Journal of Advances in Mathematics and Computer Science

29(1): 1-12, 2018; Article no.JAMCS.43056

ISSN: 2456-9968
(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

Error Analysis and Detection Procedures for Signature and Authentication Schemes

Naglaa F. Saudy[1*], Ihab A. Ali[2] and Reda Al Barkouky[3]

[1]Department of Physics and Applied Mathematics, Faculty of Engineering, Helwan University, Cairo, Egypt.
[2]Department of Communications Engineering, Helwan University, Cairo, Egypt.
[3]Egyptian Chinese University, Egypt.

Authors' contributions

This work was carried out in collaboration between all authors. Author NFS designed the study, performed the statistical analysis, wrote the protocol and wrote the first draft of the manuscript. Authors IAA and RAB managed the analyses of the study. Author IAA managed the literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2018/43056
Editor(s):
(1) Dr. Metin Basarir, Professor, Department of Mathematics, Sakarya University, Turkey.
Reviewers:
(1) Anand Nayyar, Duy Tan University, Vietnam.
(2) Zhaneta Nikolova Savova-Tasheva, National Military University "V. Levski" and University of Shumen, Bulgaria.
(3) Danjuma Samaila, Adamawa State University, Nigeria.
Complete Peer review History: http://www.sciencedomain.org/review-history/26416

Received: 28 May 2018
Accepted: 13 August 2018
Published: 27 September 2018

Original Research Article

Abstract

In this paper, a fault discovery structure is presented to increase the protection and reliability of the Elliptic Curve Digital Signature Algorithm (ECDSA) under practical considerations. As the ECDSA will work in real systems, which have their own arrangement of transient errors, being able to handle faults that happen when examining the ECDSA execution turns into an unquestionable requirement. Since even one transient mistake was to occur amid the ECDSA procedure, will bring out enormous errors in the information. We introduce applying nonlinear fault detection codes to protect ECDSA operations against fault attacks. We also apply the same idea to protect Guillou-Quisquater authentication scheme (GQ) against fault injection attacks. These codes give almost perfect error detection capacity (aside from an exponentially small probability) at sensible overhead. We present a fault detection scheme by using the nonlinear error detecting code. This fault detection scheme has shown to have over 99% fault detection coverage.

Keywords: Elliptic curve cryptography; fault injection attacks; attacks on signature and authentication; fault detection.

*Corresponding author: E-mail: naglaa_eng2003@yahoo.com;

1 Introduction

DSA [1] (Digital Signatures Algorithms) are the pillars of today's cryptosystems. Their primary use is to Prior to the digital age, handwritten signatures were relied on exclusively to guarantee the sender's identity. They are digital counterparts of handwritten signatures. The basic objective of a digital signature is to guarantee authenticity and integrity of a signed message transmitted to the receiver and to unequivocally guarantee the identity of the transmitter. ECDSA is an alternative to DSA which uses ECC (Elliptic Curve Cryptography) [2]. ECC is valued because it generates shorter signatures than other methods such as RSA (Rivest, Shamir, Adleman) and this is invaluable for use on applications with limited capacity devices [3-5].

Traditionally, the ability to defend public key authentication and signature protocols have been interconnected with challenges related to the factorization and discrete logarithm problems. If these types of algorithms are actually implemented, it may lead to a Pandora's Box of side channel attacks such as fault attacks that target private parameters [6,7]., other attacks mounted against symmetric ciphers in both block [8] and stream modes [9], and with a recent addition of strikes against the common parameters of public key cryptosystems [10,11]. The major threat is one or a combination of different types of attacks may breach defences and allow data from smart cards to be scooped up [12,13].

A most extraordinary new method of attacking RSA by Brier et al. [10] showed how to attack RSA by induction of faults in public modulus n, without making any assumption about the concrete model of modulus corruption. This approach allowed an attacker to achieve full recovery of a secret key stored in a secure device with no knowledge of the fault behaviour. The most simple and basic method to breach the system security, and remember, this is with no need of special knowledge, is to influence the change of a single change a word in an unknown way. When public key components stored in EEPROM or in Flash are transferred to RAM, this is easily achievable.

The same technique can be used to attack other asymmetric cryptographic schemes. This has been illustrated by attacking the typically more vulnerable public elements and ignoring the more secret components. A unique assault was the attack on Elliptic Curves based signature protocol (ECDSA). It was shown it is possible to recover the private key by introducing a fault in only one of the two different moduli involved in the scheme with the use of a nontrivial calculation. This same idea worked in the attack on the Guillou-Quisquater authentication scheme.

Fault detection [14-17] is a necessity to avoid malicious attacks and to transfer sensitive data like a secret key. Both private-key cryptosystems like AES and public-key cryptosystems like RSA and EEC have had fault-Detection-based countermeasures produced for them. Fault detection in AES is used to identify errors prior to transmission and to deter the possible use of misinformation [18,19]. Fault detection strategies are practical to several procedures of complex and nonhomogeneous cryptographic algorithms like AES [20-24]. RSA likewise is liable to fault attack so in [25-33] there are several fault detection procedures introduced. For ECC, fault detection procedures are displayed in [34-41].

Protection from fault attacks is best done with fault detection. To that end, this paper introduces a fault detection method based on applying systematic nonlinear error detection codes [34] to protect ECDSA and GQ authentication operations in order to counter to fault attacks. The security level delivered by this error detection method is investigated and it is demonstrated that these codes, except for an exponentially low probability, provide near faultless error detection.

The organization of the remainder of this article is as follows: The second section puts forward the fundamental survey data on the robust nonlinear residue as applied in our detection method. In the third section on how to protect the ECDSA scheme by applying nonlinear fault detection is demonstrated, including a short survey of the ECDSA scheme and fault attacks on it. In the fourth section demonstrates how to transfer this knowledge to the Guillou-Quisquater authentication scheme. In the fifth and final section, the results of this article are concluded and summarized.

2 Survey on the Robust Nonlinear Residue Codes

In [42], Karpovsky and Taubin analyzed a novel class of non-linear error-detection cyphers. These cyphers allow minimising the general (n, k, r) codes of maxima regarding undetectable faults. The (n, k, r) cyphers can be defined as coding structures where k stands for the bit length of data, r stands for the bit length of the redundant check-sum which is related to data and n = k + r. However, it is analyzed that there is a principle weakness of cyphers that it does not reserve arithmetic. Moreover, it cannot remain supportive to protect arithmetic structures contrary to fault attacks. For eliminating this issue, Gaubatz et al. [41] have introduced a new method of non-linear arithmetic codes which is basically known as 'robust quadratic codes'.

According to the general structure of the arithmetic single residue codes following points analyzed:

Let $C = \left\{ (x, w), x \in Z_{2^k}, w = f(x) \in F_p \right\}$ According to arithmetic single residue code, $f : Z_{2^k} \mapsto F_p$ This function supports to calculate the check-sum w. w is calculated for the prime check modulus is ‘p’ has size $r = \left\| \lceil \log_2 p \rceil \right\|$ bits.

An attacker majorly focuses on adding a fault in the circuit while creating the changes in the original information values. It allows fulfilling the specific end goal of representing it hypothetically. The mistakes caused are expected towards influencing the original codewords through the added substance. Every information value which is provided in the circuit supports to symbolize the pair (x, w). In this context, x stands for original data and w stands for check-sum related to it. Simultaneously, there is a need for keeping in mind regarding the end goal which remains assistive to become successful. The assailant injects a couple of error vectors on x as well as w. Note: Non-zero error ‘e' is disguised as a code word (x, w) in the context of the incorrect message is as yet a substantial code word in C. As it were, the error $e \in \left\{ (e_x, e_w), e_x \in Z_{2^k}, e_w \in Z_{2^r} \right\}$ is masked for the message (x, w) when $\left(x + e_x, w + e_w \right) \in C$ that is, iff

$$f((x + e_x) \bmod 2^k) = f(x) + e_w \bmod 2^r$$

Gaubatz et al. [41], essentially select $f(x) = x^2 \bmod p$ for the check-sum function. Using this function, allows all non-zero error patterns to combine which can be defined as probability lower bounded by $1 - \max(4, 2^k - p + 1)2^{-k}$. For example, for k = r = 32 the k-bit prime closest to 2^k is $2^{32} - 5$, thus bounding $Q(e) = \max(4, 2^k - p + 1)2^{-k}$ by $3 \cdot 2^{-31}$, so $1 - \max(4, 2^k - p + 1)2^{-k}$ is greater than 99%.

The probability of missing the error allows supporting the data-dependent in the coding scheme. So, there is a dynamic enemy which endeavours to incite undetected error in the information which allows estimating the information before the fault infusion. This process allows keeping the end goal in mind regarding registering the imperceptible mistake design. Moreover, there is a need for adequate dimension and transient precision for the purpose of infusing the error pairs. Subsequently, an assailant does not have a way of inserting the error vector. It allows reading the objective information initially in the context of the vector. It allows processing a proper error pattern which allows accurately embedding the registered example. It supports to offer with high spatial while offering a temporal resolution.

Imaginable utilizations of systematic non-linear error which allows detecting the codes while ensuring the finite-state machines and supports advancing the encryption standard (AES) [43].

In [38], the authors introduced applying robust nonlinear residue codes to protect elliptic curve point addition and doubling operations against active fault attacks. These codes provided nearly perfect error-detection capability at reasonable overhead.

3 Fault Detection Method for ECDSA

3.1 Background

Introduced by Koblitz [44] and Miller [45], as an alternative for cryptographic protocols based on the discrete logarithm problem on a multiplicative group of a finite field, Elliptic curves cryptography provides a protocol that relies on the relevance of the following question: Given 2 points G and Q on an elliptic curve E such that $Q = dG$ - find d. The arithmetic of elliptic curves in general, its properties and the way it is used in Cryptography, are common knowledge. See examples [46,47].

3.1.1 ECDSA

The ECDSA suite works in accordance with the relevance of the ECDLP problem producing a signature token from a secret value k, which represents an ephemeral key and this key's lifespan lasts only during the signature procedure. Protocol for digital signatures is derived from the classic DSA [1], by substituting the discrete logarithm problem over a number field with the one constructed over the (E,+) curve points group[48,49], including the signature algorithm of ECDSA, as described in Algorithm 1 [50].

Algorithm 1 [50]: ECDSA Signature Generation

Input: curve parameters (E,G), private key d, message m

Output: signature S

```
1: $e =$ hash(m)
2: $k =$ random € $[1, n - 1]$
3: $P = [k]G$
4: $r = xP \bmod n$
5: $s = (e + rd)/k \bmod n$
6: return $S = (r, s)$
```

In particular, the signature generation algorithm (Algorithm 1) produces the signature token S, taking as input the definition of the group (E,+) together with a default generator G € E, the private key parameter k € Zp and the message that requires authenticity is m. In order to build the signature is created using three basic steps. First, the algorithm obtains a hashed version e of the message m (Line 1) and a non zero random number (Line 2), smaller than the order of the curve k. Next, the point scalar multiplication between the random number and the generator G activates (Line 4) with the x coordinate of the resulting point which then divided by the order of the curve n and stored in r.

Occasionally the result is $r = 0$, so then the procedure must be repeated with a different random number until a non-zero r is obtained. The final step is computing the signature by combining the hash of the message, the value obtained through the point scalar multiplication, and the extracted random k (Line 5). The signature token S is represented by the pair (r, s). It is particularly important to choose A cryptographically strong random number for k must be selected and can never be reused. It is, in fact, useless to extract the value of the secret key d if two different signatures are computed with the same random k. For guaranteed validity of an ECDSA signature public key $Y = [d]G$ is provided, and the secret value d is protected by the computational hardness of the ECDLP, so the verifier can compute the message hash e and compares the received r value with $[e/s]G+[r/s]Y$. The validity of the signature is proved when the two quantities match.

3.1.2 Fault attacks on ECDSA

Durability is the determining factor for the mathematical security of ECDSA signature generation algorithm based on the Elliptic Curve Discrete Logarithm Problem (ECDLP).

A signing and a signature verification algorithm make up the basis of the ECDSA cryptosystem, each with a different function. The signing algorithm can create an authenticated token or digital signature, by using a private value, given by and known only to the signer. In contrast, the signature verification algorithm works with publicly known values to check the authenticity of the signature. An attacker who is able to retrieve the secret key can then forge valid signatures at will, resulting in voiding any authenticity warranty provided by the scheme [49]. Therefore, the infiltration of the ECDSA cryptosystem security parameters depends on the difficulty of uncovering the secret value using only publicly available parameters and the digital signature. If the devices performing the signature were captured by a raider, it is crucial that the secret needed to build authentic signatures cannot be extracted from the device. A common scenario for an attack goes as follows an attacker takes a signing token which contains the secret key, such as a smart card, duplicates it, then returns the original one to the legitimate owner, before said owner even realized it was missing. The literature on this subject describes daring secret key retrieval attacks.

The authors of [51] rely on both a particular arithmetical representation of the values involved in the algorithm and the possibility of inducing single bit faults in a specific bit of a single value with strict time accuracy. In [52] an attack is presented relying on a fault on the modulus used for the computations but needs a few thousands of faulty results to be successful against a system employing a small-sized curve. Eventually, in [53] an instruction skip fault is used to recover some bits of the nonce used during the signature. A few tenths of faulty results are again sufficient for a small-sized curve.

Example [50] is a fault based attack able to retrieve the full secret key, for all the standardized lengths, by injecting a single bit fault anywhere in the word. The attack could succeed against all the real-world word lengths of the target architecture.

3.2 Fault detection method

The application of non-linear cyphers to protect processes directed to ECDSA, that is, calculating signature processes (line5 in algorithm 1) contrary to fault infusion attacks, will be explained showing it is applicable for wide usage. ECDSA structures in shade of prime fields Fp are the process focused on here. This will be achieved by calculating signature processes utilizing the systematic non-linear (n, k, r)-code which utilizes redundancy for error detection. The accompanying error check function is characterized on a point coordinate $x \in F_p$ to attain a nonlinear error check-sum

$$f(x) = x^2 (\bmod p) \in F_p$$

Therefore, the point X is encrypted as (X, f (X)). For practicality and as an error check for redundancy, r check digits have been added to the point value.

In the following subsection, the protected uses of calculating signature processes are provided. These executions use the error detection procedure as depicted in Section 2. Non-linear error detection works by creating two calculation paths that are non-linear in alignment. The initial step in accomplishing this includes encrypting the signature in an operation by utilizing the nonlinear code depicted in Definition 1. The first non-linear path is the original non redundant datapath. The second path, which is known as the ‘predictor' piece, goes in parallel to the non-redundant path and basically guesses the checksum of the consequences of the first calculation. The predictor piece utilizes the fault-free check-sums of the inputs to calculate the check-sum of the outputs that will be calculated by the original block, so this is not a mere repetition of the first equipment to actualize the predictor. For each data path, the total process count is stated in terms of multiplications, divisions and addition/subtractions, where M refers to multiplication, D refers to division and A refers to addition or subtraction.

In the following, the predictor block functions are shown. The functions calculate the predictable S3c by means of the inputs and their check-sums. The subscript c is utilized to determine the check-sum of the data portion for a codeword as defined in Definition 1. For instance, (S, S_c) establishes a codeword where

$S_c = S^2 (\mathrm{mod}\, p)$, S is the data part, and S_c is the check-sum.

The predictable check sums for signature computation (step5 in algorithm 1) should be

$$S_c = (S)^2 = \left(\frac{(e + rd)}{k}\right)^2 \mathrm{mod}\, p$$

Where e is hash(m), r is xP mod n, and d is the private key.

Then, we want to express the relations on the right-hand side as a function of the inputs and their check-sums for point addition.

$$S_c = (S)^2 = \left(\frac{(e + rd)}{k}\right)^2 = \left(\frac{1}{k}\right)^2 \left(e^2 + r^2 d^2 + 2erd\right) \mathrm{mod}\, p$$

After some algebra, we acquire the accompanying equation array for each operation. These equations basically introduce the function actualized by the predictor unit in our implementation.

$$A = \frac{1}{k_c}$$

$$B = rd$$

$$C = eB$$

$$S_c = A\left(e_c + r_c d_c + 2C\right) \mathrm{mod}\, p$$

Where e_c is e^2mod p, r_c is r^2mod p, and d_c is d^2mod p.

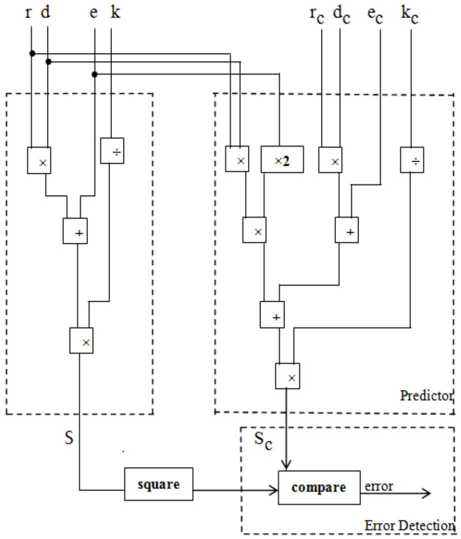

Fig. 1. ECDSA and Predictor unit

The whole operation sum for this predictor unit will be 5M + 2A + 1D, where M is multiplication and A is addition and D is division. Note that each of the operations in this setup is modulo p, where p is the prime that produces the finite field the elliptic curve is characterized over. In Fig. 1, the implementations of these operations is shown.

Example [54]: Let the elliptic curve $E: y^2 = x^3 - 3x + 69424$ with p = 114973 and a base point G = (11570, 42257) with order n=114467; select d = 86109 then Q = dG = (6345, 28549); and the message m = "worldof" the hash value is e = H(m) = 1789679805, the signature for the message m is (r,s) as following:

1. Select k = 84430 such that $1 \le k \le n-1$.
2. Compute kG = (11705,10585), r =31167(mod 114973).
3. Compute $s = k^{-1}(e + dr) = 82722(\mathrm{mod}\,114973)$.
4. Compute $e_c = e^2 \bmod p = 111437$
5. Compute $r_c = r^2 \bmod p = 89985$
6. Compute $d_c = d^2 \bmod p = 36138$
7. Compute $S_c = A(e_c + r_c d_c + 2C) \bmod p = 81243$
8. Compute $s^2 = 82722^2(\mathrm{mod}\,114973) = 81243$
9. Compare Sc and s2

4 Fault Detection Method for GQ

4.1 Background

4.1.1 Guillou-Quisquater identification scheme

Find below a brief description of the Guillou-Quisquater Identification scheme [55,17]:

Key Generation

Assume that Peggy wants to prove her identity to Victor. Peggy's public key consists of J (a set of credentials), n (a product of two large secret primes p, q) and an exponent v (v should be sufficiently large and $\gcd(v,(p-1)(q-1)) = 1$. Peggy's private key is B, calculated such that $JB^v \equiv 1(\mathrm{mod}\,n)$.

GQ Identification Scheme

i Peggy picks a random integer r, $r \in [1, n-1]$ r. Peggy computes $T \equiv r^v(\mathrm{mod}\,n)$ and sends it to Victor.

ii Victor picks a random integer $d, d \in [0, v-1]$. Victor sends d to Peggy.

iii Peggy computes $D \equiv rB^d(\mathrm{mod}\,n)$ and sends it to Victor.

iv Victor computes $T' \equiv D^v J^d(\mathrm{mod}\,n)$. If $T \equiv T'(\mathrm{mod}\,n)$, then authentication succeeds.

4.1.2 An attack on GQ scheme

A cryptographic device that holds the private key B proves its identity to another cryptographic device using the GQ protocol [17]. The attacker infiltrates the communication during the protocol and also corrupts the modulus using a glitch to a random number ni which isn't known to him. The attack on the identification scheme of Guillou-Quisquater can be performed in the following manner:

1. At the first stage, the attacker collects triples of (Ti, Di, di), which were sent between Peggy and Victor during an authentication protocol, with simultaneous attempts to produce a fault injection in the modulus n, thus producing faulty moduli ni.
2. Let L be a small prime and assume $\gcd(L-1, v) = 1$.
3. For each triple, if $\gcd(d_i, L-1) = 1$ an attacker calculates $B_i \pmod{L} \equiv D_i^{\frac{1}{d_i}} \cdot T_i^{-\frac{1}{d_i v}} \pmod{L}$
4. Most of the results $B_i \pmod{L}$ will be randomly distributed, while a small number of them will give the same result. The repetition of the result corresponds to a case when L divides n_i, and thus the attacker knows the true value of $B \pmod{L}$.
5. An attacker uses the Chinese Remainder Theorem (CRT) to reconstruct the unreduced value of B from partial residues $B \pmod{L}$.

4.2 Fault detection method for GQ

We introduce how a similar methodology can be practical to GQ Scheme. In the next we give Utilizing a similar strategy talked about in Section 3.2, we figure the accompanying indicator purposes for the authentication process to protect the private key B. The predictable check- sums for authentication computation should be

$$D_c = (D)^2 = (rB^d)^2 \bmod L$$

$$D_c = r^2 B^{2d} \bmod L$$

$$D_c = r_c B_c^d \bmod L$$

Note that every one of the operations in this setup is modulo L.

5 Conclusion

In this paper, we presented applying nonlinear fault discovery codes to defend protect ECDSA operations against fault attacks. We also presented applying the same idea to protect Guillou-Quisquater authentication scheme (GQ) against fault injection attacks. These codes give almost perfect error discovery capacity (aside from an exponentially small probability) at sensible overhead. We presented a fault detection scheme by using an error detecting code. This fault detection scheme has shown to have over 99% fault detection coverage.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] National Institute of Standards and Technology (NIST): Digital signature standard (DSS). Federal Information Processing Standard FIPS PUB 186-2; 2000.

[2] NIST "Recommended elliptic curves for federal government use" Tech. Rep. National Institute of Standards and Technology U.S. Department of Commerce; 1999.

[3] Ghanmy Nabil, Khlif Naziha, Fourati Lamia, Kamoun Lotfi. Hardware implementation of Elliptic Curve Digital Signature Algorithm (ECDSA) on Koblitz Curves. 8th IEEE, IET International Symposium on Communication Systems, Networks and Digital Signal Processing; 2012.

[4] Khatwani C, Roy S. Security analysis of ECC based authentication protocols. The Proceedings of the 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India. 12–14 December 2015;1167–1172.

[5] Khatwani C, Roy S. Cryptanalysis and improvement of ECC based authentication and key exchanging protocols. Cryptography; 2017.

[6] Mangard S, Oswald E, Popp T. Power analysis attacks: Revealing the secrets of smart cards. Advances in Information Security, Springer-Verlag; 2007.

[7] Boneh D, DeMillo RA, Lipton RJ. On the importance of checking cryptographic protocols for faults. Journal of Cryptology, Springer-Verlag. 1997;14(2):101-119.

[8] Biham E, Shamir A. Differential fault analysis of secret key cryptosystems. Proceedings of Advances in Cryptology, Springer-Verlag. 1997;513-525.

[9] Hoch J, Shamir A. Fault analysis of stream ciphers. Cryptographic Hardware and Embedded Systems CHES' 2004, Springer-Verlag, LNCS. 2004;3156:240-253.

[10] Breier E, Chevallier-Mames B, Ciet M, Clavier C. Why one should also secure public key elements. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2006) LNCS. 2006;4249: 324-338.

[11] Seifert JP. On authenticated computing and RSA-based authentication. ACM Conference on Computer and Communications Security. 2005;122-127.

[12] Naccache D, Nguyen PQ, Tunstall M, Whelan C. Experimenting with faults, lattices and the DSA. PKC' 05, LNCS, Springer Verlag. 2005;3386:16-28.

[13] Michael Kara-Ivanov, Eran Iceland, Aviad Kipnis. Attacks on authentication and signature schemes involving corruption of public key (Modulus). 2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography, IEEE Computer Society.

[14] Mehran Mozaffari-Kermani, Reza Azarderakhsh, Anita Aghaie. Fault detection architectures for post-quantum cryptographic stateless hash-based secure signatures benchmarked on ASIC. ACM Transactions on Embedded Computing Systems (TECS). 2017;16(2). Article No. 59.

[15] Daisuke Fujimoto, Yu-Ichi Hayashi, Arthur Beckers, Josep Balasch, Benedikt Gierlichs, Ingrid Verbauwhede. Detection of IEMI fault injection using voltage monitor constructed with fully digital circuit. 2018 IEEE International Symposium on Electromagnetic Compatibility and 2018 IEEE Asia-Pacific Symposium on Electromagnetic Compatibility (EMC/APEMC); 2018.

[16] Chorage SS, Somwanshi VA. Fault resistant encryption system using high speed AES algorithm on FPGA. 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA); 2017.

[17] Hassen Mestiri, Fatma Kahri, Belgacem Bouallegue, Mehrez Marzougui, Mohsen Machhout. Efficient countermeasure for reliable KECCAK architecture against fault attacks. 2017 2nd International Conference on Anti-Cyber Crimes (ICACC); 2017.

[18] Bertoni G, Breveglieri L, Koren I, Piuri V. Fault detection in the advanced encryption standard. Proc. Conf. Massively Parallel Computing Systems (MPCS' 02). 2002;92-97.

[19] Akkar M, Giraud C. Implementation of DES and AES, secure against some attacks. Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES' 01). 2001;315-325.

[20] Bertoni G, Breveglieri L, Koren I, Maistri P, Piuri V. Error analysis and detection procedures for a hardware implementation of the advanced encryption standard. Proc. IEEE Transactions on Computers. 2003;52(4).

[21] Breveglieri L, Koren I, Maistri P. Incorporating error detection and online reconfiguration into a regular architecture for the advanced encryption standard. In Proceedings of 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT). 2005;72-80.

[22] Wu K, Ramesh K, Kuznetsov G, Goessel M. Low cost concurrent error detection for the advanced encryption standard. In Proceedings of International Test Conference 2004 (ITC). 2004;1242-1248.

[23] Yen CH, Wu BF. Simple error detection methods for hardware implementation of advanced encryption standard. IEEE Transactions on Computers. 2006;55(6):720-731.

[24] Fernandez-Gomez S, Rodriguez-Andina JJ, Mandado E. Concurrent error detection in block ciphers. in Proceedings of International Test Conference 2000 (ITC). 2000;979-984.

[25] Shamir A. Improved method and apparatus for protecting public key schemes from timing and fault attacks. US Patent; 1999.

[26] Aumuller C, Bier P, Fischer W, Hofreiter P, Seifert JP. Fault attacks on RSA with CRT: Concrete results and practical counter-measures. Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES' 02), B. Kaliski Jr., C. Koc, and C. Paar, Eds. 2002;260-275.

[27] Vigilant D. RSA with CRT: A new cost-effective solution to thwart fault attacks. Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES). 2008;130-145.

[28] Giraud C. An RSA implementation resistant to fault attacks and to simple power analysis. IEEE Trans. Computers. 2006;55(9):1116-1120.

[29] Yen SM, Joye M. Checking before output may not be enough against fault-based cryptanalysis. IEEE Trans. Computers. 2000;49(9):967-970.

[30] Yen SM, Kim S, Lim S, Moon SJ. RSA speedup with chinese remainder theorem immune against hardware fault cryptanalysis. IEEE Trans. Computers. 2003;52(4):461-472.

[31] BlO¨ Mer J, Otto M, Seifert JP. A new CRT-RSA algorithm secure against Bellcore attacks. Proc. ACM Conf. Computer and Comm. Security. 2003;311-320.

[32] Boneh D. Twenty years of attacks on the RSA cryptosystems. Notices of the Am. Math. Soc. 1999; 46(2):203-213.

[33] Kun Ma, Han Liang, Kaijie Wu. Homomorphic property-based concurrent error detection of RSA: A countermeasure to fault attack. IEEE Transactions on Computers. 2012;61(7).

[34] SECG, Standards for efficient cryptography 1 (SEC1): Elliptic Curve Cryptography; 2009.

[35] Dominguez-Oviedo A, Hasan MA. Algorithm-level error detection for ECSM. Centre Appl. Crypto. Res., Univ. Waterloo, on, Canada, Tech. Rep., TR-2009-05; 2009.

[36] Fouque PA, Lercier R, R´eal D, Valette F. Fault attack on elliptic curve with Montgomery ladder implementation. In Proc. FDTC. 2008;92–98.

[37] Domınguez-Oviedo A, Hasan MA. Error detection and fault tolerance in ECSM using input randomization. IEEE Trans. Depend. Secure Comput. 2009;6(3):175–187.

[38] Akdemir KD, Karakoyunlu D, Sunar B. Non-linear error detection for elliptic curve cryptosystems. IET Inf. Secur. 2012;6(1):28–40.

[39] Sudhakar T, Natarajan V, Kannathal A. Efficient and secure implementation of elliptic curve scalar multiplication against power analysis attacks. ICIA '16, Pondicherry, India; August 25-26, 2016.

[40] Kun Ma, Kaijie Wu. Error detection and recovery for ECC: A new approach against side-channel attacks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2014;33(4).

[41] Gaubatz G, Sunar B, Karpovsky M. Non-linear residue codes for robust public-key arithmetic. Proc. Third Workshop on Fault Tolerance and Diagnosis in Cryptography Yokohama, Japan. 2006;4236: 173–184.

[42] Karpovsky M, Taubin A. A new class of nonlinear systematic error detecting codes. IEEE Trans. Inf. Theory. 2004;50(8):1818-1820.

[43] Kulikowski K, Wang Z, Karpovsky M. Comparative analysis of robust fault attack resistant architectures for public and private cryptosystems. Proc. Fifth Workshop on Fault Diagnosis and Tolerance in Cryptography, Washington, DC, USA. 2008;41–50.

[44] Koblitz N. Elliptic curve cryptosystems. Mathematics of Computation. 1987;48(177):203-209.

[45] Miller VS. Use of elliptic curves in cryptography, Crypto 85', of Lectures Notes in Computer Science; 1986.

[46] Joseph H. Silverman, the arithmetic of elliptic curves. Springer-Verlag, New York; 1986.

[47] Neal Koblitz. A course in number theory and cryptography, 2nd Edition, Springer-Verlag, New York. 1994;167 - 199.

[48] Deepti Jyotiyana, Varun P. Saxena. Lecture notes in networks and systems. 2017;12:283.
ISSN: 2367-3370.
ISBN: 978-981-10-3934-8.

[49] Alessandro Barenghi, Guido M. Bertoni, Luca Breveglieri, Gerardo Pelosi, Stefano Sanfilippo, Ruggero Susella. A fault-based secret key retrieval method for ECDSA. ACM Journal on Emerging Technologies in Computing Systems. 2016;13(1).
ISSN: 15504832.

[50] Barenghi A, Bertoni G, Palomba A, Susella R. A novel fault attack against ECDSA. Proc. IEEE Int. Symp. Hardware-Oriented Security Trust. 2011;161-166.

[51] Bl ¨omer J, Otto M, Seifert JP. Sign change fault attacks on elliptic curve cryptosystems. In Workshop on Fault Diagnosis and Tolerance in Cryptography. Prentice Hall. 2006;36–52.

[52] Kara-Ivanov M, Iceland E, Kipnis A. Attacks on authentication and signature schemes involving corruption of public key (modulus). In Workshop on Fault Diagnosis and Tolerance in Cryptography. IEEE Computer Society. 2008;108–115.

[53] Schmidt JM, Medwed M. A fault attack on ecdsa, in 6th Workshop on Fault Diagnosis and Tolerance in Cryptography – FDTC 2009, Proceedings, D. Naccache and E. Oswald, Eds. Verlag IEEE-CS Press. 2009;93–99.

[54] Hung-Zih Liao, Yuan-Yuan Shen. On the elliptic curve digital signature algorithm. Tunghai Science. 2006;8:109−126.

[55] Guillou LC, Quisquater JJ. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. Proceedings of Eurocrypt 88, Springer-Verlag Eds. 1988;123-128.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://www.sciencedomain.org/review-history/26416