2016 11th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)

Residue Number System as a Side Channel and Fault Injection Attack countermeasure in Elliptic Curve Cryptography

Apostolos P. Fournaris
Computer Engineering
& Informatics Dpt.
University of Patras, Greece
Email: apofour@ieee.org

Louiza Papachristodoulou, Lejla Batina
Digital Security Group,
Radboud University Nijmegen,
The Netherlands
Email: {louizap, lejla@cs.ru.nl}

Nicolas Sklavos
Computer Engineering
& Informatics Dpt.
University of Patras, Greece
Email: nsklavos@ceid.upatras.gr

***Abstract*—Implementation attacks and more specifically Power Analysis (PA) (the dominant type of side channel attack) and fault injection (FA) attacks constitute a pragmatic hazard for scalar multiplication, the main operation behind Elliptic Curve Cryptography. There exists a wide variety of countermeasures attempting to thwart such attacks that, however, few of them explore the potential of alternative number systems like the Residue Number System (RNS). In this paper, we explore the potential of RNS as an PA-FA countermeasure and propose an PA-FA resistant scalar multiplication algorithm and provide an extensive security analysis against the most effective PA-FA techniques. We argue through a security analysis that combining traditional PA-FA countermeasures with lightweight RNS countermeasures can provide strong PA-FA resistance.**

I. Introduction

Scalar multiplication (SM), the main mathematical operation behind Elliptic Curve Cryptography (ECC) is the target of a broad range of possible PAs and FAs on ECC [11] [15] of both horizontal and vertical nature [7]. RNS is an arithmetic representation that is advantageous when it comes to parallel arithmetic calculations and has considerable potentials as an PA/FA countermeasure [2] [4] [19] [16]. However, RNS has small adoption from the research community due to the complexity of its arithmetic and the high number of employed hardware resources to implement it. In RNS, a GF(p) number is represented by a given moduli base (RNS base) consisting of several base elements. Randomizing such base elements once per SM or in every GF(p) multiplication during SM can provide disassociation of secret information to physical leakage. Unfortunately, RNS in an ECC implementation leads to considerable computational complexity and hardware resources [16].

In this paper, we introduce a SM algorithm that uses RNS as an add-on PA-FA countermeasure that in comparison to previous RNS proposals does not use redundant RNS modulo as a FA countermeasure and adopts a well balanced use of the ”base permutation technique” (also known as leak resistant arithmetic [2]) as an efficient PA countermeasure. The level of security that these approaches can offer, when specific PA and FA attacks are applied to an ECC implementation, is evaluated and a roadmap of RNS based countermeasures is described. Extending the work of [16] we argue through a security analysis that a combination of traditional PA/FA countermeasures [10] [15] [12] and RNS based structures can provide strong side-channel resistance.

The rest of this paper is structured as follows. In section II, RNS is introduced and its PA and FA resistance in ECC systems is discussed. In section III, new RNS based PA-FA algorithms are proposed. In section IV, a security analysis of our proposal is made and section V concludes the paper.

II. RNS for EC Point operations

A number x can be represented in RNS as a set of n moduli x_i ($x \overset{RNS}{\rightarrow} X : (x_1, x_2, ...x_n)$) of a given RNS basis $B : (m_1, m_2, ...m_n)$ as long as $0 \leq x < M$ where $M = \prod_{i=1}^{n} m_i$ is the RNS dynamic range and all m_i are pair-wise relatively prime. Each x_i can be derived from x by calculating $x_i = \langle x \rangle_{m_i} = x \ mod \ m_i$. Assuming that we have two numbers a and d represented in RNS as $A : (a_1, a_2, ...a_n)$ and $D : (d_1, d_2, ...d_n)$, we can contain all arithmetic operations in RNS as $A \oslash D = (\langle a_1 \oslash d_1 \rangle_{m_1}, ... \langle a_n \oslash d_n \rangle_{m_n})$ where $\oslash : (+, -, \times)$

Binary reconstruction from its RNS representation can be done using the Chinese Remainder Theorem (CRT) $x = \left\langle \sum_{i=1}^{n} \langle x_i \cdot M_i^{-1} \rangle_{m_i} \cdot M_i \right\rangle_M$ where $M_i = \frac{M}{m_i}$ and M_i^{-1} is the multiplicative inverse of M_i. The required M modulo reduction, due to the high bit length of M , is not efficiently realized and is usually performed by introducing a correction factor w as shown in $x = \sum_{i=1}^{n} \langle x_i \cdot M_i^{-1} \rangle_{m_i} \cdot M_i - w \cdot M$ To avoid the above process, x’s Mixed Radix System (MRS) representation $\widetilde{X} : (u_1, u_2, ...u_n)$ can be used for RNS to binary conversion [3] [5].

For ECC approved ECs defined over GF(p) (ECs on $GF(2^k)$ are not discussed in this paper), all GF(p) operations (addition, subtraction, multiplication) are modular operations (modulo p). Performing RNS $GF(p)$ addition or subtraction can be easily realized by expressing p in RNS format i.e. $P : (p_1, p_2, p_3, ...p_n)$ and calculating for each moduli i $\langle \langle a_i \pm d_i \rangle_{m_i} \rangle_{p_1 i}$

978-1-5090-0336-5/16/$31.00 ©2016 IEEE

However, RNS GF(p) multiplication is a computationally difficult operation. It is usually realized through the RNS Montgomery multiplication algorithm that involves base extension operations [3] (increasing its complexity). Assuming that we introduce two RNS bases $B_n = (m_1, m_2, \ldots, m_n)$ and $\acute{B_n} = (m_{n+1}, m_{n+2}, \ldots, m_{2n})$ such that $gcd(m_i, m_j) = 1$ for all $i \in \{1, n\}$ and $j \in \{n+1, 2n\}$, we express a GF(p) number x in base B_n or $\acute{B_n}$ as X_B and $X_{\acute{B}}$ respectively, while in both RNS bases as $X_{B \cup \acute{B}}$. We also define $M_B = \prod_{i=1}^{n} m_i$ and M_B^{-1} as the multiplicative inverse of M_B in base B_n, as well as $M_{\acute{B}} = \prod_{i=n+1}^{2n} m_i$ and $M_{\acute{B}}^{-1}$ as the multiplicative inverse of $M_{\acute{B}}$ in base $\acute{B_n}$. The RNS Montgomery multiplication (RNSMM) as an outcome calculates $S_B = A \cdot B \cdot M_B^{-1}\ modp$ and $S_{\acute{B}} = A \cdot B \cdot M_{\acute{B}}^{-1}\ modp$. Base extension from one base to the other in RNSMM is needed, since M_B^{-1} does not exist in base B_n and therefore computations must be migrated to the $\acute{B_n}$ base to come up with S_B.

In the first step of RNSMM Base extension operation, the base B_n RNS number is converted into a base B_n MRS number. In the second step, the base B_n MRS number is converted into a base $\acute{B_n}$ RNS number. A similar two step procedure is followed for base extension from $\acute{B_n}$ to B_n respectively to provide a correct RNSMM outcome.

Each RNS number A must be in the Montgomery format ($A_B \cdot M_B\ modP_B$ or $A_{\acute{B}} \cdot M_{\acute{B}}\ modP_{\acute{B}}$). So, initially an RNSMM must be performed between A and $M_{B \cup \acute{B}}\ modP_{B \cup \acute{B}}$ using the bases B_n and $\acute{B_n}$ in reverse order (i.e. $RNSMM(A, M_{B \cup \acute{B}}\ modP_{B \cup \acute{B}}, P, \acute{B_n}, B_n)$). Montgomery domain normalization can be removed through an RNSMM of the Montgomery formatted RNS number A with 1. To increase computation efficiency, most studies on optimal base moduli [5] agree that moduli of the form $2^k \pm c_i$, $2^k - 2^{t_i} \pm 1$ or 2^k, $2^k - 1$, $2^{k-1} - 1$ $2^{k+1} - 1$ (Mersenne numbers) for various i values provide high performance results.

A. Using RNS for PA and FA resistance

Bajard et al. in [2] proposes, originally for modular exponentiation, a random permutation of the base B_n and $\acute{B_n}$ moduli for PA resistance thus creating $\binom{2n}{n}$ random permutations of B_n and $\acute{B_n}$. We denote each such RNS Base γ permutation as $B_{n,\gamma}$ and $\acute{B}_{n,\gamma}$. The periodic change of a base permutation during the modular exponentiation (and consecutively SM) computation flow can introduce enough randomness to thwart PAs. This leak resistant arithmetic (LRA) technique can be applied to modular exponentiation designs (used for RSA) either by choosing a new base permutation once at the beginning of each modular exponentiation or by changing a permutation in each RNSMM operation of the exponentiation process. The base transition of an RNS number A represented in a base permutation γ to a new permutation $\acute{\gamma}$ can be done by performing two consecutive RNSMMs. Initially $A_1 = RNSMM(A, M_{B \cup \acute{B}}\ mod\ P_{B \cup \acute{B}}, P, \acute{B}_{n,\acute{\gamma}}, B_{n,\acute{\gamma}})$ [1] is performed and it is followed by $RNSMM(A_1, 1, P, \acute{B}_{n,\gamma}, B_{n,\gamma})$ (Random Base Permutation operation, RBP)

Some attempts to introduce LRA in SM have been made in [19], however, they are applicable only to the CRT type of base extension using the Cox-Rower method when pseudo-Mersenne numbers are used for base moduli. In SM, a permutation transition can be done only once (per SM), in every round of the SM process or before an $GF(p)$ RNSMM operation of each point operation of every round. Taking into account that the transition from one permutation to another costs 2 RNSMM, the third approach is not affordable in terms of speed. The first approach, providing a single randomization per SM may be vulnerable to horizontal PA attacks (depending on the employed implementation methodology) so the second approach is the best option promising balance between performance and PA resistance strength.

To achieve RNS based fault detection during RNSMM [4], in the existing two RNS bases moduli B_n and $\acute{B_n}$, a redundant moduli m_r is added, thus executing RNSMM using redundant bases $B_n \cup m_r$ and $\acute{B_n} \cup m_r$. The redundant RNSMM algorithm results $S_{B \cup m_r}$ and $S_{\acute{B} \cup m_r}$ include moduli related to base element m_r If no fault is injected during an RNSMM then the 2 moduli must be the same. This approach is capable of detecting a single fault during a RNSMM and bares an additional performance cost (compared to the original RNSMM) in the RNS Base extension operations. The technique is applied in [19] only to Cox-Rower RNSMM designs (using CRT base extension method) and later is generalized for base extension approach in [4].

III. FA and PA Resistant Scalar Multiplication

Given the description of RNS PA and FA countermeasures, we propose the inclusion of LRA as an add-on countermeasure in an PA resistant SM algorithm in order to provide horizontal (eg. simple PAs) apart from vertical attacks resistance. In the proposed algorithm (Algorithm 1), LRA is combined with the base point blinding technique (additive randomization of the EC base point V) in the Montgomery Power Ladder (MPL) algorithm (MPL is considered secure against most vertical and horizontal attacks) expanding the work of [17] and [16].

In Algorithm 1, we introduce LRA RNS base randomization once in each SM round (steps 4c and 4d) and in that way manage to include a different randomization element in every round. The input point V is initially blinded by adding to it a random element R, thus preventing sophisticated, comparative simple PAs [12]. MPL is a highly regular SM algorithm (it always performs 2 point operations per round regardless of the scalar bit e_i) and also provides an intrinsic fault detection mechanism based on the mathematical coherence of R_0 and R_1. As observed in [20] and by Giraud in [18], the R_0 and R_1 points in an MPL round always satisfy the equation $R_0 = V + R_1$. Injecting a fault during computation in an R_1 or R_0 variable will ruin this coherence and by introducing an MPL coherence detection mechanism in the end of the MPL algorithm, this fault will always be detected. This technique is adopted in step 6 of Algorithm 1 where $R_0 + V \neq R_1$ if

[1] Note that A has the form $A_1 \cdot M_{B_{n,\gamma}}^{-1}\ mod\ P$ since it is an output of some previous RNSMM

a fault is injected. Note that the correct result is unblinded only after the fault detection mechanism, in order to provide protection against possible bypassing (by injecting a second fault) of the fault detection countermeasure.

Algorithm 1. LRA PA-FA Blinded MPL algorithm

```
Input: EC base point $V$, random point $R \in EC(GF(p))$, $e = (e_{t-1}, e_{t-2}, ...e_0)$
1. Choose random initial base permutation $\gamma_t$. Transform V, R to RNS format using $\gamma_t$ permutation
2. $R_0 = R$, $R_1 = R + V$, $R_2 = -R$
3. $CMF(R_0, R_1, R_2, \acute{B}_{n,\gamma_t}, B_{n,\gamma_t})$
4. For $i = t - 1$ to $0$
  (a) $R_2 = 2R_2$,
  (b) choose a random base permutation $\gamma_i$
  (c) $RBP(R_0, B_{n,\gamma_{i+1}}, \acute{B}_{n,\gamma_{i+1}}, B_{n,\gamma_i}, \acute{B}_{n,\gamma_i})$
  (d) $RBP(R_1, B_{n,\gamma_{i+1}}, \acute{B}_{n,\gamma_{i+1}}, B_{n,\gamma_i}, \acute{B}_{n,\gamma_i})$
  (e) if $e_i = 1$
      $R_0 = R_0 + R_1$ and $R_1 = 2R_1$
    else
      $R_1 = R_0 + R_1$ and $R_0 = 2R_0$
    end if
5. $RBP(V, B_{n,\gamma_t}, \acute{B}_{n,\gamma_t}, B_{n,\gamma_0}, \acute{B}_{n,\gamma_0})$
6. If ($i$ and $e$ are not modified and $R_0 + V = R_1$)
  then
    (a) $RBP(R_0, B_{n,\gamma_0}, \acute{B}_{n,\gamma_0}, B_{n,\gamma_t}, \acute{B}_{n,\gamma_t})$
    (b) return $R_0 + R_2$
  else return error
```

Conversion to Montgomery Format (CMF) operation is used for transforming all EC point coordinates into the Montgomery format, so that RNSMM can be performed correctly. This conversion will require 9 RNSMMs (all points are in projective coordinate representation). The RBP function performs base transformation from base permutation γ to permutation $\acute{\gamma}$ and requires 6 RNSMMs. The RBP function is executed in each MPL round once for point R_0 and once for R_1.

IV. Security Analysis

A. Power Analysis Attack Resistance

The approach of constant number and type of point operations per round (being a vital part of MPL) proposed in Algorithm 1 provides SPA protection. It can be further enhanced through the use of elliptic curves with unified formulas for addition and doubling like Edwards curves [8] or the recent results from Renes et al. [23] proposing complete addition formulas for every prime order short GF(p) based Weierstrass curve (char(GF(p))$\neq 2, 3$).

Regarding horizontal attacks that are focused on a single collected trace decomposition in sample time blocks and analysis per block, Feix et al. presented in [13] a powerful attack against blinded SM algorithms. This attack cannot be applied in our implementation, because it requires collisions from vertical attacks, when a dummy point addition is performed. There are no dummy operations in our algorithm. For their horizontal scenario, they find leakage between doubling and adding operations in two consecutive rounds; in our case the random base point is involved in each round. The horizontal attack of Bauer et al. [7] is based on splitting an element of GF(p) in words and finding correlation between them. Since those elements are represented in their RNS form, we expect the corresponding correlations to reveal no useful information.

MPL is not resistant against refined PA (RPA) or zero-value point attacks (ZVP) even if applying randomization of the projective coordinate by multiplying with a random number, and applying EC or field random isomorphisms [10]. However, in Algorithm 1, base point randomization is performed additively (Base point blinding), so the above mentioned attacks become unsuccessful. Using only RBP without base point blinding would not sufficiently protect against ZVP attacks. For comparative SPA attacks, MPL (and consecutively Algorithm 1) is resistant to Doubling attack (DA) [14], but not against relative DA (RDA) [24] or 2-Torsion Attack (2-TorA) [25]. Base Point blinding, if applied statically (e.g. the same random number is added in each round, BRIP method [21]), cannot thwart RDA and 2-TorA [1]. However, in Algorithm 1, the base point randomization is extended in every algorithmic round. A different randomization number (a multiple of R) is added an i round's $k_i \cdot V$ or $(k_i + 1) \cdot V$ thus effectively preventing RDA and 2-TorA.

In MPL like algorithms, an attacker can recover e_i by observing in which register (R_0 or R_1) the point addition outcome is saved. This is not possible in Algorithm 1, since in each round a non dummy value storage operation is done in parallel to all registers thus masking a specific register storage power trace (it can't be discriminated from the rest). Furthermore, since we are using RNS arithmetic, R_0, R_1 and R_2 consist of n different values each (one for each modulo). Each value is stored in a different register that adds to the complexity of discriminating the n storage operations of R_1 from the n storage operations of R_0.

Regarding DPA attacks, countermeasures are based on randomization during the SM process [11] [10] as adopted in Algorithm 1 (base point blinding, Coron second countermeasure). As long as point operations in SM rounds remain fully balanced (same point operations number per round, same point operations execution order per round for all rounds) and the random point R is not a weak mask (i.e. a randomization that can lead to unmasking (un-blinding) the point P in an intermediate SM round), then base point blinding remains a strong DPA countermeasure [11]. The above remark is true for Algorithm 1 constituting our proposal DPA resistant. This property is further ensured by the use of LRA and can be enhanced by uniform group law based EC arithmetic. The template attack of [22] is not successful in our scenario, because it uses an offline DPA phase; since we use base point blinding and RBP in each SM round, DPA cannot be applied. For the same reason Online Templete Attacks [6] should be also not possible, since point blinding makes the traces of mV and $(m + 1)V$ look random (where m and $m + 1$ are specific values for scalar e).

B. Fault Analysis Attack Resistance

Algorithm 1 fault protection mechanism is focused on FAs during SM, not aiming at weak curve attacks. This mechanism consists of infective computation and fault detection, base point blinding (randomization), LRA and RNS fault diffusion. Infective computation is an inherited characteristic of the Algorithm's 1 adopted MPL and its main goal is to propagate an injected fault through the SM process so that it will be always detected by the fault detection mechanism. Fault detection takes advantage of the MPL mathematical coherency of $R_1 - R_0 = P$ for every SM round and evaluates this equality at the end of calculations before removing randomization and releasing the result. The fact that all operations are performed in RNS, enhances SM fault diffusion. Due to the iterative use of base extension functions in RNSMM (as part of each point operation coordinate calculations) even a single fault (eg. a single bit flip on R_0 or R_1) will cause a change in the whole RNS number (in all this number's moduli). The fault will propagate uncontrollably through Algorithm's 1 execution thus considerably affecting all the computations and will eventually be detected in the Algorithm's 1 step 6 fault detection mechanism thus avoiding the need for an additional, redundant, RNS moduli to detect the fault during a single RNSMM [2], [4].

C-safe error and sign change fault attacks do not apply to the proposed approach since Algorithm 1 has no dummy operations and does not use scalar Non-adjacent form (NAF). The proposed approach is also protected against M-safe error attacks (usually successful against MPL) since the RNS computations are performed in parallel and all bits stored in RNS registers are computed concurrently. Thus, an M-safe error attack will always alter the R_0 or R_1 outcome and will be detected. Differential FAs like the Biehl-Meyer-Muller attack [9] are not successful in the proposed scheme due to the adopted fault detection mechanism which is also effective against multiple fault injection during algorithmic execution.

V. Conclusion

This paper introduced a new MPL algorithm based on the combination of RNS and LRA arithmetic. Our proposal for a random base permutation instead of exchanging between two fixed base extensions provides PA-FA resistance against a wide range of such attacks. An PA-FA security analysis against the most potent attacks proves our claim that the combination of traditional PA-FA countermeasures with RNS arithmetic inclusion can provide strong resistance against PA-FA attacks.

Acknowledgment

This work is supported by EU COST IC1204 TRUDEVICE, through the Short Term Scientific Mission, of L. Papachristodoulou, to the Computer Engineering & Informatics Dept, University of Patras, Greece, Winter 2016.

References

[1] F. Amiel and B. Feix. On the BRIP algorithms security for RSA. *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, pages 136–149, May 2008.

[2] J. Bajard, L. Imbert, P. Liardet, and Y. Teglia. Leak resistant arithmetic. *CHES*, 3156, 2004.

[3] J.-C. Bajard, L.-S. Didier, and P. Kornerup. An RNS Montgomery modular multiplication algorithm. In *Proc.13th IEEE Symp. on Comp. Arithmetic*, pages 234–239. IEEE Comput. Soc, 1997.

[4] J.-C. Bajard, J. Eynard, and F. Gandino. Fault Detection in RNS Montgomery Modular Multiplication. In *IEEE 21st Symp. on Comp. Arithmetic*, pages 119–126. IEEE, Apr. 2013.

[5] J. C. Bajard, M. Kaihara, and T. Plantard. Selected RNS Bases for Modular Multiplication. In *2009 19th IEEE Symp. on Comp. Arithmetic*, pages 25–32. IEEE, June 2009.

[6] L. Batina, L. Chmielewski, L. Papachristodoulou, P. Schwabe, and M. Tunstall. Online template attacks. In *proc. of 15th International Conference on Cryptology in India INDOCRYPT 2014, New Delhi, India, Dec. 14-17, 2014*, pages 21–36, 2014.

[7] A. Bauer, E. Jaulmes, E. Prouff, and J. Wild. Horizontal collision correlation attack on elliptic curves. In *Selected Areas in Cryptography – SAC 2013*, volume 8282 of *Lecture Notes in Computer Science*, pages 553–570. Springer Berlin Heidelberg, 2014.

[8] D. Bernstein and T. Lange. Faster addition and doubling on elliptic curves. *Advances in cryptology ASIACRYPT 2007*, pages 1–22, 2007.

[9] I. Biehl, B. Meyer, and V. Mller. Differential Fault Attacks on Elliptic Curve Cryptosystems. In *Advances in Cryptology, CRYPTO 2000*, volume 1880 of *LNCS*, pages 131–146. 2000.

[10] J.-S. Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In *Proc. of CHES '99*, pages 292–302, London, UK, 1999. Springer-Verlag.

[11] J. Fan, X. Guo, E. De Mulder, P. Schaumont, B. Preneel, and I. Verbauwhede. State-of-the-art of secure ECC implementations: a survey on known side-channel attacks and countermeasures. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust*, pages 76–87. IEEE, June 2010.

[12] J. Fan and I. Verbauwhede. An updated survey on secure ECC implementations: Attacks, countermeasures and cost. *Cryptography and Security: From Theory to Applications*, 6805:265–282, Jan. 2012.

[13] B. Feix, M. Roussellet, and A. Venelli. Side-channel analysis on blinded regular scalar multiplications. Cryptology ePrint Archive, Report 2014/191, 2014.

[14] P.-A. Fouque and F. Valette. The doubling attack: Why upwards is better than downwards. In *CHES 2003*, pages 269–280.

[15] A. Fournaris and N. Sklavos. Public key cryptographic primitive design and protection against fault and power analysis attacks. In *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices*. DATE 2015 Conference, 2015.

[16] A. P. Fournaris, N. Klaoudatos, N. Sklavos, and C. Koulamas. Fault and power analysis attack resistant RNS based edwards curve point multiplication. In *Proceedings of the 2nd Workshop on Cryptography and Security in Computing Systems, CS2 at HiPEAC 2015, Amsterdam, Netherlands, January 19-21, 2015*, pages 43–46, 2015.

[17] G. Fumaroli and D. Vigilant. Blinded fault resistant exponentiation. In L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, editors, *FDTC*, volume 4236 of *LNCS*, pages 62–70. Springer, 2006.

[18] C. Giraud. An rsa implementation resistant to fault attacks and to simple power analysis. *IEEE Trans. on Computers*, 55(9):1116–1120, 2006.

[19] N. Guillermin. A coprocessor for secure and high speed modular arithmetic. *IACR Cryptology ePrint Archive*, 2011.

[20] M. Joye and S.-M. Yen. The Montgomery Powering Ladder. In *4th CHES 2003*, pages 291–302, UK, 2003. Springer-Verlag.

[21] H. Mamiya, A. Miyaji, and H. Morimoto. Efficient countermeasures against RPA, DPA, and SPA. *Cryptographic Hardware and Embedded Systems-CHES 2004*, 3156:243–319, 2004.

[22] M. Medwed and E. Oswald. Template attacks on ECDSA. In K.-I. Chung, K. Sohn, and M. Yung, editors, *Information Security Applications*, volume 5379, pages 14–27, 2009.

[23] J. Renes, C. Costello, and L. Batina. Complete addition formulas for prime order elliptic curves. Cryptology ePrint Archive, Report 2015/1060, 2015.

[24] S. Yen, L. Ko, S. Moon, and J. Ha. Relative doubling attack against Montgomery Ladder. *Information Security and Cryptology-ICISC*, pages 117–128, 2006.

[25] S.-M. Yen, W.-C. Lien, S.-J. Moon, and J. Ha. Power Analysis by Exploiting Chosen Message and Internal Collisions - Vulnerability of Checking Mechanism for RSA-Decryption. In *Mycrypt*, volume 3715 of *LNCS*, pages 183–195. Springer, 2005.