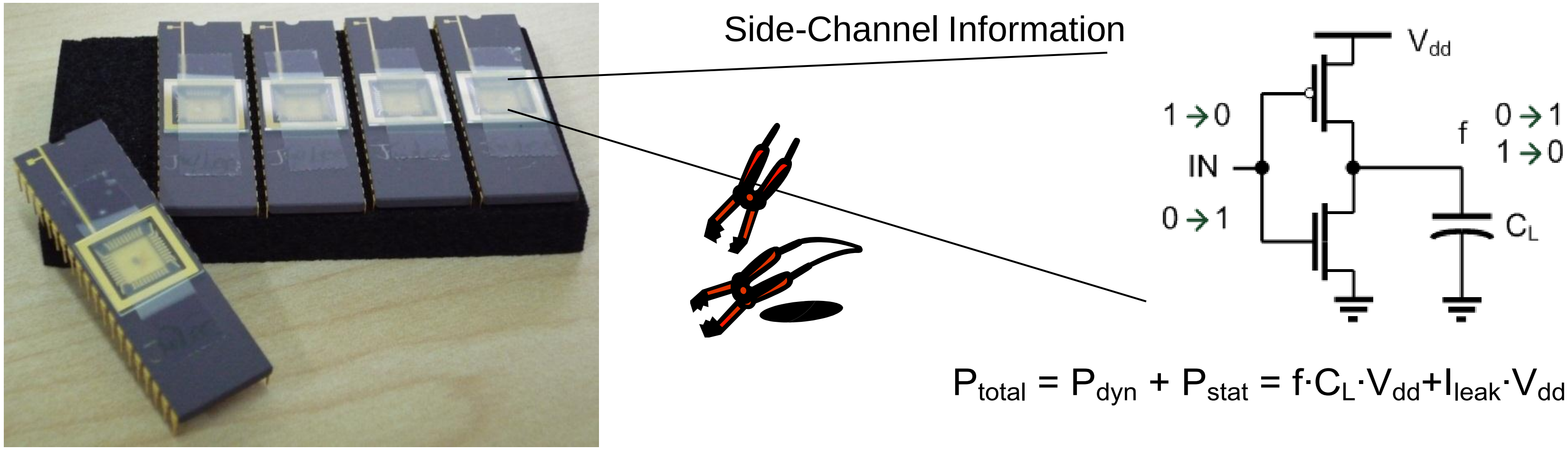
An Efficient Countermeasure against Correlation Power-Analysis Attacks with Randomized Montgomery Operations for DF-ECC Processor

Jen-Wei Lee, Szu-Chi Chung, Hsie-Chia Chang, and Chen-Yi Lee

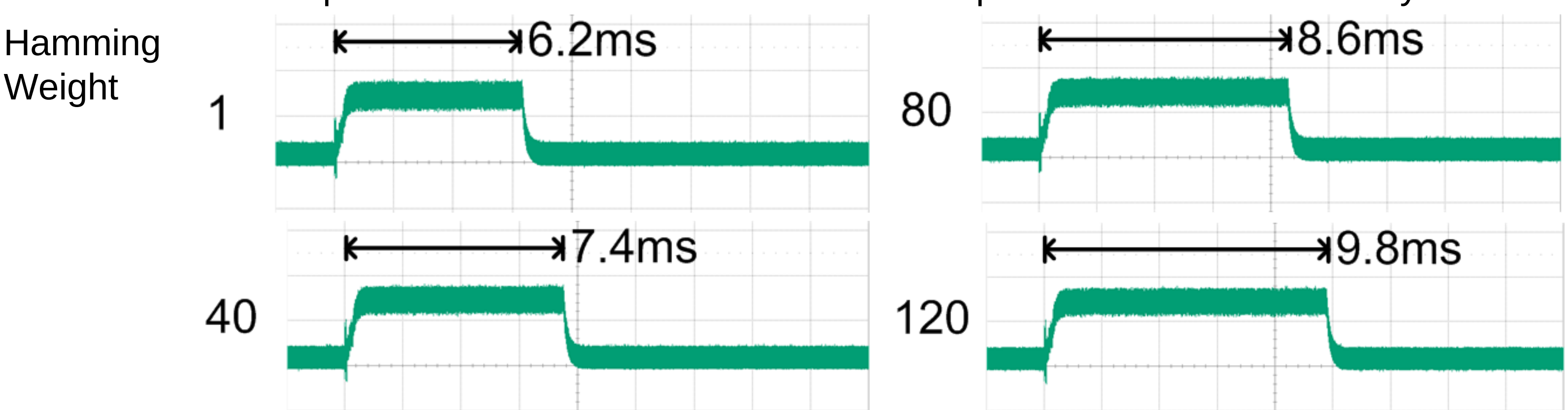
Department of Electronics Engineering and Institute of Electronics,
National Chiao-Tung University (NCTU), Hsinchu, Taiwan
Email: jenweilee@gmail.com

Power-Analysis Attacks



$$P_{total} = P_{dyn} + P_{stat} = f \cdot C_L \cdot V_{dd} + I_{leak} \cdot V_{dd}$$

Example of Power Traces for 160-bit ECC Chip with Different Private Key Values



Execution time depends on key value by direct implementation

→ secrete information leakage through simple power-analysis (SPA) attack

Power-Analysis Attacks

- **SPA attack can be counteracted by unified operations**
- **Correlation power-analysis (CPA) attack**
  - utilize statistical analysis to disclose private information of cryptographic devices
  - work on EC integrated encryption, single pass EC Diffie-Hellman or Menezes-Qu-Vanstone key agreement
- **CPA attack on SPA-resistant ECC device**
  - key-dependent EC scalar multiplication (ECSM)

Algorithm: Montgomery Ladder
Input: an integer K and a point P
Output: KP
1. $P_1 \leftarrow P, P_2 \leftarrow 2P$;
2. For i from $m-2$ down to 0 do
   If $K_i = 1$ then
      $P_1 \leftarrow ECPA(P_1, P_2), P_2 \leftarrow ECPD(P_2)$
   else
      $P_2 \leftarrow ECPA(P_2, P_1), P_1 \leftarrow ECPD(P_1)$
   End
3. Return P_1

$P_1 = P$, $P_2 = 2P$
- $K_{m-2}=1$: $P_1 = 3P$, $P_2 = 4P$
  - $K_{m-3}=1$: $P_1 = 7P$, $P_2 = 8P$ …
  - $K_{m-3}=0$: $P_1 = 6P$, $P_2 = 7P$ …
- $K_{m-2}=0$: $P_1 = 2P$, $P_2 = 3P$
  - $K_{m-3}=1$: $P_1 = 5P$, $P_2 = 6P$ …
  - $K_{m-3}=0$: $P_1 = 4P$, $P_2 = 5P$ …

Time complexity is O(2m)

Previous Works

- **Circuit level**
  - wave dynamic differential logic [HWANG’06]
  - random switching logic [SAEKI’09]
- **Register addressing**
  - random register renaming [ITOH’03]
- **Algorithm level**
  - randomized EC point [CORON’99]
  - randomized scalar key [CORON’99]
  - randomized projective coordinates [CORON’99]
  - elliptic curve isomorphisms over $GF(p)$ [JOYE’01]
- **Software implementation**
  - random delay generation [CORON’09]

Motivation

- **Provide a solution that is suitable and efficient for ECC hardware implementation**
  - support dual-field operations for high security level
    - dual-field ECC (DF-ECC) function is approved in IEEE P1363
  - compatible to current public-key cryptography
    - use initial EC parameters
  - hardware speed
    - field inversion/division and multiplication dominate execution time
  - hardware complexity
    - arithmetic unit integration

Our Solution

- **Mask intermediate values by computing field arithmetic in a randomized domain**
  - Montgomery domain
    - $A \equiv a \cdot 2^m (mod\ p)$, a is in integer domain and m is field length
  - random domain (or random field automorphism)
    - $A \equiv a \cdot 2^\lambda (mod\ p)$, domain value λ equals to hamming weight of an m-bit non-zero random value α

Table 1. Operations in Randomized Domain

Operation	Arithmetic
randomized Montgomery multiplication (RMM)	$RMM(X,Y) \equiv x \cdot y \cdot 2^\lambda (mod\ p)$
randomized Montgomery division (RMD)	$RMD(X,Y) \equiv x \cdot y^{-1} \cdot 2^\lambda (mod\ p)$
randomized addition (RA)	$RA(X,Y) \equiv (x+y) \cdot 2^\lambda (mod\ p)$
randomized subtraction (RS)	$RS(X,Y) \equiv (x-y) \cdot 2^\lambda (mod\ p)$

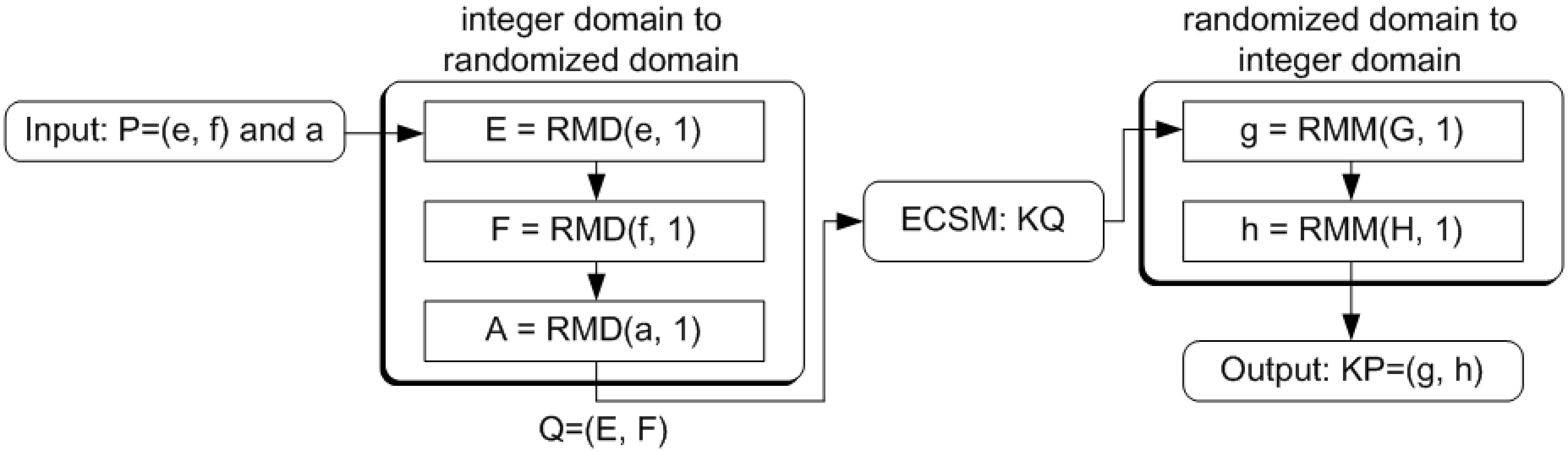
Our Solution

- **Random field automorphism for ECSM calculation**
  - field automorphic function φ

$$\varphi: P = (e, f) \rightarrow Q = (E, F)$$

    - $e, f, E \equiv e \cdot 2^{\lambda} (mod\ p), F \equiv f \cdot 2^{\lambda} (mod\ p)$
    - $e \neq E, f \neq F$ i.i.f. $2^{\lambda} \neq 1 (mod\ p)$ with $0 < \lambda \leq m$
  - inverse field automorphic function φ^{-1}

$$\varphi^{-1}: \mathrm{KQ} = (\mathrm{G}, \mathrm{H}) \rightarrow KP = (g, h)$$



Proposed Randomized Montgomery Algorithm

- **Radix-2 RMM**
  - if $\alpha_i = 1$
    - decrease domain value by 1 in step 4
      - $R = R/2$
  - if $\alpha_i = 0$
    - remain domain value in step 5
      - $R = R$
  - after m iterations
    - domain value is $-\lambda$

Algorithm 2 Radix-2 randomized Montgomery multiplication

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

Proposed Randomized Montgomery Algorithm

➢ Radix-2 RMD

- if $\alpha_i = 1$
  - increase domain value by 1 in steps 4, 7, 10, 13
    - $U = U/2$
    - $R = 2R$
- if $\alpha_i = 0$
  - remain domain value in steps 5, 8, 11, 14
- after m iterations
  - domain value is λ

Algorithm 4 Radix-2 randomized Montgomery division

Input: X, Y, p, and α

Output: $R = \text{RMD}(X, Y)$

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** $\alpha_i = 1$ **then** $S \equiv 2S \pmod p$
5.   **else** $R \equiv R/2 \pmod p$
6.   **else if** V is even **then** $V = V/2$
7.   **If** $\alpha_i = 1$ **then** $R \equiv 2R \pmod p$
8.   **else** $S \equiv S/2 \pmod p$
9.   **else if** $U > V$ **then** $U = (U - V)/2$
10.   **If** $\alpha_i = 1$ **then** $R \equiv R - S \pmod p$, $S \equiv 2S \pmod p$
11.   **else** $R \equiv (R - S)/2 \pmod p$
12.   **else** $V = (V - U)/2$
13.   **If** $\alpha_i = 1$ **then** $S \equiv S - R \pmod p$, $R \equiv 2R \pmod p$
14.   **else** $S \equiv (S - R)/2 \pmod p$
15.   **If** $i < m$ **then** $i = i + 1$
16. **Return** R

Extend Radix-2 to Radix-4 Approach

- **Based on extended Euclidean algorithm**

$$X^{-1}\cdot Y\cdot R\equiv U\cdot 2^{i}(\text{mod } p)$$
$$X^{-1}\cdot Y\cdot S\equiv V\cdot 2^{i}(\text{mod } p)$$

initial values: $(U,V,R,S)\Rightarrow(p,Y,0,X)$

final iteration: $(U,V,R,S)\Rightarrow(1,0,XY^{-1}2^{m}(\text{mod } p),0)$

1. U or V (mod 4) = 0
2. U (mod 4) = V (mod 4)
3. U/V (mod 4) is even and V/U (mod 4) is odd
4. U and V (mod 4) is odd

c	d	Properties
0	0, 1, 2, or 3	$gcd(U,V)=gcd(\frac{U}{4},V)$
1, 2, or 3	0	$gcd(U,V)=gcd(U,\frac{V}{4})$
$c=d$		$gcd(U,V)=gcd(\frac{U-V}{4},V)=gcd(U,\frac{V-U}{4})$
2	1 or 3	$gcd(U,V)=gcd(\frac{\frac{U}{2}-V}{2},V)=gcd(\frac{U}{2},\frac{V-\frac{U}{2}}{2})$
1 or 3	2	$gcd(U,V)=gcd(\frac{U-\frac{V}{2}}{2},\frac{V}{2})=gcd(U,\frac{\frac{V}{2}-U}{2})$
other		$gcd(U,V)=gcd(\frac{U-V}{2},V)=gcd(U,\frac{V-U}{2})$

$c = U$ (mod 4), $d = V$ (mod 4)

Extend Radix-2 to Radix-4 Approach

- **Modify iterative calculation in radix-4 RMM/RMD to ensure domain value decreases/increases by 2 to 0**
  - if two-bit random value is (11)
    - decrease/increase domain value by 2
  - if two-bit random value is (10) or (01)
    - decrease/increase domain value by 1
  - if two-bit random value is (00)
    - remain domain value

Proposed Randomized Montgomery Algorithm

➢ Radix-4 RMM

- if $(\alpha_{2i+1}, \alpha_{2i}) = (11)$
  - decrease domain value by 2 in step 5
    - $R = R/4$
- if $(\alpha_{2i+1}, \alpha_{2i}) = (10)$ or (01)
  - decrease domain value by 1 in step 6
    - $R = R/2$
- if $(\alpha_{2i+1}, \alpha_{2i}) = (00)$
  - remain domain value in step 7
    - $R = R$
- after $m/2$ iterations
  - Domain value is $-\lambda$

Algorithm 3. Radix-4 randomized Montgomery multiplication

Input: X, Y, p, and α

Output: $R = \mathrm{RMM}(X, Y)$

1. Let $V = X$, $R = 0$, $S = Y$
2. **For** i from 0 to $\lceil \frac{m}{2} \rceil - 1$ **do**
3. **If** $m \pmod 2 \equiv 1$ and $i = \lceil \frac{m}{2} \rceil - 1$ **then**
   $R \equiv R + V_0 \cdot S \pmod p$, $V = \frac{V}{2}$
4. **else**
   $R \equiv R + V_0 \cdot S + V_1 \cdot 2S \pmod p$, $V = \frac{V}{4}$
5. **If** $(\alpha_{2i+1}, \alpha_{2i}) = (1, 1)$ **then**
   $R \equiv \frac{R}{4} \pmod p$
6. **else if** $(\alpha_{2i+1}, \alpha_{2i}) = (1, 0)$ or $(0, 1)$ **then**
   $R \equiv \frac{R}{2} \pmod p$, $S \equiv 2S \pmod p$
7. **else**
   $S \equiv 4S \pmod p$
8. **Return** R

Proposed Randomized Montgomery Algorithm

➢ Radix-4 RMD

- if $(\alpha_{i+1}, \alpha_i) = (11)$
  - increase domain value by 2 in step 24
    - $R = 4R$
- if $(\alpha_{i+1}, \alpha_i) = (10)$ or (01)
  - increase domain value by 1 in step 25
    - $R = 4R/2$
- if $(\alpha_{i+1}, \alpha_i) = (00)$
  - remain domain value in step 26
    - $R = 4R/4$
- after $m/2$ iterations
  - domain value is λ

fixed

randomized

Algorithm 5. Radix-4 randomized Montgomery division

Input: X, Y, p, and α

Output: $R = \mathrm{RMD}(X, Y)$

1. Let $U = p, V = Y, R = 0, S = X, i = 0$
2. **While** $(V > 0)$ **do**
3. $\quad c \equiv U \pmod 4, d \equiv V \pmod 4, t = 2$
4. $\quad$ **If** $i = m - 1$ **then**
   $\quad\quad R \equiv 2R \pmod p, S \equiv 2S \pmod p, t = 1$
5. $\quad$ **else if** $c = 0$ **then** $U = \frac{U}{4}, S \equiv 4S \pmod p$
6. $\quad$ **else if** $d = 0$ **then** $V = \frac{V}{4}, R \equiv 4R \pmod p$
7. $\quad$ **else if** $c = d$ **then**
8. $\quad\quad$ **If** $U > V$ **then** $U = \frac{U-V}{4}$,
   $\quad\quad\quad R \equiv R - S \pmod p, S \equiv 4S \pmod p$
9. $\quad\quad$ **else** $V = \frac{V-U}{4}$,
   $\quad\quad\quad S \equiv S - R \pmod p, R \equiv 4R \pmod p$
10. $\quad$ **else if** $c = 2$ **then**
11. $\quad\quad$ **If** $\frac{U}{2} > V$ **then** $U = \frac{\frac{U}{2} - V}{2}$,
    $\quad\quad\quad R \equiv R - 2S \pmod p, S \equiv 4S \pmod p$
12. $\quad\quad$ **else** $V = \frac{V - \frac{U}{2}}{2}, U = \frac{U}{2}$,
    $\quad\quad\quad S \equiv 2S - R \pmod p, R \equiv 2R \pmod p$
13. $\quad$ **else if** $d = 2$ **then**
14. $\quad\quad$ **If** $U > \frac{V}{2}$ **then** $U = \frac{U - \frac{V}{2}}{2}, V = \frac{V}{2}$,
    $\quad\quad\quad R \equiv 2R - S \pmod p, S \equiv 2S \pmod p$
15. $\quad\quad$ **else** $V = \frac{\frac{V}{2} - U}{2}$,
    $\quad\quad\quad S \equiv S - 2R \pmod p, R \equiv 4R \pmod p$
16. $\quad$ **else**
17. $\quad\quad$ **If** $U > V$ **then** $U = \frac{U-V}{2}$,
    $\quad\quad\quad R \equiv R - S \pmod p, S \equiv 2S \pmod p, t = 1$
18. $\quad\quad$ **else** $V = \frac{V-U}{2}$,
    $\quad\quad\quad S \equiv S - R \pmod p, R \equiv 2R \pmod p, t = 1$
19. $\quad$ **If** $i < m$ **then**
20. $\quad\quad$ **If** $i = m - 1$ or $t = 1$ **then**
21. $\quad\quad\quad$ **If** $\alpha_i = 1$ **then** $R \equiv R \pmod p, S \equiv S \pmod p$
22. $\quad\quad\quad$ **else** $R \equiv \frac{R}{2} \pmod p, S \equiv \frac{S}{2} \pmod p$
23. $\quad\quad$ **else**
24. $\quad\quad\quad$ **If** $(\alpha_{i+1}, \alpha_i) = (1, 1)$ **then**
    $\quad\quad\quad\quad R \equiv R \pmod p, S \equiv S \pmod p$
25. $\quad\quad\quad$ **else if** $(\alpha_{i+1}, \alpha_i) = (1, 0)$ or $(0, 1)$ **then**
    $\quad\quad\quad\quad R \equiv \frac{R}{2} \pmod p, S \equiv \frac{S}{2} \pmod p$
26. $\quad\quad\quad$ **else**
    $\quad\quad\quad\quad R \equiv \frac{R}{4} \pmod p, S \equiv \frac{S}{4} \pmod p$
27. $\quad\quad i = i + t$
28. $\quad$ **else** $R \equiv \frac{R}{2^t} \pmod p, S \equiv \frac{S}{2^t} \pmod p$
29. **Return** R

Hardware Architecture of DF-ECC Processor

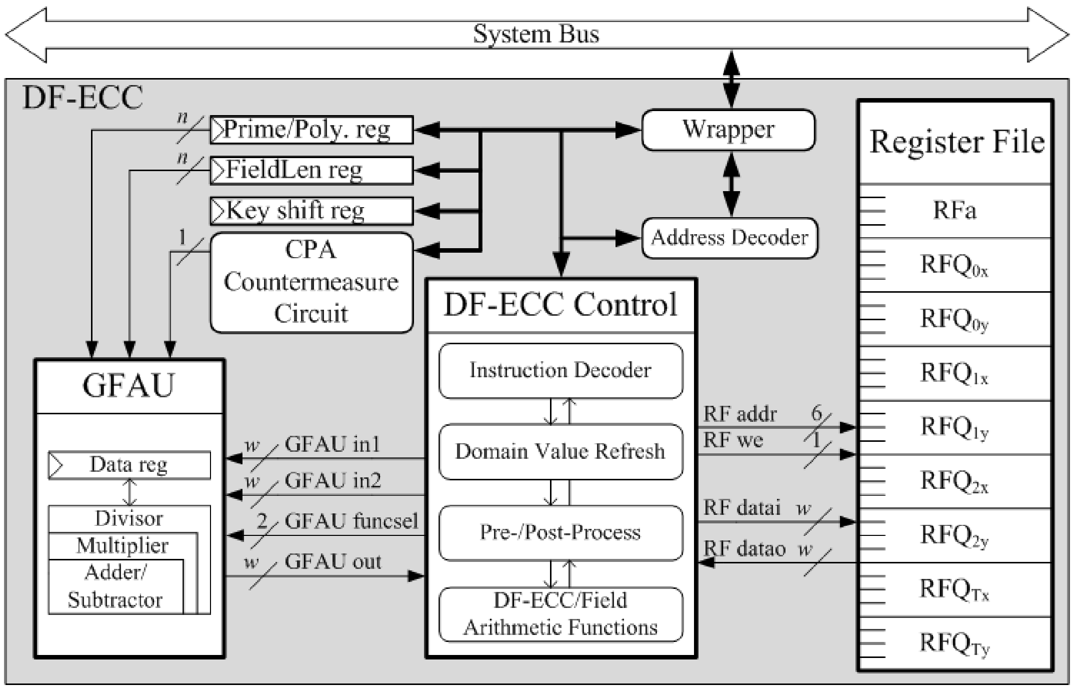


Fig. 2. Overall diagram for the DF-ECC processor.

DF-ECC processor
1. Galois field arithmetic unit (GFAU)
2. instant domain conversion (RMD(a,1) = A, RMM(A, 1) = a)
3. CPA countermeasure circuit

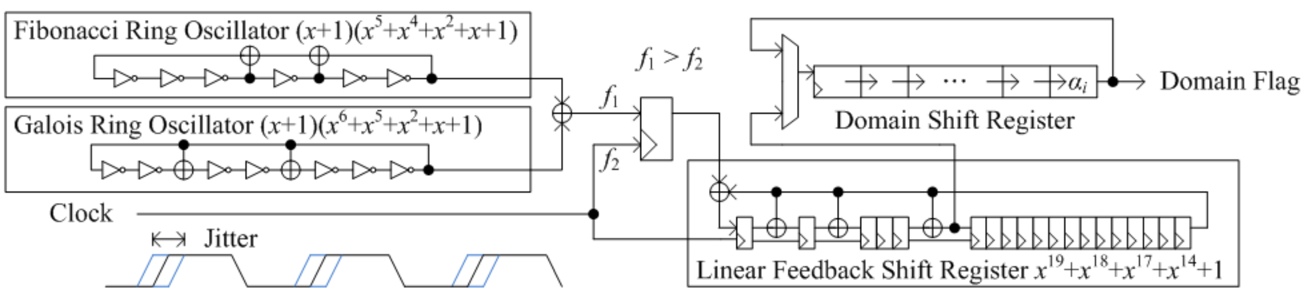
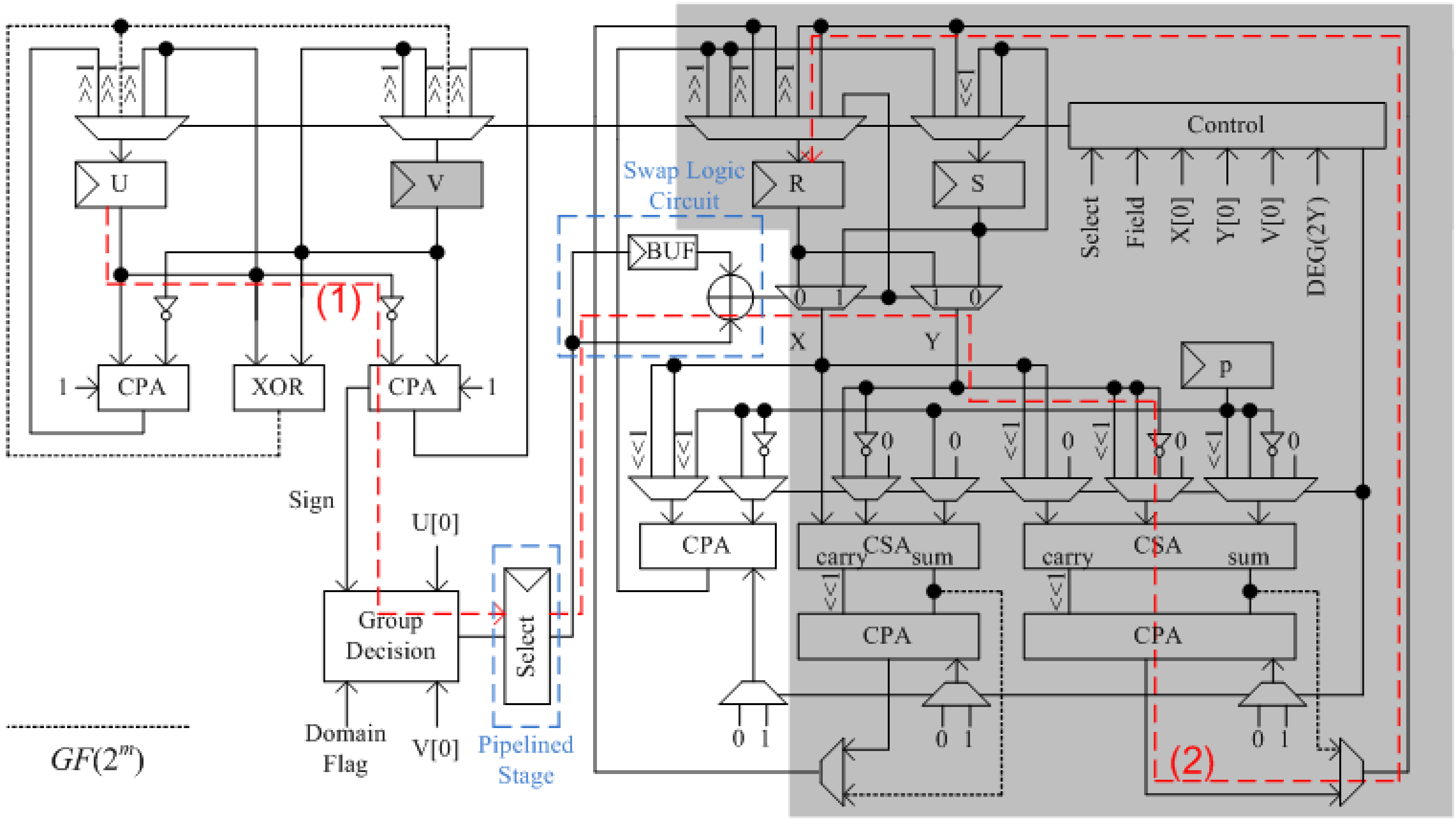


Fig. 3. The domain flag is to randomly assign operating domain for GFAU.

Ring-oscillator based RNG
1. portable applications
2. resolve reset problem

Hardware Architecture of DF-ECC Processor

- **Radix-2 GFAU**



1. fully-pipelining to remove path (1)
2. multiplier is shared in gray color

Verification and Measurement

➢ FPGA device

Fig. 7. (a) Environment of power measurement. (b) Current running through the DF-ECC processor recorded by measuring the voltage drop via a resistor in series with the board power pin and FPGA power pin.

Table 3. FPGA Implementation Results

Design	Area (Slices)	f_{max} (MHz)	Field Arithmetic
I	7,573 (32%)	27.7	Radix-2 Montgomery
II	8,158 (34%)	27.7	Radix-2 Randomize Montgomery
II	9,828 (41%)	20.2	Radix-4 Montgomery
IV	10,460 (43%)	20.2	Radix-2 Randomized Montgomery

Power Analysis

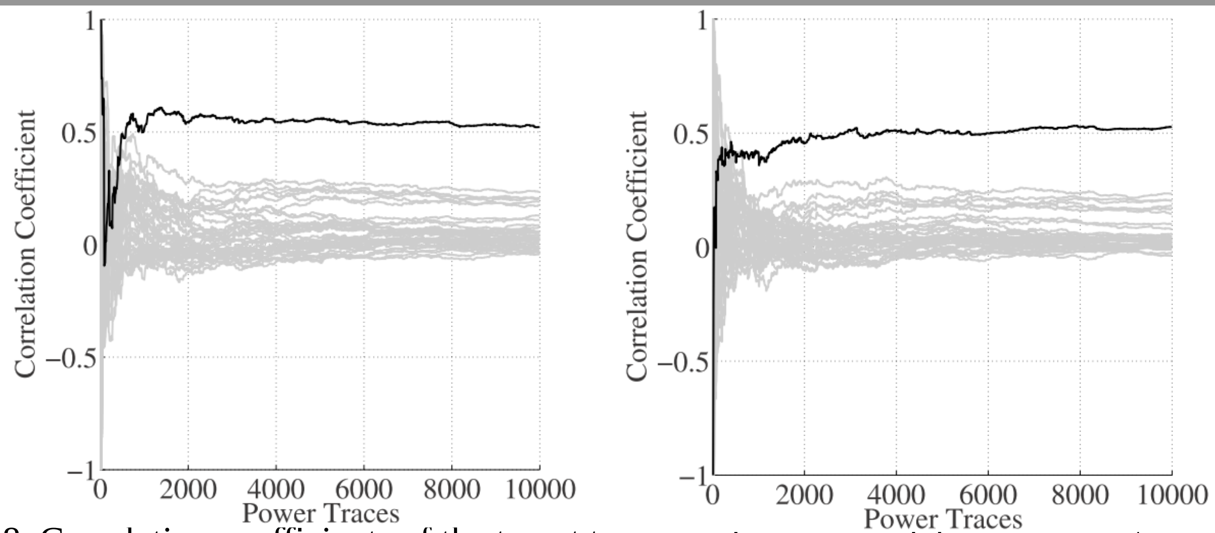


Fig. 8. Correlation coefficients of the target traces and power model over power traces obtained from the (L) Design-I (R) Design-III performing arithmetic in a fixed domain.

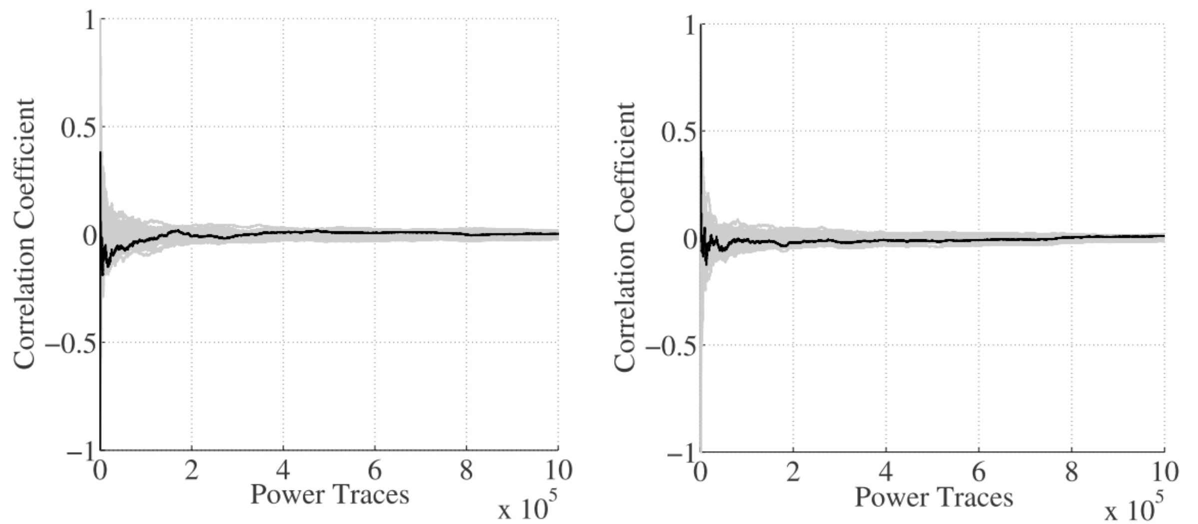


Fig. 9. Correlation coefficients of the target traces and power model over power traces obtained from the (L) Design-II (R) Design-IV performing arithmetic in a randomized domain.

Performance and Comparison

Table 4. Implementation Results Compared with Related Works

	CMOS Process	Length	Area (mm^2)/ KGates	Finite Field	f_{max} (MHz)	Time(ms/ ECSM)	Energy (μJ/ECSM)	AT Product
Ours (Radix-2)	90-nm	160	0.21/61.3	$GF(p_{160})$	277	0.71	11.9	1
				$GF(2^{160})$	277	0.61	9.6	1
Ours (Radix-4)	90-nm	160	0.29/83.2	$GF(p_{160})$	238	0.43	11.2	0.82
				$GF(2^{160})$	238	0.39	8.97	0.87
TCAS-II'09 [5]	130-nm	160	1.44/169	$GF(p_{160})$	121	0.61	42.6	1.63*
				$GF(2^{160})$	146	0.37	30.5	1.16*
Ours (Radix-2)	90-nm	521	0.58/168	$GF(p_{521})$	250	8.08	452	1
				$GF(2^{409})$	263	4.65	246	1
Ours (Radix-4)	90-nm	521	0.93/265	$GF(p_{521})$	232	4.57	435	0.89
				$GF(2^{409})$	238	2.77	238	0.94
ESSCIRC'10 [9]	90-nm	521	0.55/170	$GF(p_{521})$	132	19.2	1,123	2.40
				$GF(2^{409})$	166	8.2	480	1.78

* Technology scaled area-time product = gates $\times$ (time $\times$ f), where f = 90-nm/130-nm.

Performance and Comparison

Table 5. Overhead for CPA Resistance

	Ours (Radix-2)	Ours (Radix-4)	ESSCIRC'10 [9]	JSSC'06 [12]	JSSC'10 [13]
Design	521 DF-ECC	521 DF-ECC	521 DF-ECC	128 AES	128AES
Area	4.3%	3.6%	10%	210%	7.2%
Time	0	0	14.0% [a]	288%	100%
Energy	5.2%	3.8%	20.8% [b]	270%	33%

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

a. Estimated by cycle count × clock period.

b. Estimated by operation time × average power.

Conclusion

- An efficient CPA-resistant DF-ECC processor supporting arbitrary modulus is presented
  - no need to modify ASIC or FPGA design flow
  - applicable to IEEE P1363
  - low overhead (< 5%) for hardware speed, area, power

Q and A

Thanks for Your Attention!

References

[1] Koblitz, N.: Elliptic Curve Cryptosystems. Math. Comp., 2001

[2] Miller, V.: Uses of Elliptic Curves in Cryptography. CRYPTO’85, 1986

[3] McIvor, C. J. et al: Hardware Elliptic Curve Cryptographic Processor over GF(p). IEEE Trans. Circuits Syst. I, 2006

[4] Sakiyama, K. et al: Multicore Curve-Based Cryptoprocessor With Recon-figurable Modular Arithmetic Logic Units over GF(2^n). IEEE Trans. Comput., 2007

[5] Lai, J.-Y., Huang, C.-T.: A Highly Efficient Cipher Processor for Dual-Field Elliptic Curve Cryptography. IEEE Trans. Circuits Syst. II, 2009

[6] Chen, J.-H. et al : A High-Performance Unified-Field Reconfigurable Cryptographic Processor. IEEE Trans. VLSI Syst., 2010

[7] Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. CRYPTO’99, 1999

[8] Montgomery, P.: Speeding the Pollard and Elliptic Curve Methods of Factorization. Math. Comp., 1987

[9] Lee, J.-W. et al : A 521-bit Dual-Field Elliptic Curve Cryptographic Processor With Power Analysis Resistance. ESSCIRC’10, 2010

[10] Brier, E., Clavier, C., Olivier, F.: Correlation Power Analysis With a Leakage Model. CHES’04, 2004

[11] IEEE: Standard Specifications or Public-Key Cryptography. IEEE Std. 1363, 2000

[12] Hwang, D. et al: AES-Based Security Coprocessor IC in 0.18-μm CMOS With Resistance to Differential Power Analysis Side-Channel Attacks. IEEE J. Solid-State Circuits, 2006

[13] Tokunaga, C., Blaauw, D.: Securing Encryption Systems With a Switched Capacitor Current Equalizer. IEEE J. Solid-State Circuits, 2010

[14] Liu, P.-C. et al: A True Random-Based Differential Power Analysis Countermeasure Circuit for an AES Engine. IEEE Trans. Circuits Syst. II, 2012

[15] Coron, J.: Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. CHES’99, 1999

[16] Joye, M., Tymen, C.: Protections against Differential Analysis for Elliptic Curve Cryptography – An Algebraic Approach. CHES’01, 2001

[17] Montgomery, P.: Modular Multiplication Without Trial Division. Math. Comp., 1985

[18] Kaliski, B.: The Montgomery Inverse and Its Applications. IEEE Trans. Comput., 1995

[19] Cohen, H., Miyaji, A., Ono, T.: Efficient Elliptic Curve Exponentiation Using Mixed Coordinates. ASIACRYPT’98, 1998

[20] Golic, J.D.: New Methods for Digital Generation and Postprocessing of Random Data. IEEE Trans. Comp., 2006

[21] Chen, Y.-L. et al: A Dual-Field Elliptic Curve Cryptographic Processor With a Radix-4 Unified Division Unit. ISCAS’11, 2011

References

[HWANG'06] D. Hwang, et al., "AES-Based Security Coprocessor IC in 0.18-µm CMOS With Resistance to Differential Power Analysis Side-Channel Attacks," IEEE J. Solid-State Circuits, 2006

[SAEKI'09] M. Saeki, D. Suzuki, K. Shimizu, and A. Satoh, "A design methodology for a DPA-resistant cryptographic LSI with RSL techniques," in Cryptographic Hardware and Embedded Systems (CHES'09), vol. 5747, 2009, pp. 189–204.

[CORON'99] J. Coron, "Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems," in Cryptographic Hardware and Embedded Systems (CHES'99), 1999

[ITOH'03] K. Itoh, T. Izu, and M. Takenaka, "A practical countermeasure against address-Bit differential power analysis," in Cryptographic Hardware and Embedded Systems (CHES'03), vol. 2779, 2003, pp. 382–396.

[JOYE'01] M. Joye and C. Tymen, "Protections against differential analysis for elliptic curve cryptography – an algebraic approach," in Cryptographic Hardware and Embedded Systems (CHES'01), vol. 2162, 2001, pp. 377–390.

[CORON'09] J.-S. Coron and I. Kizhvatov, "An efficient method for random delay generation in embedded software," in Cryptographic Hardware and Embedded Systems (CHES'09), vol. 5747, 2009, pp. 156–170.