7th International Conference on
Combinatorics, Cryptography, Computer Science and Computing
November. 16-17, 2022

Combining ECDSA and Image Secret Sharing, implementing it on the FPGA board, and evaluating its security against DPA

Massoud Hadian Dehkordi[1*1], Mona Alizadeh[۲]
Iran University of Science and Technology
Narmak, Tehran, Iran
mhadian@iust.ac.ir; alizadehmona2@gmail.com

Javad Fard Bagheri
Payame Noor Mazandaran University
Sari, Mazandaran, Iran
jvdfard@gmail.com

ABSTRACT

Blockchain has become so important in today's world that it can be used wherever a database or data sharing system was needed, eliminating the need for trust. One of the Blockchain vulnerabilities is caused by weak randomness in ECDSA. A random number is not secure, cryptographically, which leads to leakage in private keys and even the user's fund theft. So by combining ECDSA with image secret sharing, we presented an algorithm that has acceptable security against DPA.

KEYWORDS: Blockchain, Cryptography, ECDSA, ISS, DPA, FPGA

1 INTRODUCTION

The concept of blockchain was first coined by Satoshi Nakamoto with the advent of bitcoin in 2008, and the king of digital currencies used it to store information about users' as sets [1]. Since the launch of Bitcoin, it has attracted the attention of many organizations, governments, and businesses, and recently the governments of Germany and Japan have adopted it as a method of international payment. Blockchain relies on many classical cryptographic technologies and distributed systems [2] such as ESDSA, Ring, con- sensus mechanisms such as Proof-of-work, Z- K-SNARK, and the Merkle tree. Therefore, if one of its main components is exposed to security vulnerabilities, the security of the blockchain will be affected. The concept of digital signatures was coined by Diffie and Hellman in 1976 when they opened the gate to public-key cryptography. The digital signature is used as the basic cryptographic algorithm to ensure source authentication, non-denial, source rejection, and integrity. Fake a new message, even if it can access Oracle Signature, which can provide signature services [3].

RSA's digital signature is one of the most well-known cryptographic systems [4]. As we know, the RSA secure key size is 4096 bits [5], which means that processing the large key has slower algorithms for signature generating. The ECDSA signature solves this problem. The ECDSA algorithm is essentially based on the stiffness of the elliptic curve version of the discrete logarithm problem, and many cryptographic systems use it instead of RSA, such as bitcoin [1]. Liao and Shen 2006 [6] show that there is a weakness in ECDSA in that by repeating a random variable for ECDSA the enemy can easily extract the

[1] Corresponding author: M. Hadian Dehkordi
Email: mhadian@iust.ac.ir

private key. To address this, Liao and Shen proposed an advanced ECDSA [6]. Computational costs are reduced with improved designs, but safety is not improved.

Each algorithm leaks information when implemented in hardware due to the presence of side channels, which can be used to understand the data used in the algorithm. Since digital signatures are easily defeated by side-channel attacks, we decided to come up with a design by combining the digital signature and image secret sharing, then implement and run our proposal on the FPGA hardware hardboard. And by creating sensors, we measured the power consumption of the FPGA during the execution of the algorithm. We then performed the DPA attack using the measured capabilities and evaluated the security of our design against this attack [7].

This article is organized as follows: In the second part of the ECDSA Introduction, we briefly describe the ISS. The third part is our improved ECDSA. In Section 4, we analyze the security of our proposal against DPA. The last part contains the conclusion.

2 PRELIMINARIES

In this section, we briefly explain the elliptic curve (EC) with the elliptic curve discrete logarithm problem ECDLP [8], ECDSA [9], and ISS [11].

2.1 Elliptic curve

Consider $GF(p)$, be a prime field and $a, b \in GF(p)$ are constants. An elliptic curve $E_p(a, b)$, over $GF(p)$ is defined as the set of points $(x, y) \in GF(p) \times GF(p)$ where the following equation is satisfied:
$y^2 = x^3 + ax + b$ (2.1)

2.1.1 Elliptic Curve Digital Signature Algorithm

The ECDSA comprises three steps. The first step is parameter generation, the second is signature generation, and signature verification. First of all, the transmitter A requirement to generate private (32 bytes) and public key are as follows [10]:

- Choose an elliptic curve E defined over F_q. The number of points should be partible by a large prime n.
- Choose a point $P \in E(F_q)$ of order n.
- Choose a statistically unpredictable and unique integer d in the interval $[1, n - 1]$.
- Calculate $Q = dP$.
- The public key of A is (E, P, n, Q), A's secret key is d.

Then the transmitter A generates the signature. To sign a message m, the transmitter should do as follows [9]:

1. Select a statistically unpredictable and unique integer k in the interval $[1, n - 1]$.
2. Calculate $h(m)$. where h is a secured hash function such as $SHA - 1$ or $SHA - 2$.
3. Calculate $KP = (x, y)$ and $r = x \bmod n$. If $r = 0$ then go to step 1.
4. Calculate $s = K^{-1}(h(m) + dr) \bmod n$.
5. If $s = 0$ then go to step 1.
6. The signature for the message m is the integers pair (r, s).

The recipient B should do the following to verify A's signature (r, s)[9]:

1. Gain an authentic of A's public key (E, Q). Verify that r and s are integers in the interval $[1, n - 1]$.
2. Calculate $w = s^{-1} \bmod n$ and $h(m)$.
3. Calculate $u_1 = h(m)w \bmod n$ and $u_2 = rw \bmod n$.

4. Calculate $u_1 + u_2 = (x_0, y_0)\ mod\ n$ and $v = x_0\ mod\ n$.
5. The signature is accepted if and only if $v = r$.

2.2 Image Secret Sharing

Image secret sharing (ISS) for the case (k, n) encodes a secret image into n noise-like shares and then dispenses the shares to several participants. The secret image can be ameliorated by mustering any k or more permitted shares while less than k shares overall decode nothing of the secret. ISS can be exerted to not only data concealing but also authentication, watermarking, access control, transmitting passwords, dispensed storage and computing, etc[11].

To avoid duplication, the ISS algorithm will be presented in our scheme section.

3 THE ECDSA IMPROVEMENT

In this section, as we know, multiplication operations in mathematics have a lot of power consumption, and in the ECDSA signature, the multiplication operator has a large amount of power consumption at the point of the curve and is, therefore, a good point to attack the algorithm. To prevent an attack, we provide a digital signature algorithm using image secret sharing, which changes the overall power of the algorithm and draws the attacker's attention to save the image.

In this algorithm, we convert the signature to an image and then encrypt the image pixels.

Signature generation algorithm

KeyGen: $(E, q, a, b, G, l, h, S, p; d, Q)$

- A elliptic curve E: $y^2 = x^3 + ax + b\ over\ F_q$.
- A prime $2^{255} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$.
- choose $a = 0$, $b = 7$.
- A random base point in E with prime order l
- A hash, instantialed with $SHA - 1$.
- A $M \times N$ binary secret image S, the parameteres (k, n) and permutation key and n shares $SC_1, SC_2, \ldots, SC_n$.

1. $Q = dG$ from in $[1, l - 1]$ and put Q in image S.
2. Select a cryptographically secure random A from $[1, l - 1]$.
3. Calculate $h(m)$.
4. Calculate the curve point $(x_1, y_1) = AG$.
5. Calculate $r = x_1\ mod\ l$. if $r = 0$, go back to step 2.
6. Calculate $s = A^{-1}(h(m) + rd) mod\ l.$ if $s = 0$ go back to step 2.
7. The signature is the pair (r, s). (And $(r, -s\ mod\ l)$ is also valid signature) and put (r, s) in image S.
8. Output the image S.

ISS algorithm

1. A prime number $P = 251$ is selected. For each position $(i, j) \in \{(i, j) \mid 1 \leq i \leq M, 1 \leq j \leq N\}$, if $R = S(i, j) > P - 1$ we set $R = P - 1$.
2. Encrypt the secret image to gain S_1 using permutation method with the input key.
3. Every k not-shared-yet pixels (a block) of S_1 are sequentially picked up to gain a block, denoted as $B_0, B_1, \ldots, B_{k-1}$.

4. Generate a $k - 1$ degree polynomial $f(x) = (a_0 + a_1 x + \ldots + a_{k-1} x^{k-1}) \; mod \; P$ in which $a_i = B_i, i = 0, 1, 2, \ldots, k - 1$.
5. Compute $sc_1 = f(1), \ldots, sc_i = f(i), \ldots, sc_n = f(n)$
6. Arrange $sc_1, sc_2, \ldots, sc_n$ to $SC_1(i, j), SC_2(i, j), \ldots, SC_n(i, j)$.
7. Repeat Steps 3–6 until all the pixels of $S_1$are processed.
8. Output the n shares $SC_1, SC_2, \ldots, SC_n$.

Signature verification algorithm

1. Calculate $a_i = (\sum_{j=i}^{k-1} f(x_j) \prod_{m=i,m\neq j}^{k-1} \frac{x_m}{x_m - x_j}) \; mod \; P$ and Capture (r, s) and Q from image S.
2. Verify that r and s are integers in $[1, l - 1]$. If not, the signature is invalid.
3. Calculate $h(m)$.
4. Calculate $u_1 = h(m)s^{-1} \; mod \; l$ and $u_2 = rs^{-1} \; mod \; l$.
5. Calculate the curve point $(x_1, y_1) = u_1 G + u_2 Q$. If $(x_1, y_1) = 0$ then the signature is invalid.
6. The signature is valid if $r \equiv x_1 \; mod \; l$, invalid otherwise.

4 SECURITY ANALYSIS

According to the figure below, which shows that the power consumption in the FPGA range follows the normal distribution, it can be concluded that a DPA attack can be applied to this algorithm. In the following, we evaluate the security level of the proposed algorithm against this attack.

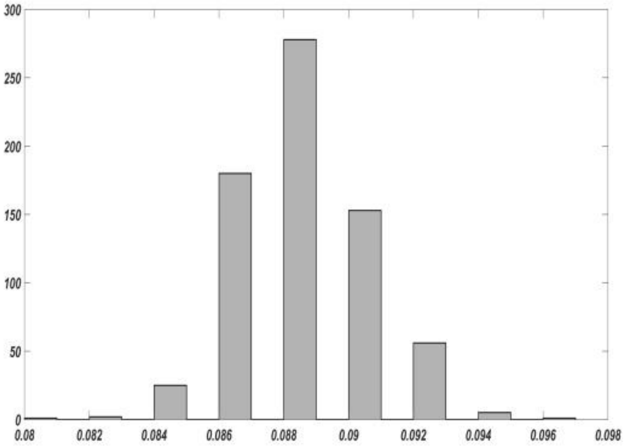

Fig1. normal distribution

To perform DPA, we have to guess the bytes in the image matrix and classify the Power Traces based on our guess and apply the DPA attack. If we see a peak in the waveform after the DPA attack, it means that our attack is successful and otherwise unsuccessful. In this attack, the number of Power Traces is the same as the number of executions of the algorithm, and the DPA attack is completely dependent on this number, and the higher the number of Power Trace needed to attack, the harder the attack and the higher the security level of the algorithm. It takes about a few seconds to run each algorithm, which is divided into some samples when measuring power consumption. For example, the proposed algorithm is executed in a time that this period has 10×10^4 samples. Now in the following figures, the time of saving some image bytes in Power Trace. We show:

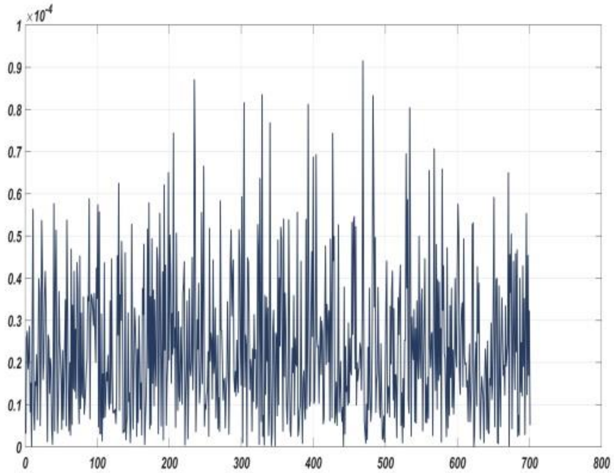

Fig2. hypothesis of byte1-25000

In this figure, the proposed algorithm is executed 25,000 times and 25,000 Power Traces are taken, but with the correct guess of the first byte of the image, no peak occurred in the diagram.

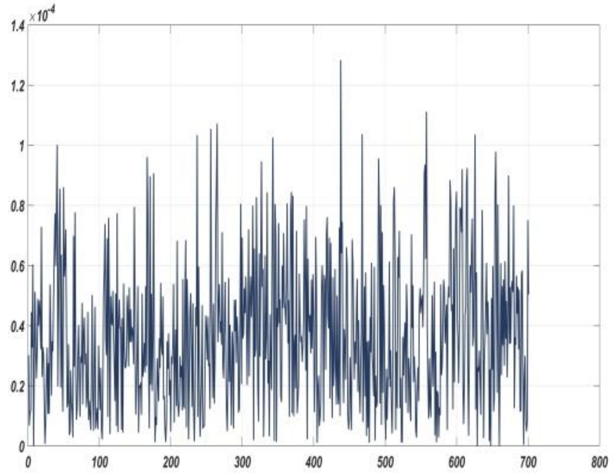

Fig3.hypothesis of byte2-25000

In this figure, the proposed algorithm is executed 25,000 times and 25,000 Power Traces are taken, which with the correct guess of the second byte of the image, no peak occurred in the diagram. Also, for the third byte, there was no peak with 25,000 Power Trace.

In the following figures, with 30,000 Power Trace for the first, second, and third bytes, some peaks occur but are indistinguishable. But in the following figures with 35000 Power Trace, a well-separated peak can be created.

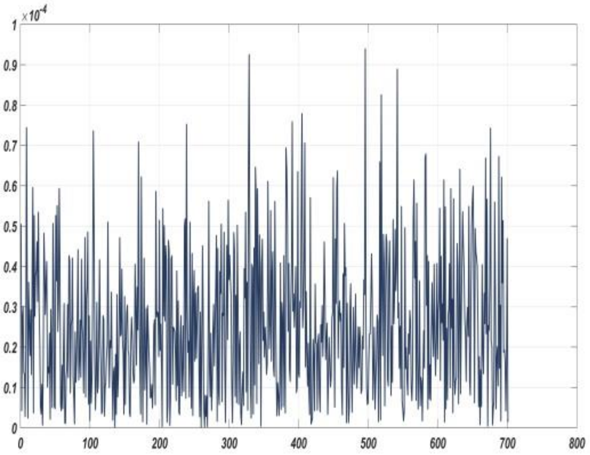

Fig4. hypothesis of byte3-25000

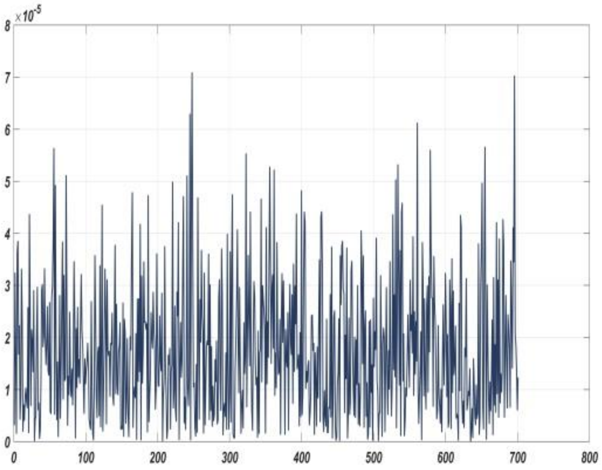

Fig5. hypothesis of byte1-30000

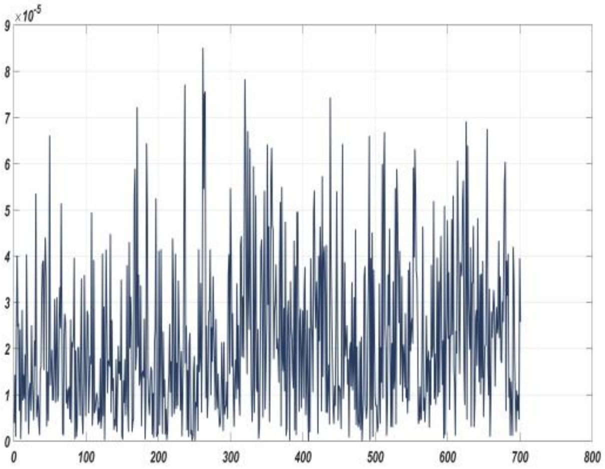

Fig6. hypothesis of byte2-30000

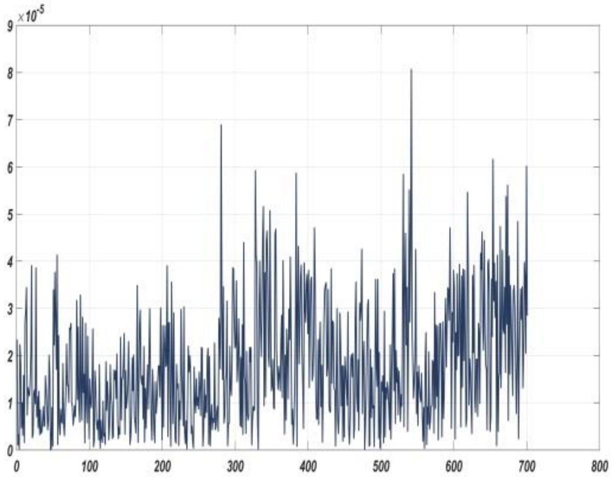

Fig7. hypothesis of byte3-30000

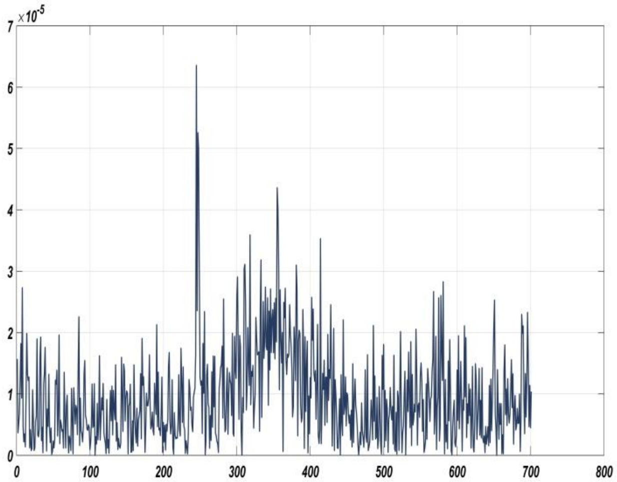

Fig٨. hypothesis of byte١-3٥000

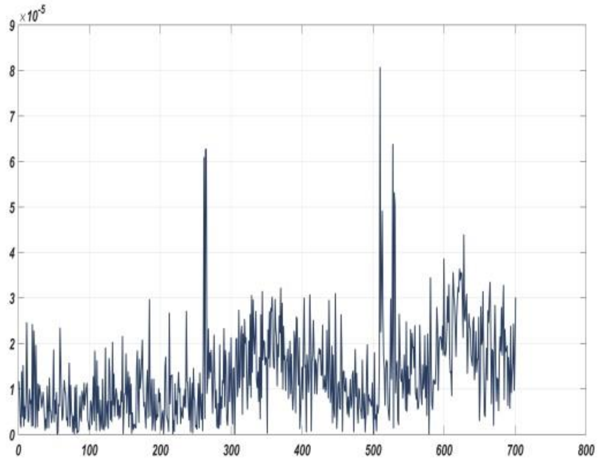

Fig٩. hypothesis of byte٢-3٥000

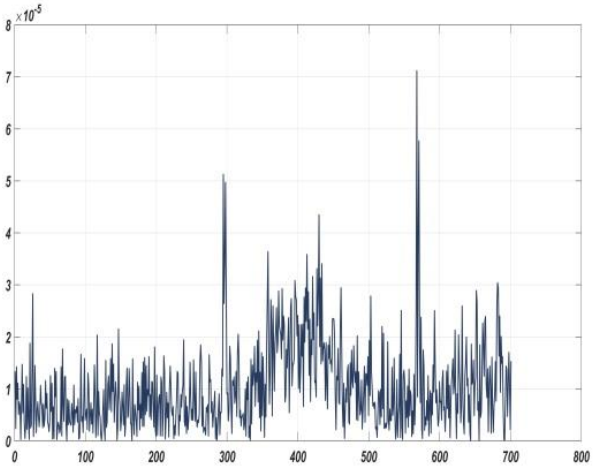

Fig١٠. hypothesis of byte٣-3٥000

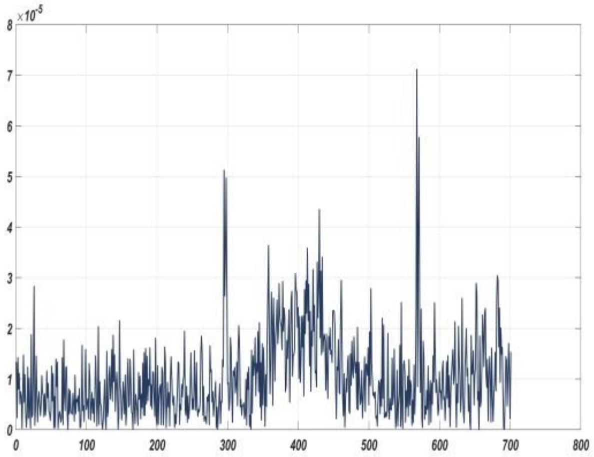

Fig١١. hypothesis of byte٤-3٥000

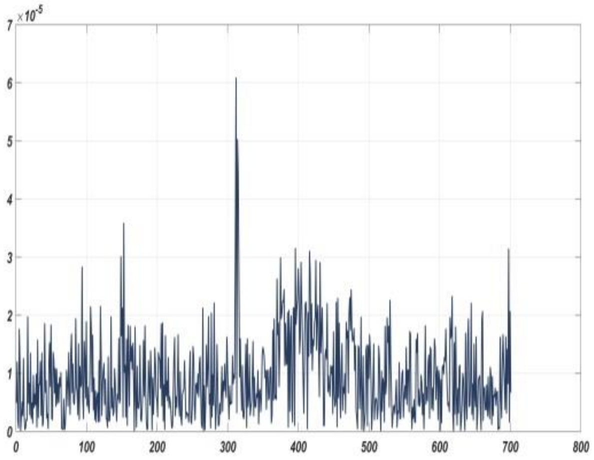

Fig١٢. hypothesis of byte٥-3٥000

Based on this, it can be concluded that to attack the proposed plan, the algorithm runs 35,000 times, plus the time required to transmit Power Trace, which is a long time for the attack, and therefore the level of security is acceptable.

5 CONCLUSION

In this paper, we present an algorithm and increase its security against DPA attacks by combining image secret sharing and ECDSA.

It is hoped that soon we will be able to optimize the proposed algorithm and mask the bytes of the image with random values to make the attack much more difficult because it reduces the relationship between power consumption and the actual data being processed.

REFERENCES

[1] akamoto S, Bitcoin A. A peer-to-peer electronic cash system. Bitcoin–URL: https://bitcoinorg/bitcoin pdf. 2008.

[2] Narayanan A, Clark J. Bitcoin's academic pedigree. Communications of the ACM. 2017;60(12):36-45.

[3] Licheng Wang , Xiaoying Shen , Jing Li , Jun Shao , Yixian Yang , Cryptographic primitives in blockchains, Journal of Network and Computer Applications 127 (2019) 43–58

[4] Zhou X, Tang X, editors. Research and implementation of RSA algorithm for encryption and decryption. Proceedings of 2011 6th international forum on strategic technology; 2011: IEEE.

[5] Marchesan GC, Weirich NR, Culau EC, Weber II, Moraes FG, Carara E, et al., editors. Exploring RSA Performance up to 4096-bit for Fast Security Processing on a Flexible Instruction Set Architecture Processor. 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS); 2018: IEEE.

[6] Liao H-Z, Shen Y-Y. On the elliptic curve digital signature algorithm. Tunghai Science. 2006;8:109-26.

[7] Gozmeh, Mojtaba and Dostari, Mohammad Ali and Yousefi, Hamed, 1397, Presenting a Differential Power Analysis Attack on Mask RSM Reinforcement Implemented on AES Encryption Algorithm and Reinforcement Improvement, Third Conference International Electrical Engineering, Tehran, https: //civilica.com/doc/831920.

[8] Galbraith SD, Gaudry P. Recent progress on the elliptic curve discrete logarithm problem. Designs, Codes and Cryptography. 2016;78(1):51-72.

[9] Jyotiyana D, Saxena VP. A Fault Attack for Scalar Multiplication in Elliptic Curve Digital Signature Algorithm. Computing and Network Sustainability: Springer; 2017. p. 283-91. [10] Gunjan VK, Diaz VG, Cardona M, Solanki VK, Sunitha KVN. ICICCT 2019 – System Reliability, Quality Control, Safety, Maintenance and Management: Applications to Electrical, Electronics and Computer Science and Engineering: Springer Singapore; 2019.

[11] Xuehu Yan, Lintao Liu, Yuliang Lu , Qinghong Gong, Security analysis and classifi- cation of image secret sharing, Journal of Information Security and Applications 47 (2019) 208–216.