Journal homepage www.jzs.univsul.edu.iq
Journal of Zankoy Sulaimani
Part-A- (Pure and Applied Sciences)

Evaluation Study of Original McEliece Cryptosystem Against Side Channel Attack

Newroz N. Abdulrazaq[1]&Thuraya M. Qaradaghi[2]

1 College of Science/Computer Science & IT Department, Salahaddin University-Erbil, Kurdistan Region – Iraq
2 College of Engineering/ Electrical Engineering Department, Salahaddin University-Erbil, Kurdistan Region – Iraq
newroz.adudlrazaq@su.edu.krd , Thuraya.Alqaradaghi@su.edu.krd

Article info

Original:
1 December .2015
Revised: 5 July 2016
Accepted: 31 July 2016
Published online:
20 December 2016

Key Words:
McEliece Cryptosystem
Side Channel Attack

Abstract

Side channel attack is the most efficient attack against original McEliece cryptosystem, especially ball-collision and Bernstein et al.Stern attacks. The modified Stern attack hasan ability to break original McEliece cryptosystem with parameter [1024,524,101] in 1400 days with personal computers.While with 200 clusters CPU breaking could be done in 7 days. While ball-collision attacks have smallerexponent time than Stern algorithm.This paper will present a modified version of Patterson decoding algorithm using a new evaluation for finding error locations.This approach gave the sender an opportunity to choose errors less than identified errors in public key without notifying the receiver;therefore, it reduces the probability of modified Stern attack against McEliece cryptosystem to (0.02) and increases exponent time of ball-collision attack. In this paper alsothe leakage of proposed implementation has beenmeasured using a measurement type for possible leakage in Patterson's decoding algorithm suggested by previous work, and we concluded that the designed system have fewer leakage compared to previous implementation. The workhas done using Visual Studio C#.

Introduction

McEliece cryptosystem harness coding theory in order to make the ciphered message unbreakable. It is based on hardness of finding nearest codeword for a linear binary code, which is considering a NP- hard problem (Non-deterministic Polynomial-time hard). The original McEliece cryptosystem used binary irreducible Goppa code, which is suggested by McEliece [1] with parameters [1024, 524, 101]. Due to its suffering from large size of public key matrix, many variants of McEliece cryptosystem has been proposed [2-4], which makes the size of public key matrix smaller than its original form. Unfortunately, most of proposed cryptosystems have been broken [5, 6],whereas the original form is unbreakable until now.

The best efficient attack against McEliece cryptosystem is side channel attacks (Decoding attacks). An eavesdropper attempt to gain information from the physical implementation (Encryption device) of a cryptosystem in order to cryptanalysis the message (i.e. side channel attacks try to recover the plain text directly from the cipher text without knowing the private keys). Varied types of side channel attacks have been proposed, the strength one among the variants, is that attacks which attempt to find minimum- or low-weight codewords [7-10]. Bernstein et al. [7] modified Stern algorithm [10], and they showed that the ciphered message using 2.4 GHz Core 2 Quad CPU can be breakable in 1400 days, while with a cluster of 200 CPU's can be breakable in 7 days. While Ball-collision attack [8] produce smaller exponent time than Bernstein et al. Stern attack to break ciphered message.Due to above result, Original McEliece Cryptosystem is considered an insecure cryptosystem.

This paper provides a new evaluation for finding error locations in Patterson decoding algorithm to make anopportunity for sender to choose errors less than identified errors in public key without notifying the receiver. This process reduces the probability of modified Stern attack against McEliece cryptosystem to (0.02) and increase exponent time of ball-collision attack. Also this paper measuresthe leakage of proposed implementation using a measurement type for possible leakage in Patterson's decoding algorithm suggested by pervious work [11], and found that the designed system have fewer leakage compared with previous implementations. The implementation has graphical user interface (GUI) using Visual Studio C#.

Coding Theory

Communications over insecure channel (Noisy Channel) may cause errors in the message (as shown in Figure (1)). In insecure channel, errors occurred due to different factors, one of the major factors is natural factors like (lighting, thunder, etc.). Another factor, when eavesdropper attempts to make errors in the message in order to make the message unreadable. Even electrical impulses and neighboring channels may cause errors in the message [12].

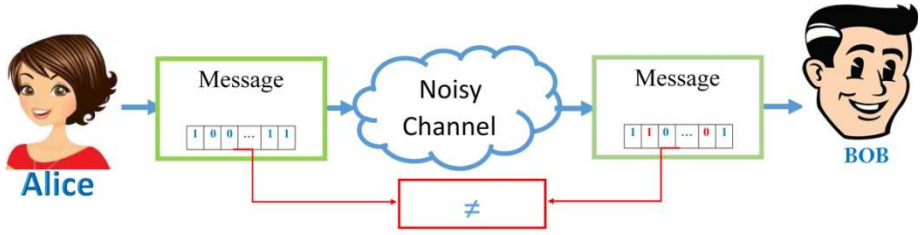

Figure (1): Communication over Noisy Channel.

Error correcting codes harnesses the coding theory in order to detect and fix the errors. The main drawback of error correcting codes is adding redundant bits, which makes the message size larger than it is original size. The reason behind adding redundant bits to the message is to help the error correcting code to detect the errors and then fix it [12].

In general, there are two main types of coding:Source Coding (data compression) and Channel Coding (error correcting). In source coding, message compressed to avoid any effects, while it is transmitted over noisy channel. Whereas in channel coding, the sender attempts to add redundant bits in order to detect and correct errors whenever occurredat the receiver.

One of the major types of forward error correcting codes is linear block codes. It divides the message into independent blocks, i.e. no need for memory. Linear block codes can be classified into hamming codes and polynomial codes. Constructing code words from all polynomials that can be divisible on irreducible generator polynomial are known as polynomial codes *[13]*.

Irreducible Binary Goppa Code

The Binary Goppa code is denoted by Γ(g(z); L), where g(z) is a Goppa generator polynomial of degree t over the extension field GF(2^m) and L is the range of code such that L⊆GF(2^m) [14].

$$g(z) = \sum_{i=0}^{t} g_i z^i \qquad (1)$$

and

$$L = \{\forall \alpha_i \in GF(2^m) \backslash g(\alpha_i) = 0\} \qquad (2)$$

A. Parameters of Goppa Code

Let n=2^m be the length of codeword c, which is constricted by range L, k is a dimension bounded by k ≥n-mt, and minimum distance d ≥2t+1. Then [n, k, d] denotes to the parameters Goppa code Γ(g(z); L) [1].Obviously, according to (1) and (2) in irreducible Goppa code, none of α's yields the condition $g(\alpha_i) = 0$, i.e. L=GF{2^m}.

B. Parity Check Matrix of the Binary Goppa Code

Parity check matrix H is the most important matrix in decoding the message, itis important for detecting and correcting errors. To determine the matrix H:

$$H_{di} = \sum_{\alpha_i \in L} \sum_{k=1}^{t-(d-1)} g_{t-(k-1)} \alpha_i^{t-d+(1-k)} h_i \quad \text{where } 1 \leq d \leq t \quad (3)$$

If c is a code word, then the parity check matrix H should yield $cH^T=0$.

C. Generator Matrix of the Binary Goppa Code

The generator matrix G of the binary Goppa code used to encode and decode message.It is derived from the parity check matrix H, the row space of G is the vectors of null space of H modulo 2 such that: $GH^T = 0$ (4)

D. Encoding Message in Binary Goppa Code

In encoding process, the message is partitioned into blocks of k bits and then multiplying each block by the generator matrix G [24], i.e.:

$$(m_1, m_2, \dots, m_k).G = (c_1, c_2, \dots, c_n) \quad (5)$$

E. Error Correcting of Irreducible Binary Goppa Code

To detect and fix errors in irreducible Goppa code, the steps in Figure (2) should be followed.

Input: Generator polynomial g(Z), Error vector (e) and the encrypted message v=u+e, where u is a plain message (readable message) and v∈ $\Gamma(L, g(Z))$.

Output: The plain message (u).

Step1 (Syndrome): $S(e,Z) \equiv S(u,Z) mod\ g(z)$.

Step2 (T): $T(e,z) \equiv S^{-1}(e,Z) + Z\ mod\ g(Z)$.

Step3 (Tau): $\tau(e,Z) \equiv \sqrt{T(e,Z)}\ mod\ g(Z)$.

Step4 (EEA & Alpha): Use Extension Euclidian Algorithm to find

$$\alpha(e,Z)\ and\ \beta(e,Z)\ \ s.t\ \beta(e,Z)\ \tau(e,Z) \equiv \alpha(e,Z)\ mod\ g(Z)$$

Step5 (Sigma): $\sigma(e,Z) = \alpha^2(e,Z) + \beta^2(e,Z)$

Step6 (Correction): $\begin{cases} Find\ roots\ of\ \sigma(e,Z). \\ Find\ indexes\ of\ roots\ in\ order\ to\ find\ error\ postions. \\ Correct\ the\ message\ to\ get\ (u) \end{cases}$

Figure (2): Algorithm of Finding and Correcting Errors in Irreducible Goppa Code.

F. Extension Euclidian Algorithm Stop Condition

The dynamic errors (dynamic errors means that the sender have the opportunities to choose number of errors less than the published one), changing the condition of extended Euclidean algorithm to obtain the sigma polynomial (Error calculator) as follows:

Let u, v, and rem are polynomials in the form of array, thoseare calculated in extended Euclidean algorithm for step4 (as shown in Figure (2)) and k is the algorithmiteration. Then, the stop condition should be in the form:

Degree(rem[k]) < Degree(g(z))/2 and Degree(rem[k-1]) ≥ Degree(g(z))/2

After that, α (error locator) and β, should be calculated as follows:

$$\begin{cases} \alpha = rem[k] and \beta = v[k+1] if k = 0 \\ \alpha = rem[k] and \beta = v[k+1] if degree(rem[k-1]) - degree(rem[k]) \neq 1 \\ \alpha = rem[k-1] and \beta = v[k]\ if elsewhere \end{cases} \quad (6)$$

G. Decoding a Message in Binary Goppa Code

When the errors are fixed, the received message can be decoded [14], using equation (5), which can be written as matrix representation:

$$G^T . \begin{pmatrix} m_1 \\ \vdots \\ m_k \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \quad (7)$$

Gauss elimination method is applied in order to remove generator matrix G:

$$\left(G^T \middle| \begin{matrix} c_1 \\ \vdots \\ c_n \end{matrix} \right) \sim \cdots \sim \left(\begin{matrix} I_k \\ \hline P \end{matrix} \middle| \begin{matrix} m_1 \\ \vdots \\ m_k \end{matrix} \right) \quad (8)$$

Where I_k is the identity matrix with size k × k and P is a matrix with size (n-k) × (k+1).

Original McEliece Cryptosystem

A. Key Generation of McEliece Cryptosystem

Public key cryptosystem based on two types of keys (public and private), which are linked together mathematically. A public key is published and used to cipher a message, while a private key must keep it secret and use it to decipher the message. To prepare keys depending on Goppa code, the following steps as shown in Figure (3)should be followed [15]:

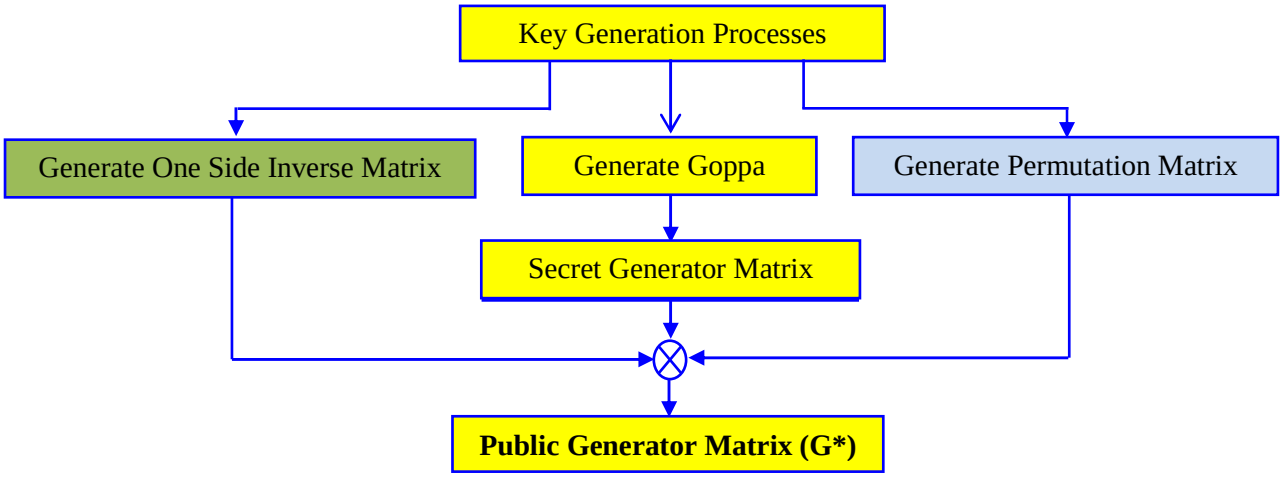

Figure (3): General Block Diagram for Key Generation Process

B. Encryption Process in McEliece Cryptosystem

To encrypt any message, the block diagram asshown in Figure (4)) should be followed [15]:

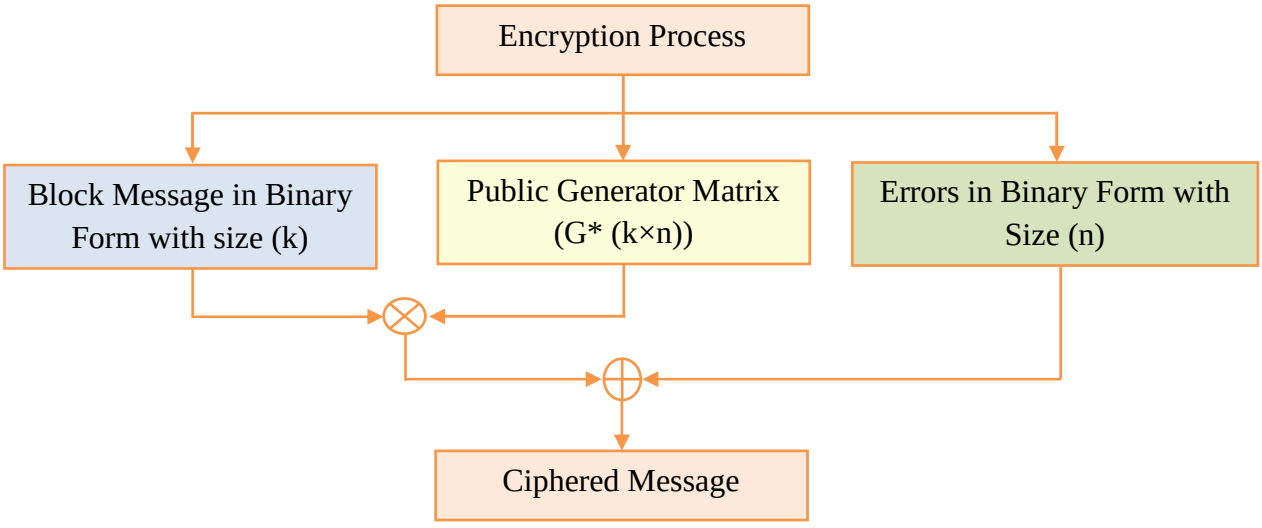

Figure (4): General Block Diagram for Encryption Process

C. Decryption Process in McEliece Cryptosystem

To recover plain message from cipher message c, the following steps should be followed (as shown in Figure (5) [15]:

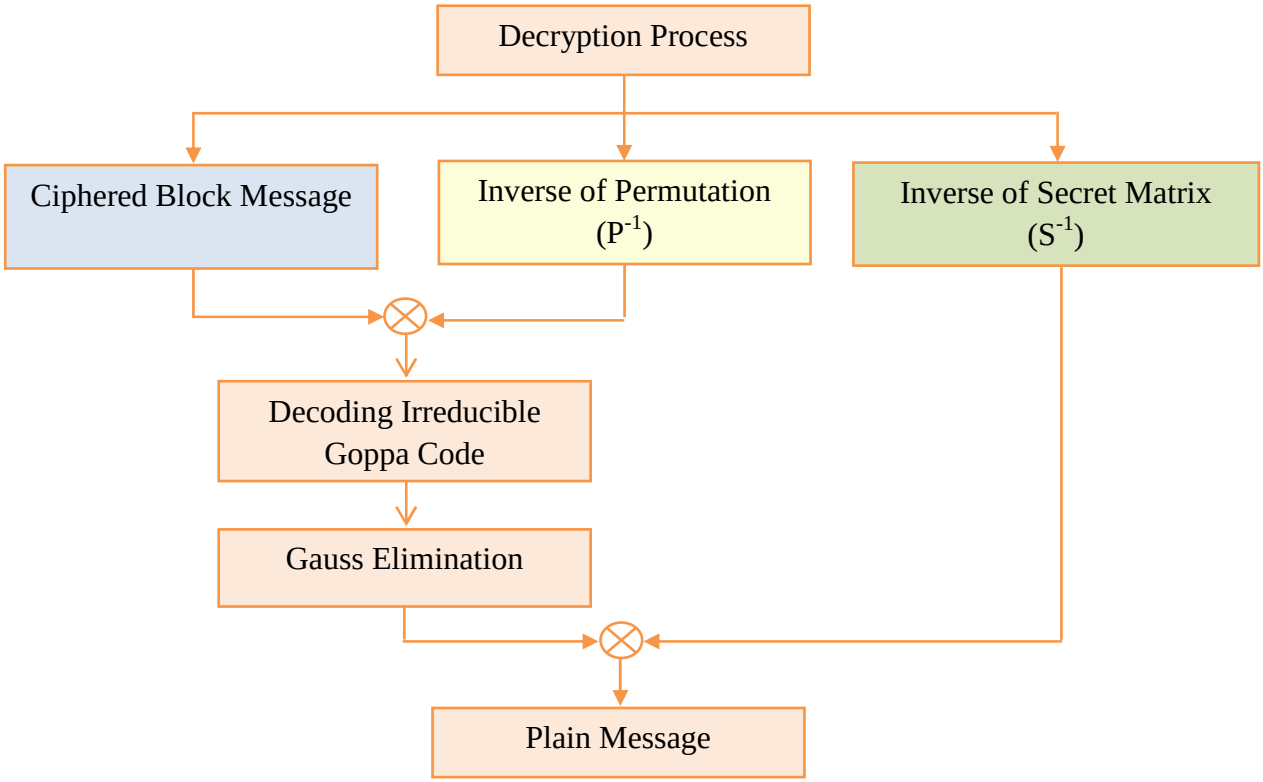

Figure (5): General Block Diagram for Decryption Process

Increasing attacktime against (Bernstein- Stern) and Ball-Collision Attacks

Side channel attack is the most effective attack against McEliece cryptosystem. Different variants of side channel were introduced. The best efficient type was improved by Bernstein et al. [7] for Stern attack [10](as shown in Figure (6)). They modified Stern attack in order to decrease the time attack against McEliece Cryptosystem. The results show that the original McEliece cryptosystem with parameters [1024, 524,101] can be breakable in 1400 days for 2.4 GHz Core 2 Quad CPU. While for a cluster of 200 CPU's needs 7 days to break it with required parameters. Ball-collision attacks [8] (as shown in Figure (7)) show better time attack against McEliece cryptosystem. The procedures of two previous algorithms are based on several parameters,the most efficient parameter is the number of errors (w=t) should be occurred in one block message, which is a public key cryptosystem.

The designed system gives the sender opportunities to choose number of errors between 1 to published number of errors (t) without notifying the receiver (i.e. the receiver has ability to fix errors between 1 to t). This opportunity is satisfied, whenever conditions in section III-F are used. Therefore, the designed system increases the time computation of attack against proposed implementation. For example, let the original McEliece cryptosystem be with parameters [1024, 254, 101] where t=50 have been chosen, then, an attacker has probability of (0.02) for choosing correct number of errors, or try 50 attempts in order to find the correct errors (i.e. 50 × 1400 = up to 70000 days) for 2.4 GHz Core 2 Quad CPU. While for a cluster of 200 CPU's needs (50 × 7 = up to 350) days.

Input: A generator matrix G, a cipher text $\varphi = y \in F_q^n$ and a parameters $\ell, p \in \mathbb{N}$.

Output: An error vector e of weight w.

1. Pick a random information set I and compute y_I, G_I, and $G^{'(1)}$.
2. Determine $y' = y - y_I G'$.
3. Pick at random a subset $X \subset I$ of size k/2 and set Y=I\X, then select at random a size-ℓ set Z in {1, 2, …, n}\I.
4. For any size-P subset A = $\{a_1, \dots, a_p\} \subset X$ form the set U as the union of the sets $\{y' - \hat{g} : \hat{g} = \sum_{i=1}^{p} x_i \, G'_{a_i}\}$ over all choices of $x_1, \dots, x_p \in F_q \{0\}$.
5. For any size-P subset B = $\{b_1, \dots, b_p\} \subset Y$ form the set V as the union of the sets $\{\hat{h} : \hat{h} = \sum_{i=1}^{p} y_i \, G'_{b_i}\}$ over all choices of $y_1, \dots, y_p \in F_q \{0\}$.
6. For Each pair (A, B): Look for collisions, i.e. vectors u ∈ U and v ∈ V such that uz = vz then write e= u – v. if wt(e) = w then return e
7. Go back to step 1.

Figure (6): The Generalized Stern Algorithm.

Input: A generator matrix G, a cipher text $\varphi = y \in F_q^n$ and a parameters ℓ_1, ℓ_2, $p_1, p_2, q_1, q_2, k_1, k_2 \in \mathbb{N}$ such that $p_1 + p_2 + q_1 + q_2 \leq w$ and $k_1 + k_2 = k$

Output: An error vector e of weight w.

1. Pick a random information set I and compute y_I, G_I, and $G^{'(1)}$.
2. Determine $y' = y - y_I G'$.
3. Partition I in two subsets X, Y of size k_1 and k_2.
4. Select in {1, 2, … , n}\I two subsets Z_1 and Z_2 of size ℓ_1 and ℓ_2.
5. Calculate "weight sums" having respective weights p_1, p_2 for X and Y, and q_1, q_2 for Z_1 and Z_2
6. For each choice of the above sets:
   Look for collisions, i.e. vectors e such that e_I has exactly weight $p_1 + p_2$ and $e_{Z_1 \cup Z_2}$ has exactly weight $q_1 + q_2$.
   If $wt(e_{(1,\dots,n)\setminus(I \cup Z_1 \cup Z_2)}) = w - p_1 - p_2 - q_1 - q_2$ then return e.
7. Go back to step 1.

Figure (7): Ball- Collision Attack.

Timing- Side Channel Attack Measurement Tool

Timing attack is a type of side channel attack in which the eavesdropper tries to analyze the time taken to implement cryptographic algorithms. The success of eavesdropper to attack on any cryptosystem needs knowledge of multi factors that effect on executing the algorithms [16], such as CPU, RAM, designed cryptosystem and the used algorithms.

A. Collecting Data

Due to studying the leakage of original McEliece cryptosystem on decoding algorithm stage, information about time consuming and hamming weights on each steps in decoding algorithm (as shown in Figure (2)) has been recorded (as shown in Tables 1 and 2) by implementing 100 random messages for each of 10 random pair keys. Also information about degree of processed polynomial has been recorded (as shown in table 3).

Table 1: Measurement of Computation Time in Respect to HW(e)

HW(e)	Syndrome	T	Tau	EEA	Sigma	Correction
1	8.2E+02	9.6E+03	1.8E+05	7.0E+04	5.6E+01	2.1E+03
2	8.5E+02	2.6E+04	1.7E+05	7.1E+04	8.2E+01	3.8E+03
3	8.3E+02	1.0E+05	2.0E+05	9.8E+03	1.8E+02	6.3E+03
⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	8.2E+02	1.3E+05	1.9E+05	7.4E+04	5.6E+02	4.2E+04

Table 2: Measurement of HW of polynomial Processed in Respect to HW(e)

HW(e)	Syndrome	T	Tau	alpha	Sigma
1	41	4	4	4	5
2	41	8	8	8	9
3	40	41	40	8	16
⋮	⋮	⋮	⋮	⋮	⋮
10	40	40	41	24	44

Table 3: Measurement of degree of polynomial Processed in Respect to HW (e)

HW(e)	Syndrome	T	Tau	alpha	Sigma
1	9	0	0	0	1
2	9	2	1	1	2
3	9	9	9	1	3
⋮	⋮	⋮	⋮	⋮	⋮
10	9	9	9	5	10

The designed cryptosystem is implemented for original Goppa code Γ(L; g(z)), where m=8 and t=10 in order to provide notable graphs. After the recording process completed, the average values for time consuming, hamming weight of polynomials processed with respect to HW(e), and degree of polynomials processed with respect to HW(e) have been evaluated.

B. Time Analyzing

As shown in Figures (8 to 11), the graph for the determined tables has been drawn. Figure (8) shows that step 3 in Patterson's decoding algorithm (which is determining the square root) takes more time.

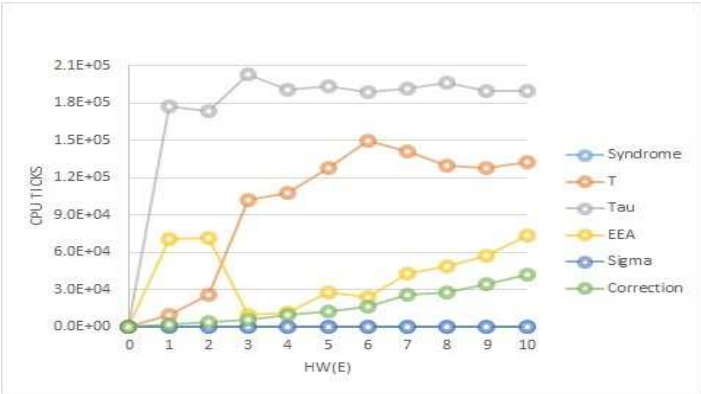

Figure (8):Computation Time with Respect to HW(e).

In order to analyze which steps of decoding algorithm are based on the hamming weight of errors, the graph in the Figure (9) are drawn, which is the same as in Figure (8), but only with step1 and step5.

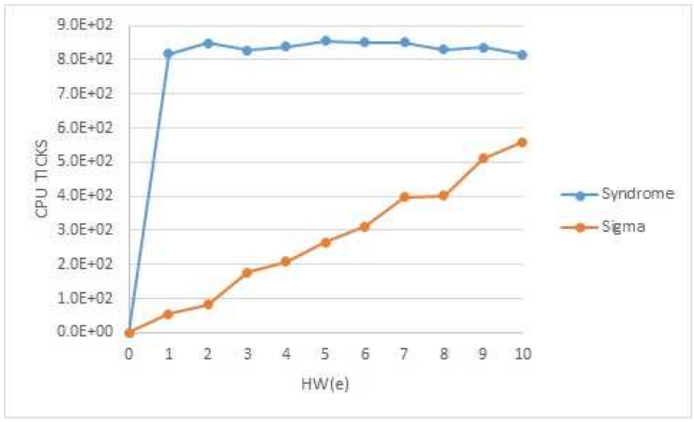

Figure (9): Computation Time with Respect to HW(e) forSyndrome and Sigmaonly.

Similarly, graph for hamming weight of processed shown in Figure (10), and graph for degree of polynomial is shown in Figure (11). It shows the impact of HW(e) on computation time steps.

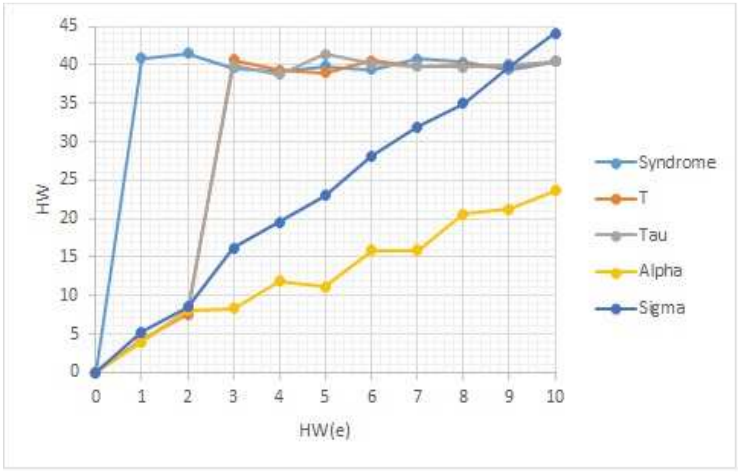

Figure (10):HW of Polynomial Processed with Respect to HW(e).

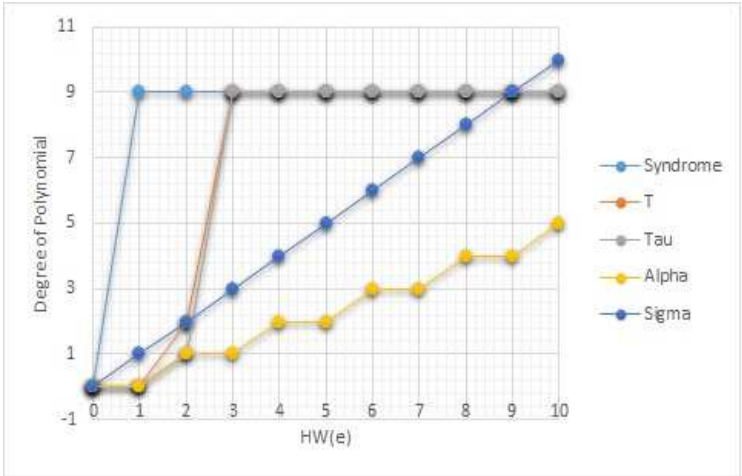

Figure (11):Polynomial degree Processed with Respect to HW(e).

In general, it is hard to compare the proposed implementation with the other implementations due to mentioned factors (CPU, RAM, designed cryptosystem, etc). Then, a comparison between the implemented one and the suggested type by Mark Repka [11] has been done, and some points have been concluded:

1. The HW of polynomial processed in the proposed implementation is higher than thesuggested type by Mark Repka (as shown in Figure (10)).

2. All steps have faster computation time than the implemented cryptosystem, except step4 in decoding algorithm (as shown in Figures (8 and 9)).
3. The EEA algorithm designed to work with all parts of implementation (e.g. factoring the polynomial, Irreducibility test, determining sigma in separable and irreducible Goppa code, etc.), while other implementations rewrite EEA at least in two forms,which considered the reason behind the slower of step4 in proposed implementation.

Conclusions

This paper presents a modified version for Patterson decoding algorithm using a new evaluation for finding error locations after stop condition in Extension Euclidean algorithm. This approach increases the complexity of original McEliece cryptosystem against side channel attack due to the sender has opportunity to choose errors less than the identified one without notifying the receiver.

Also in this paper, the impact of Hamming weight in Patterson decoding algorithm have been studied in order to specify the leakage of decoding algorithm for timing-side channel attack. The following have been observed:

a) The most time consuming step in the designed system of decoding algorithm is determining the square root of the polynomial.
b) The results show which computation time steps of Patterson decoding algorithmare based on Hamming weight of error vector.
c) The designed system has faster computation time of decoding algorithm compared to the previous implementation.

References

[1] R. McEliece, "*A Public-Key Cryptosystem Based on Algebraic Coding Theory"*, Technical report, NASA, (1978).

[2] T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani, "*Reducing Key Length of the McEliece Cryptosystem*", In B. Preneel, editor, AFRICACRYPT, Lecture Notes in Computer Science, Vol. 5580, pp 77-97. Springer, (2009).

[3] E. M. Gabidulin, A. V. Ourivski, B. Honary, and B. Ammar, "*Reducible rank codesand their applications to cryptography*", IEEE Transactions on Information Theory, Vol. 49, No.12, pp 3289-3293, (2003).

[4] C. Monico, J. Rosenthal, and A. Shokrollahi, "*Using low density parity check codesin the McEliece cryptosystem*", IEEE International Symposium on Information Theory, ISIT 2000, pp 215. IEEE, (2000).

[5] J. -C. Faug*e′*re, A. Otmani, L. Perret, and J. -P. Tillich, "*Algebraic Cryptanalysis of McEliece Variants with Compact Keys*", EUROCRYPT, Lecture Notes in Computer Science, Vol. 6110, pp 279-298. (2010).

[6] R. Overbeck, "*A New Structural Attack for GPT and Variants*", E. Dawson and S. Vaudenay, editors, Mycrypt, Lecture Notes in Computer Science, Vol. 3715, pp 50-63. (2005).

[7] D. J. Bernstein, T. Lange, and C. Peters, *"Attacking and Defending the McEliece Cryptosystem*", J. Buchmann and J. Ding, editors, PQCrypto, Lecture Notes in Computer Science, Vol. 5299, pp 31-46. (2008).

[8] D. J. Bernstein, T. Lange, and C. Peters, "*Smaller Decoding Exponents: Ball- Collision Decoding*", P. Rogaway, editor, CRYPTO, Lecture Notes in Computer Science, Vol. 6841, pp 743-760. (2011).

[9] A. Canteaut and F. Chabaud, "*A New Algorithm for Finding Minimum- Weight Words in a Linear Code: Application to McEliece Cryptosystem and to Narrow- Sense BCH Codes of Length 511*", IEEE Transactions on Information Theory, Vol. 44, No.1, pp 367-378, (1998).

[10] J. Stern, *"A Method for Finding Codewords of Small Weight"*, G. D. Cohen and J. Wolfmann, editors, Coding Theory and Applications, Lecture Notes in Computer Science, Vol. 388, pp 106-113. (1988).

[11] Repka Marek, "*Leakage Measurement Tool of McEliece PKC Calculator*", TELEDU-17, WSEAS, ISBN 978-1-61804-262-0, Istanbul, pp 128, (2014).

[12] Gadoulean Maximilien, "*Cryptosystem Using Error-Correction Codes Based on the Rank Metric*", Lehigh University, Master Thesis (2005).

[13] Hudde Hans, "*Development and Evaluation of Code-Based Cryptography Library for Constrained Devices*", Ruhr University, (2013).

[14] Qaradaghi, T., Abdulrazaq, N. "*Comparison between Separable and Irreducible Goppa Code in McEliece Cryptosystem*", World Academy of Science, Engineering and Technology, International Science Index 106, International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol. 9, No. 10, pp 1717 - 1723.(2015).

[15] Abdulrazaq N.; Qaradaghi T., "*Cryptosystem Based on Error Correcting Codes*", Salahaddin University-Erbil, Zanco Journal of Pure and Applied Science, Vol. 28, No. 2, pp 99-109. (2016).

[16] Edoardo Persichetti, *"Improving the Efficiency of Code-Based Cryptography*", Auckland University, PHD Thesis, Nov. (2012).