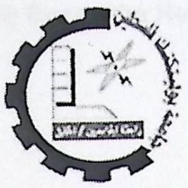


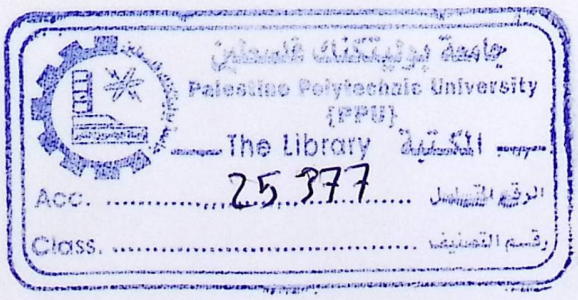
Palestine Polytechnic University
Deanship of Graduate Studies and Scientific Research
Master of Informatics

A Practical One Way Hash Algorithm (POH) Based On Non-Invertible Matrix BY using matrix Multiplications

Submitted By
Mohammed Said Abutaha

In Partial Fulfillment of the Requirements for the Degree
Master of Informatics

July, 2011



The undersigned hereby certify that they have read, examined and recommended to the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University the approval of a thesis entitled:

A Practical One Way Hash Algorithm (POH) Based On Non-Invertible Matrix BY using matrix Multiplications

Submitted by **Mohammed S. Abutaha**

in partial fulfillment of the requirements for the degree of Master in Informatics

Graduate Advisory Committee:

Committee Chair Name, University:
Dr. Ismail Romi, Palestine Polytechnic University
Signature:_____ Date: 30.11.2011

Committee Member Name (Supervisor), University:
Dr. Radwan Tahboub, Palestine Polytechnic University
Signature:_____ Date: 30/11/2011

Committee Member Name, University:
Dr. Mohammed Aldasht, Palestine Polytechnic University
Signature:_____ Date: 30.11.2011

External Committee Member Name, University:
Dr. Rushdi Hamamreh, AL-Quds University
Signature:_____ Date: 30.11.2011

Thesis Approved

Prof. Dr. Karim Tahboub
Dean of Graduate Studies and Scientific Research
Palestine Polytechnic University

Signature:_____ Date: 4.12.2011

The undersigned hereby certify that they have read, examined and recommended to the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University the thesis entitled:

A Practical One Way Hash Algorithm (POH) Based On Non-Invertible Matrix Multiplications

Submitted by **Mohammed S. Abutaha**

in partial fulfillment of the requirements for the degree of Master in Applied Mathematics

Graduate Advisory Committee:

Committee Chair Name, University:
Dr. Ismail Romi, Palestine Polytechnic University
Signature:_____

Committee Member Name (Supervisor), University:
Dr. Radwan Tahboub, Palestine Polytechnic University
Signature:_____

Committee Member Name, University:
Dr. Mohammed Aldasht, Palestine Polytechnic University
Signature:_____

External Committee Member Name, University:
Dr. Rushdi Hamamreh, AL-Quds University
Signature:_____

Thesis Approved

_____
Prof. Dr. Karim Tabbou
Dean of Graduate Studies
Palestine Polytechnic University

Signature:_____

Abstract

Nowadays, Cryptography plays a major role in protecting information. Hash function is a well-defined procedure that digests a large data chunks into a small one. The returned value from hash function is called hash code. Hash codes can be used for data integrity purposes while storing or transmitting data. It is well known that Hash algorithms work in one way and cannot be reversed. Many hash algorithms and standards exist today, Among these algorithms, the Message Digest algorithms (MD4, MD5) and Secure Hash Algorithms(SHA1,SHA2 and SHA-512).

Since its invention by Lester S. Hill in 1929, the Hill cipher model which is one of the most famous symmetric cryptosystems that can be used to protect information from any unauthorized access. Hill cipher uses matrix operations to produce cipher matrix from a data and key matrices. Hill cipher requires the inverse of the key matrix for decryption. This inverse depends on a suggested modular number, but the matrix which doesn't have a prime determinant relative to a previous suggested number doesn't have an inverse. Non-invertible key matrix is the main problem of Hill Cipher. This problem leads to many other sub problems such as the disability of decrypting any encrypted text.

In this research, we made use of the aforementioned problem in Hill cipher, namely the non-invertible matrix problem, to design a novel one way hash algorithm. The first round of the proposed algorithm depends on the multiplication of non-invertible matrix with the plaintext message in column vector. Then, we use the output of the first step to make a digest for these data chunks and generate the final hash value.

A comparison between our practical one way hash algorithm and Message Digest (MD5), Secure Hash Algorithm (SHA1), and Secure Hash Algorithm(SHA-512) has been made by using the data size and matrix size factors to compare the time per second. Our algorithm`s security power against brute force attacks and dictionary attack has been improved and discussed. We give a strong indication that our algorithm is resistant against collisions by using hamming distance algorithm. We also give a proof that our algorithm satisfies the one way hash algorithm properties.

Implementations of our algorithm and other algorithms shows that our algorithm has similar performance as MD5 algorithm for different matrix size, and it is more secure than MD5 against brute force attack and collision attacks, also our algorithm has a better performance than SHA1

algorithm for different matrix size, and it has a better performance than SHA-512 with less security.

The key contribution of this research is the development of a new one way hash algorithm based on Hill Cipher. The second important contribution is to design an algorithm to convert an invertible matrix to non-invertible one. The third contribution is to solve the dictionary attack problem by using salt algorithm.

ملخص

ان قضية امن المعلومات تعد من اهم قضايا العصر الحديث حيث ان الاعتماد الحالي على الانترنت اصبح في جميع مجالات الحياة؛ فأصبحت قضية حماية هذه المعلومات من الاختراق قضية مهمة. عملية التشفير هي عملية تجمع بين علم الرياضيات وعلم الحاسوب، هي العلم والمقدرة على حماية البيانات من الاختراق حيث تستخدم في معظم المجالات والتطبيقات التقنية والعملية. الطلب على عملية التشفير في تزايد لحماية البيانات لكن ما يتم حفظه وتأمينه اليوم من الممكن ان يتعرض للاختراق في الغد.

يشمل علم التشفير مجموعة من الخوارزميات والتقنيات لتحويل البيانات الى شكل اخر بحيث تظهر محتوياتها بشكل غير مقروء وغير قابل للتفسير لأي شخص ليس لديه الصلاحيات للقراءة او الكتابة على هذه البيانات. الهدف الرئيسي من استخدام خوارزميات التشفير حماية المعلومات والبيانات بهدف تحقيق الخصوصية، التكاملية وامكانية الوصول للمصادر والخدمات التي يقدمها نظام المعلومات. هناك مجموعة من المخاطر التي من الممكن أن تؤذي نظم المعلومات الحاسوبية ومنها ما يكون موجه لخوارزمية التشفير نفسها مثال على ذلك خوارزمية التشفير التماثلية قد تواجه خطر المهاجمين لمحتويات الخوارزمية وتكوينها او خصائصها و معرفة أجزاء من الرسالة الاصلية التي يريد المستخدم لهذه الخوارزمية استخدامها مما يؤدي الى معرفة الرسالة كاملة اوقد يعتمد المهاجمين على تجربة مجموعة من مفاتيح التشفير المحتملة على جزء النص المشفر ومن الممكن في هذه الحالة ان يصل الى الرسالة الاصلية قبل التشفير.

من المعروف ان خوارزمية التشفير ذات الاختزال تعمل بالاتجاه الواحد ولا يمكن الوصول الى اصلها او استرجاعه وقد قمنا باقتراح خوارزمية مشابهة تعمل بالاتجاه الواحد وذلك باستخدام ضرب المصفوفات. تقنية هيل تعتمد على التشفير باستخدام مفتاح على شكل مصفوفة واستخدام معكوس هذه المصفوفة كمفتاح لفك التشفير. وتعتبر من الخوارزميات المهمة التي قدمت حلا لمشكلة تحليل تكرارية الحروف داخل نص معين باستخدام التشفير لأكثر من ثلاثة حروف دفعة واحدة.

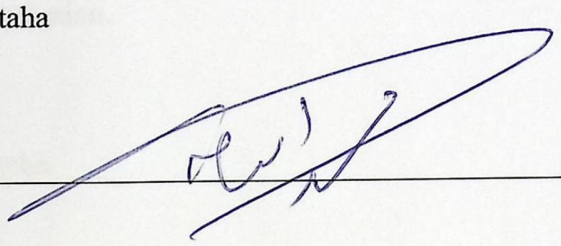
معكوس المصفوفة المستخدم في تقنية هيل غير متاح دائما ومن هذا المنطلق يمكن استخدام المصفوفات التي لا يوجد لها معكوس لاشتقاق تقنية جديدة لخوارزمية بعثرة عملية ذات الاتجاه الواحد. وبهذا نقدم نموذجا افضل لإيجاد قيمة مبعثرة حيث يتميز هذا النظام بقدرته على مواجهة خطر هجوم القاموس وذلك عن طريق اضافة القيمة الملحية على البيانات مما يجعل من الصعب اعداد قاموس من القيم المبعثرة. في هذا البحث تم عمل مقارنة بين النموذج المقترح وانظمة MD5,SHA1,SHA512.

خلال عملنا في هذا البحث قمنا بنشر ثلاث أوراق علمية في مجلات عالمية. في الورقة العلمية الأولى قمنا بعرض النموذج الرياضي للخوارزمية وقمنا بإثبات المتطلبات الخاصة بخوارزمية التشفير ذات الاتجاه الواحد. في الورقة العلمية الثانية والتي تم تقديمها باللغة العربية ونشرها في مجلة علوم الحاسبات العربية تم تقديم النموذج المقترح ومقارنة بينه وبين النماذج المستخدمة حالية في ايجاد قيمة مبعثرة. في الورقة العلمية العلمية الثالثة قمنا بعمل مسح للخوارزميات المستخدمة في التشفير حاليا وقدمنا مقارنة بينها وقدمنا ايضا مجموعة من الامثلة عليها.

DECLARATION

I declare that the Master Thesis entitled "*A practical One Way Hash Algorithm (POH) Based on Non-invertible Matrix using Matrix Multiplications*" is my own original work, and hereby certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgment is made in the text.

Mohammed Said Abutaha

Signature: 

Date: 1 - 12 - 2011

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for the master degree in Informatics at Palestine Polytechnic University, I agree that the library shall make it available to borrowers under rules of the library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from, reproduction, or publication of this thesis may be granted by my main supervisor, or in his absence, by the Dean of Graduate Studies and Scientific Research when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Mohammed Said Abutaha

Signature:_____ Date:_____

DEDICATION

I dedicate this work to my parents and my wife. Without their patience, understanding, there continuous support, and most of all love, the completion of this work would not have been possible during my vital educational years. Also their endless patience and encouragement when it was most required.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my supervisor Dr. Radwan Tahboub. He offered me great freedom on choosing my favorite research topic and developing my research interests. He continuously provided me with help, encouragement, and with extensive knowledge. He also guided and helped me to publish my first three papers in my academic profile during my thesis. His ideas have been a source of inspiration for this work. I thank the examining committee, all of my colleagues and relatives for their support. I also thank all of my lecturers, especially, Dr. Mohammed Aldasht and Eng. Mousa Farajalla who gave me a lot of knowledge, and answered all my questions.

Abbreviations

Abbreviations	Full term
1. **AES**	Advanced Encryption Standard
2. **Adj**	Adjoint Matrix
3. **C**	Cipher Text Message
4. D_K	Decryption Algorithm
5. **Det**	Matrix Determinant
6. E_K	Encryption Algorithm
7. **Gcd**	Greatest Common Divisor
8. **H**	Hash Function
9. **Key_inv**	Inverse Of Key Matrix
10. **Fix**	Convert To Integer Function
11. **MD5**	Message Digest
12. **Mod**	Modular Arithmetic
13. **OWHF**	One Way Hash Function
14. **POH**	Practical One Way Hash
15. **P**	Plaintext Message
16. **RSA**	Rivest-Shamir-Adleman
17. **SHA**	Secure Hash Algorithm
18. **NIST**	National Institute Of Standards And Technology
19. **Word**	A Group Of Either 64 Bits Relative To SHA-512
20. $H^{(i)}$	i^{th} Hash Value, Where $H^{(0)}$, Is The Initial Hash Value; $H^{(N)}$ Is The Final Hash Vale
21. $H_j^{(i)}$	The j^{th} Word Of i^{th} Hash Value

Table of Content

Page

ABSTRACT IV
ملخص VI
TABLE OF CONTENT XII
LIST OF FIGURES XIV
LIST OF TABLES XV
CHAPTER 1 2
1.1 OVERVIEW 2
1.2 MOTIVATION 3
1.3 PROBLEM STATEMENT 3
1.4 PROPOSED SOLUTIONS 3
1.5 RESEARCH METHODOLOGY 4
1.6 OBJECTIVES 4
1.7 CONTRIBUTIONS 4
1.8 LITERATURE REVIEW 5
1.9 THESIS OUTLINES 8
CHAPTER 2 10
2.1 INTRODUCTION TO CRYPTOGRAPHY 10
2.2 CRYPTOGRAPHY GOALS: 10
2.3 BASIC TERMINOLOGY OF CRYPTOGRAPHY 10
2.4 SYMMETRIC AND ASYMMETRIC ENCRYPTION 12
2.5 MATHEMATICS OF CRYPTOGRAPHY 14
-INTEGER OPERATIONS 14
-MATRIX OPERATIONS 15
-MODULAR ARITHMETIC 15
2.6 HILL CIPHER 18
2.7 HASH ALGORITHMS 21
2.8 COMPARISON BETWEEN MD5, SHA1 25
2.9 SECURE HASH ALGORITHM-512 25
CHAPTER 3 31
3.1 INTRODUCTION 31
3.2 PROPOSED MODEL 31

3.3 MATHEMATICAL MODEL 31

3.4 ALGORITHM STEPS: 32

3.5 PROOF OF PRACTICAL ONE WAY PROPERTIES FOR HASH ALGORITHM REQUIREMENTS: 37

3.7 SUMMARY 41

CHAPTER 4 44

4.1 SIMULATION AND RESULTS ANALYSIS: 44

-COMPARISON BASED ON MATRIX SIZE 1X1 44
-COMPARISON BASED ON MATRIX SIZE 2X2 45
-COMPARISON BASED ON MATRIX SIZE 3X3 46
-COMPARISON BASED ON MATRIX SIZE 4X4 47
-COMPARISON BASED ON MATRIX SIZE 5X5 48
-COMPARISON BASED ON MATRIX SIZE 6X6 49
-COMPARISON BASED ON MATRIX SIZE 7X7 50
-COMPARISON BASED ON MATRIX SIZE 8X8 51
-COMPARISON BASED ON MATRIX SIZE 9X9 52
-COMPARISON BASED ON MATRIX SIZE 10X10 53
-COMPARISON BASED ON MATRIX SIZE 11X11 54
-COMPARISON BASED ON MATRIX SIZE 12X12 55

4.2 RESULTS: 56

4.3 SECURITY OF POH AGAINST BRUTE FORCE ATTACK: 57

4.4 SECURITY OF POH AGAINST DICTIONARY ATTACK: 59

4.5 SECURITY OF POH AGAINST COLLISIONS: 59

4.6 SUMMARY 61

CHAPTER 5 62

5.1 CONCLUSIONS 62

5.2 FUTURE WORK 63

REFERENCE 62

APPENDIX A 62

APPENDIX B 65

APPENDIX C 69

List of Figures

FIGURE PAGE

FIGURE 2.1: SIMPLIFIED MODEL OF CONVENTIONAL ENCRYPTION.....13
FIGURE 2.2: SIMPLIFIED MODEL OF ASYMMETRIC ENCRYPTION.....13
FIGURE 2.3: COMMON DIVISORS OF TWO INTEGERS.....14
FIGURE 2.4: FLOWCHART OF FINDING GREATEST COMMON DIVISOR.....15
FIGURE 2.5: FLOWCHART OF FINDING INVERSE OF MATRIX RELATIVE TO MODULAR VALUE.....16
FIGURE 2.6: MESSAGE AUTHENTICATION.....22
FIGURE 2.7: SECURE HASH FUNCTIONS.....24
FIGURE 2.8: SHA-512 STRUCTURE.....28
FIGURE 3.1: SALT ALGORITHM.....36
FIGURE 3.2: POH BLOCK DIAGRAM.....38
FIGURE 4.1: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 1X1.....44
FIGURE 4.2: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 2X2.....45
FIGURE 4.3: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 3X3.....46
FIGURE 4.4: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 4X4.....47
FIGURE 4.5: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 5X5.... 48
FIGURE 4.6: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 6X6.....49
FIGURE 4.7: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 7X7.....50
FIGURE 4.8: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 8X8.....51
FIGURE 4.9: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 9X9.....52
FIGURE 4.10: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 10X10.....53
FIGURE 4.11: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 11X11.....54
FIGURE 4.12: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 12X1...55
FIGURE 4.13: TIME PER SECOND OF POH IN DIFFERENT MATRIX SIZE AND DIFFERENT DATA SIZE.....56
FIGURE 4.14: HAMMING DISTANCE BETWEEN TWO HASH VALUE STRING IN DIFFERENT DATA SIZE FOR MD5, POH, SHA.....60
FIGURE 4.15: HAMMING DISTANCE ALGORITHM.....60

List of Tables

TABLE page

TABLE 3.1: MODEL TO MAKE PADDING.....32

TABLE 3.2: MODEL TO CONVERT INVERTIBLE MATRIX TO NON- INVERTIBLE MATRIX.....34

TABLE 4.1: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 1X1.....44

TABLE 4.2: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 2X2.....45

TABLE 4.3: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 3X3.....46

TABLE 4.4: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 4X4.....47

TABLE 4.5: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 5X5.....48

TABLE 4.6 : TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 6X6.....49

TABLE 4.7: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 7X7.....50

TABLE 4.8 : TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 8X8.....51

TABLE 4.9: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 9X9.....52

TABLE 4.10: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 10.....53

TABLE 4.11: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 11.....54

TABLE 4.12: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 12x12.....55

TABLE 4.13: SECURITY OF POH.....57

TABLE 4.14: NUMBER OF TRIES NEEDED BY THE HACKER TO CRYPTANALYSIS THE MATRIX.....57

Chapter 1

1.1 Overview

Cryptography has become increasingly used to provide security of information. But, things that can be secure today could be broken in the future. Cryptology is the "science of codes and ciphers". Cryptology includes many algorithms and techniques which help to transfer data in a way that makes their content unreadable to anyone who does not have permission to read or write on these data [3],[4],[59],[62].

HMACs provides integrity by using the hash algorithm means combined with a participated secret key. The hash differs from the digital signature; the hash uses the same key in both the sender and the receiver sides; but the digital signature uses a public key infrastructure. "Hash functions are also sometimes referred to as message digests or one-way transforms. One-way transforms or functions are so named for two reasons: each party must perform the computation on their respective end, and because it is easy to go from message to digest but mathematically infeasible to go from digest to message. Conversely, two-way functions can go either way; encryption schemes are examples of two-way functions"[4].

The hash value code is actually a Message Integrity Code (MIC) that each party must calculate to verify the message. For example, the sender uses an HMAC algorithm and shared key to compute the hash for the message. The receiver must perform an HMAC computation on the received message, and compare it to the original. If the message has changed in transit, the hash values are different and the packet is rejected[59].

One way hash algorithm changes messages or text into a fixed string of digits. Usually for systems security integrity, confidentiality and availability. The one way means that it's hard to recover the original text from the hash value string. A one-way hash function is used to create digital signatures. Which in its turn identify and authenticate the sender of a digitally distributed message[5],[18].

1.2 Motivation

Cryptography is considered as a mixture of both mathematics and computer science. It's the study and the ability of hiding data; it's used in many technological and business applications. "Computer Security aims to protect an automated information system" in order to provide the goals of preserving the integrity, availability and confidentiality of the information system resources and services. Several social aspects today have critical information that depends in networks for data transmissions, and the secure for this information is a big challenge [4],[61].

Ensuing security systems have many challenges like systems hackers. We can notice that the number of hackers is increasing as a result of the internet and information technology progress with many attacks like dictionary attacks and collision attacks. Another challenge is that the value of information becomes very sensitive, and we need to protect this information in storing or transmission between the senders and the receivers. One-way Hash functions are important cryptographic primitive, and can be used to solve many problems involving authentication and integrity. As a result, we are best motivated by these challenges in order to develop a new practical one way hash algorithm based on Hill Cipher [4].

1.3 Problem statement

Hill Cipher use invertible matrix in encryption and decryption .If the inverse of this matrix is not found then the encrypted text cannot be decrypted. Hash function is a well-defined procedure that converts a large data into a small one. The returned value from hash function is called hash code. One-way hash function is a function that converts a variable string length into a fixed length binary sequence that cannot be reversed. Our goal is to develop a new one way hash algorithm [22], [24], [25][10],[12],[13].

1.4 Proposed Solutions

It is well known that Hash algorithms work in one way and cannot be reversed. From this point, we need to choose the non-invertible matrix to use it inside a practical one-way hash algorithm. First we take non-invertible matrix and multiply it by plaintext as column vector with modular value n to generate the hash value H. Also if the used matrix is invertible, a conversion algorithm will be used to produce a non-invertible matrix. The sender of the message calculates the hash value or message digest by using the model. After that, message and hash value are sent to the receiver who make same calculation to the message by using the model to generate a

message hash value. Then the receiver compares between the message digest from the sender and hash value that he calculated.

1.5 Research Methodology

In this research, the original Hill cipher models will be analyzed, and the problems of the original Hill cipher will be pointed out. A new mathematical model will be formulated. At first we will build an algorithm to check whether the matrix is invertible or not, and if the matrix is invertible, convert it to non-invertible matrix. After that the practical one way hash algorithm is developed. The proposed model will be tested and simulated by using Matlab tools, and then will be compared with MD5, SHA1 and SHA-512 from speed and security perspectives .

1.6 Objectives

- Study the one way hash algorithm requirements.
- Design an algorithm that can convert the matrix to non-invertible matrix.
- Design the practical one way hash algorithm based on matrix multiplications used in Hill Cipher.
- Prove that our algorithm satisfies the requirements of one way hash algorithms.
- Compare the proposed model with MD5 and Secure Hash Algorithm (SHA1, SHA512), using a high performance language for technical computing (Matlab), in order to test our technique.
- Study the security of the new algorithm against brute force attack, collision resistance and dictionary attack.

1.7 Contributions

In this thesis we can summarize many contributions:

1. First, derive the algorithm that convert the invertible matrix to non-invertible matrix.
2. Second, design a practical one way hash algorithm that cannot be reversed based on matrix multiplications.
3. Third, design the salt algorithm to solve dictionary attack problem.

During our work on this research, we published a three papers. In the first paper, we proposed a mathematical proof for our model which considered to be as an invention contribution from " Journal of computer applications" in the USA[61]. In the second paper, we proposed our model in Arabic language, and provided a comparison between our model with MD5, SHA1, and SHA-

512 in a "Journal of communication Arab society in computer science". The third paper was a survey about cryptography which was published in a "Journal of information and computer security" in Malaysia[62].*(see appendix c for more information about papers)*

1.8 Literature review

A lot of work has been directed towards a one way hash function and Hill Cipher:

In 1993, Yuliang Zheng [52], talked about one-way hashing algorithm called HAVAL which compresses a message of arbitrary length into a digest of different data size. In addition of that, HAVAL has a parameter that controls the number of passes. A message block (of 1024 bits) is processed. A message block can be processed more than one pass. By combining output length with pass, he can provide fifteen choices for a practical applications where different levels of security are required.

In 1999, Murray Eisenberg [53] described what Hill ciphers were and how they were broken. And he discussed the requisite notions and facts about modular arithmetic.

In 2000, Victor Shoup [54] presentd a new scheme for constructing universal one-way hash functions that hashed arbitrarily long messages out of universal one-way hash functions that hashed fixed-length messages. The new construction is extremely simple and is also very efficient.

In 2005, Johannes Mittmann [59] . On his paper he gave definitions of the basic terms of cryptographic hash functions, First, he discusses generic attacks that can be applied to arbitrary hash functions and gives a comparison of security criteria. Second, he describes design principles of iterated hash functions in general, and the Secure Hash Algorithm (SHA-1) in particular. Finally, he introduces message authentication codes and shows their construction from other cryptographic primitives.

In 2005, Kimmo jravnien [56] Hardware implementation aspects of the MD5 hash algorithm are discussed in his paper. A general architecture for MD5 is proposed and several implementations are presented.

In 2006 Danilo Gligoroski [55] proposed a new design principle for construction of iterated hash functions. Then he showed that to reach the cryptographic strength to find collisions, much lower number of iterations is necessary.

In August 2006, Ismail et al [22] tried to modify the Hill cipher by adjusting the encryption key matrix to form a different key for each encryption block.

In 2009, Farajallah tried to overcome hill cipher problem by converting all none prime determinant matrices into prime determinant matrix, then all cipher text can be decrypted again.

In 2010 Harshvardhan Tiwari [60] presented a review of cryptographic hash functions. He presented numerous definitions of hash functions, different types of hash functions such as block cipher based on hash function and dedicated hash function, and various applications of hash functions. He gave special emphasis on dedicated hash functions like MD5, SHA-1 and RIPEMD-160.

Yellapu Naveen Kumar and Bikkina Narendra [41] at National Institute of Technology-Rourkela, in 2008 wrote thesis on Hill Cipher using the technique that presented on his paper called "Novel Methods of Generating Self-Invertible Matrix for Hill Cipher Algorithm" but by making some changes that generate self-repetitive matrix instead of Self-Invertible Matrix, and he made a simulation for matrix of size 3×3.

In 2009, Saroj Kumar Panigrahy and others[42], made a paper in biometrics area, and applied advanced Hill cipher algorithm for hiding information in the images.

In the same year Ahmed S. Hadi and Ali H. Mahdi [43] presented the idea of combine error free and encryption at the same system, they built mathematical model to use Hill cipher algorithm in a deferent way, this is done by encoding the plaintext and encrypted it before sending, the encryption process is done by using both Hill cipher and permutation, while at the decryption side only Hill cipher is used.

Ramchandra S. Mangrulkar and Pallavi V. Chavan in May 2009 [44] at International Journal of Recent Trends in Engineering, wrote a paper that implemented Hill Cipher algorithm, to hide the plaintext behind the cover image at the encryption side, and then at the decryption side, they decrypted the received cover image to retrieve the original plaintext which is hidden behind the cover image .

Ahmed Y. Mahmoud and Alexander G. Chefranov [45] in International Conference on Security of Information and Networks, in 2009, published their paper that analyzes a new modification of Hill cipher algorithm, that generates a dynamic encryption and decryption key matrix by

exponentiation which can be made efficiently with help of eigenvalues. The security of their system is improved by use of large numbers of dynamic keys.

In the previous section, we provide a literature review for many hashing algorithms and hill cipher technique. Chu-Hsing Lin and his colleagues tried to enhance the security of Hill Cipher using one-way hash function. It was a very good idea, but still cannot overcome the none-invertible matrix problem [25], Victor Shoup [54] developed a new scheme for constructing universal one-way hash functions that hash arbitrarily long messages out of universal one-way hash functions that hash fixed-length messages. Many people directed to work in hashing algorithms and encryption techniques for their great importance in information security applications as we see in the previous works some researcher work on design a new algorithm or encryption technique another work on security improvements and performance optimization. This factor encouraged me to work in this area to design a practical one way hash algorithm.

1.9 Thesis Outlines

The organization of this thesis can be summarized as follows:

1. Chapter one includes a brief overview of network security, motivation of thesis, problem statement, proposed solution, research methodology, objectives of the thesis, and literature review of previous studies that were performed in Hill cipher and hash algorithms .
2. Chapter two gives an introduction to cryptography, cryptography goals and math, a brief history of cryptography, symmetric and asymmetric encryption models, Hill cipher algorithm, one way hash algorithms .
3. Chapter three introduces the new practical one way hash algorithm based on matrix multiplications and makes needed mathematical proves or justifications. Finally, give many examples of the system.
4. Chapter four presents a list of simulation results and comparison between the proposed model for practical one way hash algorithm, MD5, SHA1 and SHA512.
5. Chapter five presents a comprehensive conclusions and future work.

Chapter 2

2.1 Introduction to Cryptography

Cryptography used in the past to protect national securities, and to secure military information and diplomatic correspondence. However, the use was limited. Nowadays, the range of cryptography applications is expanding a lot in the modern areas after the development of communication means; Cryptography is essentially required to ensure that the data are protected against penetrations and espionage. Cryptography is also a powerful mean in securing e-commerce[26], [16], [19], [62], [63].

2.2 Cryptography Goals:

By using cryptography, many goals can be achieved:

1. Confidentiality: it is the most important goal, and its ensure that nobody can understand the received message except the one who has the decipher key [10].
2. Authentication: it is the process of proving the identity, and its assure that the communication entity is the one which is claimed to be [12].
3. Data Integrity: it ensures that the received message has not been altered in any way from its original form [13].
4. Non-Repudiation: it is a mechanism used to prove that the sender really sent this message.
5. Access Control: it is the process of preventing any unauthorized use of resources [14].

2.3 Basic Terminology of Cryptography

Computers are used by millions of people for many purposes, "such as banking, shopping, military, and student records, etc…. . Privacy is a critical issue in many of these applications. We need to make sure that any unauthorized parties cannot read or modify the messages "[63].

Cryptography is the transformation of readable and understandable data into a form of data which cannot be understood for the purpose of securing data. Cryptography refers exactly to the methodology of concealing the content of messages. The word cryptography comes from the Greek word "Kryptos", that means hidden, and "graphikos" which means writing [3].

As Farajalla said in his theses "The information that we need to hide, is called **plaintext** (P). It's the original text, it can be in form of characters, numerical data, executable programs, pictures, or

any other kinds of information. The plaintext for example is the first draft of the message in the sender side before encryption, or it is the text at the receiver after decryption.

The data that will be transmitted is called **cipher text** (C), it's a term refers to the string of a "meaningless" data, or an unclear text that nobody except the recipients must understand. It is exactly the data that will be transmitted through network. Many algorithms used to transform plaintext into cipher text [4].

Cipher is the algorithm which is used to transform plaintext into cipher text. This method is called encryption or enciphers (encode). In other words, it's a mechanism of converting readable and understandable data into "*meaningless*" data, and it is represented as follows:

$$C = E_{(K)}(P) \tag{2.1}$$

Where $E_{(K)}$ is the encryption algorithm using key k.

The opposite of cipher mechanism is called **decipher (decode)** that it is the algorithm which recovers the cipher text. This method is called decryption. In other words it's the mechanism of converting "*meaningless*" data into readable data.

$$P = D_{(K^{-1})}(C) \tag{2.2}$$

The Key is an input to the encryption algorithm, and this value must be independent from the plaintext. This input is used to transform the plaintext into cipher text, so different keys will yield different cipher text, in the decipher side; the inverse of the key will be used inside the algorithm instead of the key.

Computer security it's a generic term for a collection of tools designed to protect data from hackers, theft, corruption, or natural disaster, while allowing these data to be available to the users at the same time. One example of these tools the A-vast antivirus program [1].

"**Network security** refers to any activity designed to protect the usability, integrity, reliability, and safety of data during their transmission on a network. Network security deals with hardware and software. The activity can be one of the following, anti-virus and anti-spyware, firewall, Intrusion prevention systems, and Virtual Private Networks" [63].

The Measures or procedures which are used to protect data during their transmission over a collection of interconnected networks are called **Internet Security**; but **information security** is used to prevent and to detect attacks on information-based systems [2].

Cryptanalysis (code breaking) is the study of the principles and the methods of deciphering cipher text without knowing the key. Typically this includes finding and guessing the secrete key. It's a complex process involving statistical analysis, analytical reasoning, math tools and pattern-finding. The field of both cryptography and cryptanalysis is called **cryptology** [23], [59].

Symmetric encryption refers to the process of converting plaintext into cipher text at the sender side with the same key that will be used to retrieve plaintext from cipher text at the recipient side, while **asymmetric encryption** refers to the process of converting plaintext into cipher text at the sender with different key that will be used to retrieve plaintext from cipher text at the recipient" [63]as in figure 2.2 [59], [62].

Passive attacks mean that the attackers or the unauthorized parties just monitor the traffic, or the communication between the sender and the recipient, and they don't attempt to breach or shut down the service. This kind of attacks is very difficult to be discovered, since the unauthorized party doesn't leave any traces. On the other hand, **active attacks** mean that the attackers are actively attempt to cause harm to the network or data. The attacker is not just monitoring the traffic but is also attempting to breach or shut down a service [23], [59].

Authentication is the process of determining whether someone is the same person who to be, such as login and password in login pages, but authorization is the process to ensure that this person has the ability to do something [23], [9], [59].

Brute force is the attacker who is trying all possible keys that may be used in either decrypt or encrypt information [59].

2.4 Symmetric And Asymmetric Encryption

Encryption is the strongest and the safest way in securing data. Certainly, it is the most common one. Encryption systems are divided into two main types or forms, symmetric and asymmetric[59], [62].

"Symmetric encryption, which is also known as secret key or single key, means that the receiver decrypts the message with the same key, which the sender uses to encrypt the data. This system was the only system used earlier the discovering and developing the public key. In symmetric encryption, a safe way of data transfer must be used to move the secret key between the sender and the receiver"[63]. Figure 2.1 shows how the system works. Symmetric encryption occurs either by substitution or transposition technique, or by a mixture of both techniques. Substitution

maps each plaintext element into cipher text element, but transposition transposes the positions of plaintext elements[7], [2], [62].

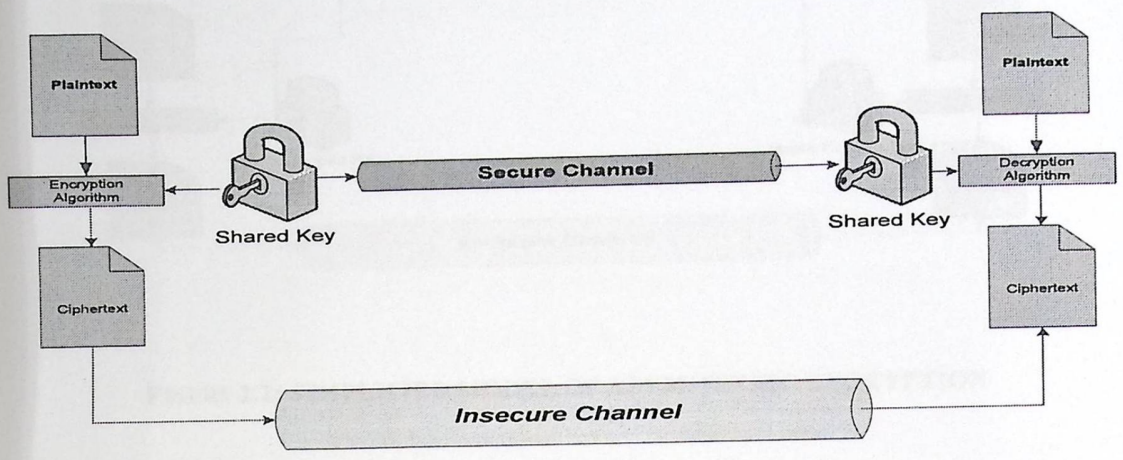


FIGURE 2.1: SIMPLIFIED MODEL OF CONVENTIONAL ENCRYPTION

The common simplified cipher algorithm which is called Caesar cipher, assigns each character of plaintext into numerical value, and sums the key value to the numerical value of plaintext character; and then assigns the rest of the division by modular value into cipher text character where the modular value is the max numerical value plus one [17]. The mathematical model of Caesar cipher is[11]:

At encryption side: $E_n(x) = (x + n) \bmod p$ (2.3)

At decryption side: $E_n(x) = (x - n) \bmod p$ (2.4)

Where x is a plaintext and n is a shift value.

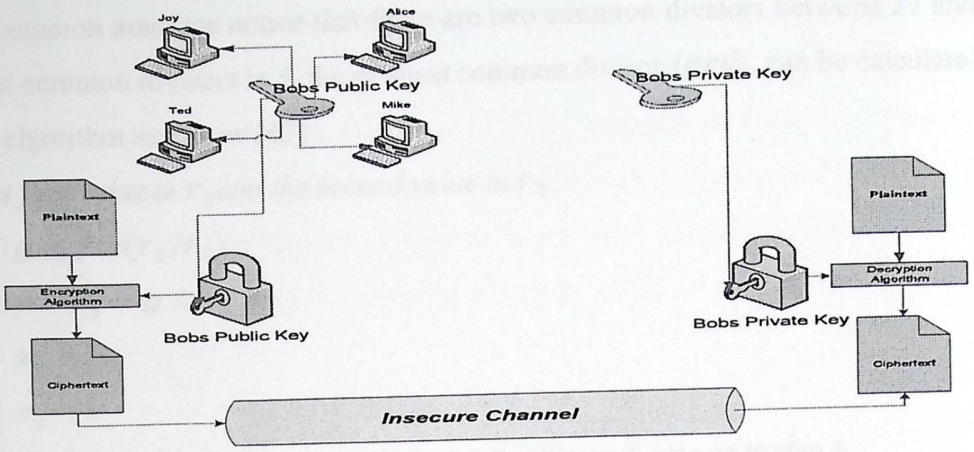


FIGURE 2.2: SIMPLIFIED MODEL OF ASYMMETRIC ENCRYPTION

2.5 Mathematics of Cryptography

Cryptography is based on mathematics. Many aspects of mathematics are used in cryptography such as number theory, modular arithmetic, linear algebra and matrices. These aspects play an important role in cryptography. In this section, the mathematics groups and operations which will be used in this research will be discussed, and the algorithms for these operations will be written as flowcharts [19], [63].

- **Integer Operations.**

In this part, I will discuss one of the integer operations, which is the greatest common divisor, this operation is needed in this research.

Any two positive integers have one or more common divisors, but we are interested in the greatest common divisor, for example the common divisors between ***120*** and ***27*** are listed in the following figure[63].

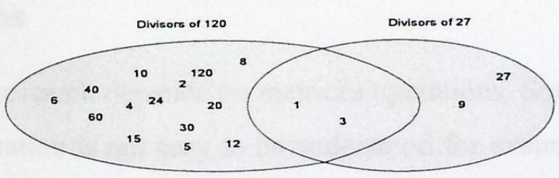


FIGURE 2.3: COMMON DIVISORS OF TWO INTEGERS

rom the common area, we notice that there are two common divisors between *27* and *120*, but he greatest common divisors is *3*, the greatest common divisor (***gcd***), can be calculate by using uclidean algorithm as follow [63]:

Set first value in r_1 *and the second value in* r_2.

1- $g = fix(r_1/r_2)$

2- $r = r_1 - g \times r_2$

3- $r_1 = r_2$

4- $r_2 = r$

5- Check if $r_2 > 0$, *then repeat from step* **2** *until step* **5**, *else go to step* **6**

6- $gcd(first\ value, second\ value) = r_1$

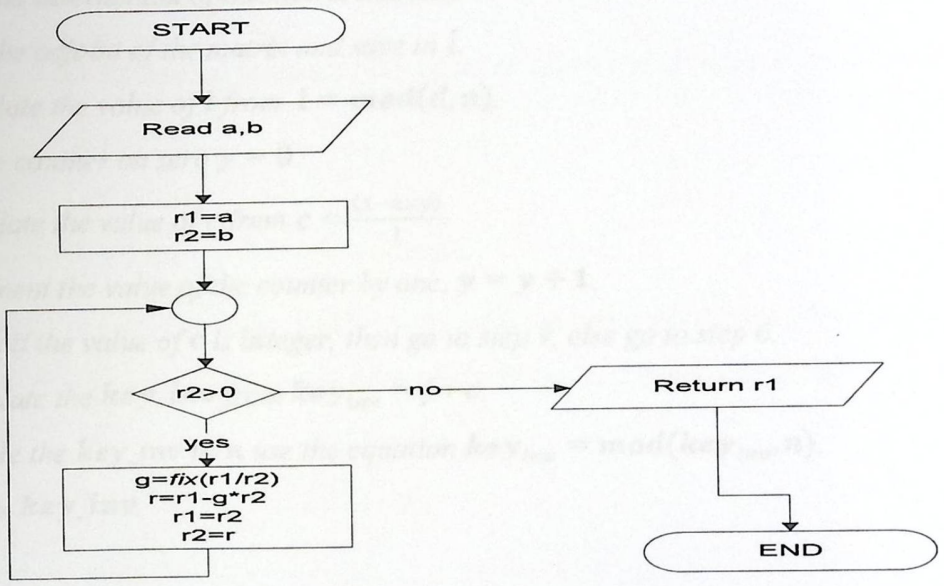


FIGURE 2.4: FLOWCHART OF FINDING GREATEST COMMON DIVISOR

- **Matrix Operations**

Most of the work on this research depends on matrices operations. Some of these operations are familiar for us. Other operation is not easy to be understood for example, when we need to find the inverse of a matrix relative to the specified modular value [40].

- **Modular Arithmetic**

In mathematics, modular arithmetic is a subsystem of arithmetic operations; it is only for integers, where the integers wrap around after they reach a certain value called the modulus.

Modular arithmetic was invented by Carl Friedrich Gauss in his published book "Disquisitiones Arithmeticae", in 1801, in other words; when two integers is divided, the result may contain the reminder, this reminder is called the residue, and if for example we have two integers a and b, and divide a by b, then if a is positive, the reminder is the result of module a by b, but if a is negative integer, then $(b - reminder)$ is the result of a module b, where b is called the modulus and must be positive integer, and a any integer [20].

Example2.1:

11 mod 05=01 -11 mod 05=04

13 mod 17=13 -13 mod 17=04

Algorithm steps to find the inverse of matrix relative to modular value[57], [63]:

1- *Receive the key matrix* k *and the modular value* n.

2- *Find the determinant of the matrix and save in* d.

3- *Find the adjoint of the matrix and save in* j.

4- *Calculate the value of* l *from* $l = mod(d, n)$.

5- *Set the counter on zero* $y = 0$.

6- *Calculate the value of* c *from* $c = \frac{(1-n\times y)}{l}$.

7- *Increment the value of the counter by one,* $y = y + 1$.

8- *Check if the value of* c *is* ***integer****, then go to step 9, else go to step 6.*

9- *Calculate the* key_inv *from* $key_{inv} = j * c$.

10- *Module the key_inv by n use the equation* $key_{inv} = mod(key_{inv}, n)$.

11- *Return* key_inv.

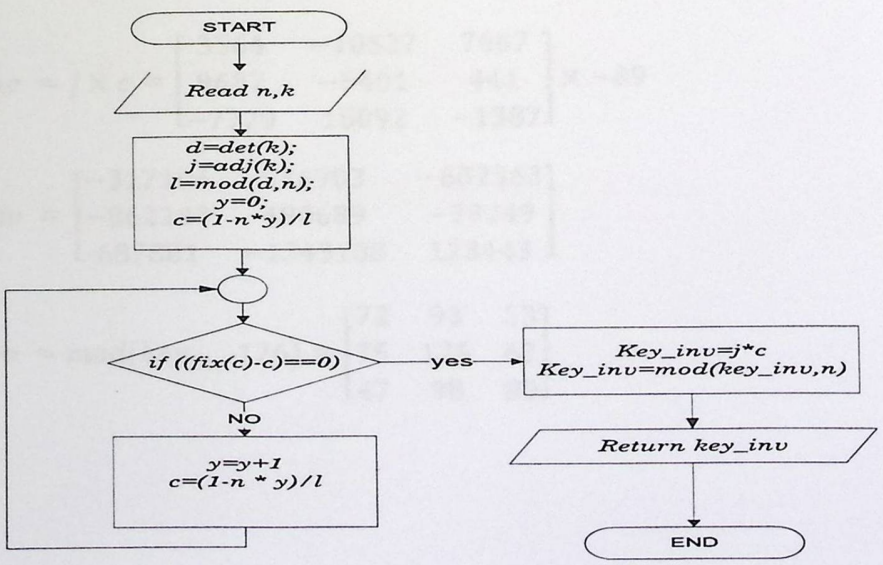


FIGURE 2.5: FLOWCHART OF FINDING INVERSE OF MATRIX RELATIVE TO MODULAR VALUE

Example2.2:

Find the inverse of $A = \begin{bmatrix} 1 & 121 & 44 \\ 12 & 65 & 87 \\ 125 & 33 & 99 \end{bmatrix}$ with respect to **126**

$$d = det(k) = 835615$$

$$j = adj(k) = \begin{bmatrix} 3564 & -10527 & 7667 \\ 9687 & -5401 & 441 \\ -7729 & 15092 & -1387 \end{bmatrix}$$

$$l = mod(835615,126) = 109$$

$$y = 0$$

$$c = \frac{(1 - n \times y)}{l} = \frac{(1 - 126 \times 0)}{109} = -0.0092$$

Notice that c value is not integer, so increment the value of counter y by one and recalculate c value.

$$y = y + 1 = 1$$

$$c = \frac{(1 - n \times y)}{l} = \frac{(1 - 126 \times 1)}{109} = -1.1468$$

Also, the value of c is not integer, so we need to increment the value of counter y by one and recalculate c value, finally after *77* tries, the value of c will become eventually integer.

$$y = y + 1 = 77$$

$$c = \frac{(1 - n \times y)}{l} = \frac{(1 - 126 \times 77)}{109} = -89$$

$$key\_inv = j \times c = \begin{bmatrix} 3564 & -10527 & 7667 \\ 9687 & -5401 & 441 \\ -7729 & 15092 & -1387 \end{bmatrix} \times -89$$

$$key\_inv = \begin{bmatrix} -317196 & 936903 & -682363 \\ -862143 & 480689 & -39249 \\ 687881 & -1343188 & 123443 \end{bmatrix}$$

$$key\_inv = mod(key_{inv}, 126) = \begin{bmatrix} 72 & 93 & 53 \\ 75 & 125 & 63 \\ 47 & 98 & 89 \end{bmatrix}$$

2.6 Hill Cipher

Hill cipher, invented by Lester S. Hill in 1929, uses matrix multiplication for mixing the plaintext [27]. It works by displaying a group of letters as a vector and encryption is done by matrix multiplication [26].

Hill cipher satisfies properties that a good cryptosystems would have:

-***Diffusion*:** one change in plaintext character should affect as many characters as possible in cipher text. We know that Hill cipher convert any plaintext characters to a number then inserts it in a matrix of column vector. If we take – *be* - as plaintext characters then it will be - 14 -, the matrix of column vector will be $\begin{bmatrix}1\\4\end{bmatrix}$, so that any change in plaintext must affect cipher text characters [36], [37], [61].

-***Confusion*:** The key should not relate to the cipher text, hill cipher use key matrix to encrypt the message and key inverse to decrypt it [38], [61].

Hill cipher example with Key Matrix 2×2 *use math equation: The key matrix must be invertible with modular value 26* [61] .

Given Plaintext: $p1p2p3p4.....pn-1pn$ *Given Key Matrix*: $k=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}$

Encryption:

1. *Form vectors as follows:*

$$\begin{pmatrix}p_1\\p_2\end{pmatrix}\begin{pmatrix}p_3\\p_4\end{pmatrix}\begin{pmatrix}p_5\\p_6\end{pmatrix}\cdots\begin{pmatrix}p_{n-1}\\p_n\end{pmatrix} \qquad (2.5)$$

2. *Multiply each vector by k to obtain a pair of cipher text letters:*

$$k\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} (\text{mod } 26)$$

$$k\begin{pmatrix} p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} c_3 \\ c_4 \end{pmatrix} (\text{mod } 26)$$

.
.
.

$$k\begin{pmatrix} p_{n-1} \\ p_n \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} p_{n-1} \\ p_n \end{pmatrix} = \begin{pmatrix} c_{n-1} \\ c_n \end{pmatrix} (\text{mod } 26) \qquad (2.6)$$

3. The cipher text message is: c1c2.....cn

Decryption:

1. Calculate k^{-1}

2. For each pair of cipher text find a plaintext by:

$$k^{-1}\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} (\text{mod } 26)$$

$$k^{-1}\begin{pmatrix} c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} p_3 \\ p_4 \end{pmatrix} (\text{mod } 26)$$

.
.
.
.

$$k^{-1}\begin{pmatrix} c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} p_{n-1} \\ p_n \end{pmatrix} (\text{mod } 26) \qquad (2.7)$$

Example 2.3: Let $n = 126$ and the key matrix that use $\boldsymbol{k} = \begin{bmatrix} 1 & 121 & 44 \\ 12 & 65 & 87 \\ 125 & 33 & 99 \end{bmatrix}$ and part of plaintext is $\boldsymbol{p} = \begin{bmatrix} 21 \\ 17 \\ 87 \end{bmatrix}$, note that the key is chosen so that the determinant is prime relative to **126**, this happens since determinant of key is **835615** and $\boldsymbol{\gcd(835615, 126) = 1}$.

Using encryption equation:

$$c = \begin{bmatrix} 1 & 121 & 44 \\ 12 & 65 & 87 \\ 125 & 33 & 99 \end{bmatrix} \times \begin{bmatrix} 21 \\ 17 \\ 87 \end{bmatrix} mod\ 126 = \begin{bmatrix} 5906 \\ 8926 \\ 11799 \end{bmatrix} mod\ 126 = \begin{bmatrix} 110 \\ 106 \\ 81 \end{bmatrix}$$

This vector is called the cipher text and is transferred across the network, from sender to the receiver, and after that, at decryption side K^{-1} is found by applying row echelon form as follow[63].

$$\left[\begin{array}{ccc|ccc} 1 & 121 & 44 & 1 & 0 & 0 \\ 12 & 65 & 87 & 0 & 1 & 0 \\ 125 & 33 & 99 & 0 & 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & 121 & 44 & 1 & 0 & 0 \\ 0 & -1387 & -441 & -12 & 1 & 0 \\ 0 & -15092 & -5401 & -125 & 0 & 1 \end{array}\right] mod\ 126$$

$$= \left[\begin{array}{ccc|ccc} 1 & 121 & 44 & 1 & 0 & 0 \\ 0 & 125 & 63 & 114 & 1 & 0 \\ 0 & 28 & 17 & 1 & 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & 121 & 44 & 1 & 0 & 0 \\ 0 & 15625 & 7875 & 14250 & 125 & 0 \\ 0 & 28 & 17 & 1 & 0 & 1 \end{array}\right] mod\ 126$$

$$= \left[\begin{array}{ccc|ccc} 1 & 121 & 44 & 1 & 0 & 0 \\ 0 & 1 & 63 & 12 & 125 & 0 \\ 0 & 28 & 17 & 1 & 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & 0 & -7579 & -1451 & -15125 & 0 \\ 0 & 1 & 63 & 12 & 125 & 0 \\ 0 & 0 & -1747 & -335 & -3500 & 1 \end{array}\right] mod\ 126$$

$$= \left[\begin{array}{ccc|ccc} 1 & 0 & 107 & 61 & 121 & 0 \\ 0 & 1 & 63 & 12 & 125 & 0 \\ 0 & 0 & 17 & 43 & 28 & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & 0 & 107 & 61 & 121 & 0 \\ 0 & 1 & 63 & 12 & 125 & 0 \\ 0 & 0 & 1513 & 3827 & 2492 & 89 \end{array}\right] mod\ 126$$

$$\begin{bmatrix} 61 & 121 & 0 \\ 12 & 125 & 0 \\ 47 & 98 & 89 \end{bmatrix}$$

$$\begin{bmatrix} -4968 & -6-10365 & -9523 \\ -2949 & -6049 & -5607 \\ 47 & 98 & 89 \end{bmatrix} mod\ 126$$

$$\begin{bmatrix} 0 & 72 & 93 & 53 \\ 0 & 75 & 125 & 63 \\ 1 & 47 & 98 & 89 \end{bmatrix}$$

$$= \begin{bmatrix} 72 & 93 & 53 \\ 75 & 125 & 63 \\ 47 & 98 & 89 \end{bmatrix}$$

use this key at the decryption side to find the original plaintext as follow:

$$\begin{bmatrix} 72 & 93 & 53 \\ 75 & 125 & 63 \\ 47 & 98 & 89 \end{bmatrix} \times \begin{bmatrix} 110 \\ 106 \\ 81 \end{bmatrix} mod\ 126 = \begin{bmatrix} 22071 \\ 26603 \\ 22767 \end{bmatrix} = \begin{bmatrix} 21 \\ 17 \\ 87 \end{bmatrix}$$

his is the original vector that is sent from the sender.

2.7 Hash Algorithms

Definition, Descriptions and Applications

Algorithm changes messages or texts into a fixed string of digits usually for systems security integrity, confidentiality and availability. The one way means that it's hard to recover the original text from the hash value string. A one-way hash function is used to create digital signatures which in its turn identify and authenticate the sender and the message [55], [56].

An important element in many computer security services and applications is the use of cryptographic algorithms .The first type is symmetric encryption like DES algorithm, which is used in the widest variety of contexts, primarily to provide confidentiality, another type is a secure hash functions like SHA-512, MD5 that used in message authentication. The third type is public-key encryption like RSA. Asymmetric encryption and secure hash functions are combined to produce an extremely useful tool; Hash functions are used in cryptography with digital signatures and for ensuring data integrity when used with digital signatures, a publicly available

$$= \left[\begin{array}{cccccc} 1 & 0 & 107 & 61 & 121 & 0 \\ 0 & 1 & 63 & 12 & 125 & 0 \\ 0 & 0 & 1 & 47 & 98 & 89 \end{array}\right]$$

$$= \left[\begin{array}{ccccc} 1 & 0 & 0 & -4968 & -6-10365 & -9523 \\ 0 & 1 & 0 & -2949 & -6049 & -5607 \\ 0 & 0 & 1 & 47 & 98 & 89 \end{array}\right] mod\ 126$$

$$= \left[\begin{array}{cccccc} 1 & 0 & 0 & 72 & 93 & 53 \\ 0 & 1 & 0 & 75 & 125 & 63 \\ 0 & 0 & 1 & 47 & 98 & 89 \end{array}\right]$$

$$\therefore\ k^{-1} = \left[\begin{array}{ccc} 72 & 93 & 53 \\ 75 & 125 & 63 \\ 47 & 98 & 89 \end{array}\right]$$

Now use this key at the decryption side to find the original plaintext as follow:

$$p = \left[\begin{array}{ccc} 72 & 93 & 53 \\ 75 & 125 & 63 \\ 47 & 98 & 89 \end{array}\right] \times \left[\begin{array}{c} 110 \\ 106 \\ 81 \end{array}\right] mod\ 126 = \left[\begin{array}{c} 22071 \\ 26603 \\ 22767 \end{array}\right] = \left[\begin{array}{c} 21 \\ 17 \\ 87 \end{array}\right]$$

This is the original vector that is sent from the sender.

2.7 Hash Algorithms

- ***Definition, Descriptions and Applications***

Algorithm changes messages or texts into a fixed string of digits usually for systems security integrity, confidentiality and availability. The one way means that it's hard to recover the original text from the hash value string. A one-way hash function is used to create digital signatures which in its turn identify and authenticate the sender and the message [55], [56].

An important element in many computer security services and applications is the use of cryptographic algorithms .The first type is symmetric encryption like DES algorithm, which is used in the widest variety of contexts, primarily to provide confidentiality, another type is a secure hash functions like SHA-512, MD5 that used in message authentication. The third type is public-key encryption like RSA. Asymmetric encryption and secure hash functions are combined to produce an extremely useful tool; Hash functions are used in cryptography with digital signatures and for ensuring data integrity when used with digital signatures, a publicly available

hash function hashes the message and signs the resulting hash value. The party who receives the message then hash the message and check if the block size is authentic for the given hash value [15], [4].

Figure 2.6 illustrates three ways in which the message can be authenticated." The message digest can be encrypted using conventional encryption (part a); if it is assumed that only the sender and receiver share the encryption key, then authenticity is assured. The message can also be encrypted using public-key encryption (part b). The public-key approach has two advantages: it provides a digital signature as well as message authentication; and it does not require the distribution of keys to communicating parties. These two approaches have an advantage over approaches that encrypt the entire message in that less computation is required. Nevertheless, there has been interest in developing a technique that avoids encryption altogether. Part c shows a technique that uses a hash function but no encryption for message authentication. This technique assumes that two communicating parties, say A and B, share a common secret value S_{AB}. When A has a message to send to B, it calculates the hash function over the concatenation of the secret value and the message: $MD_M = H(S_{AB}||M)$. It then sends $[M||MD_M]$ to B. Because B possesses S_{AB}, it can re-compute $H(S_{AB}||M)$ and verify MD_M. Because the secret value itself is not sent, it is not possible for an attacker to modify an intercepted message. As long as the secret value remains secret, it is also not possible for an attacker to generate a false message" [15], and the part c is what we need to make in our algorithm as we will discuss in section 3.4.

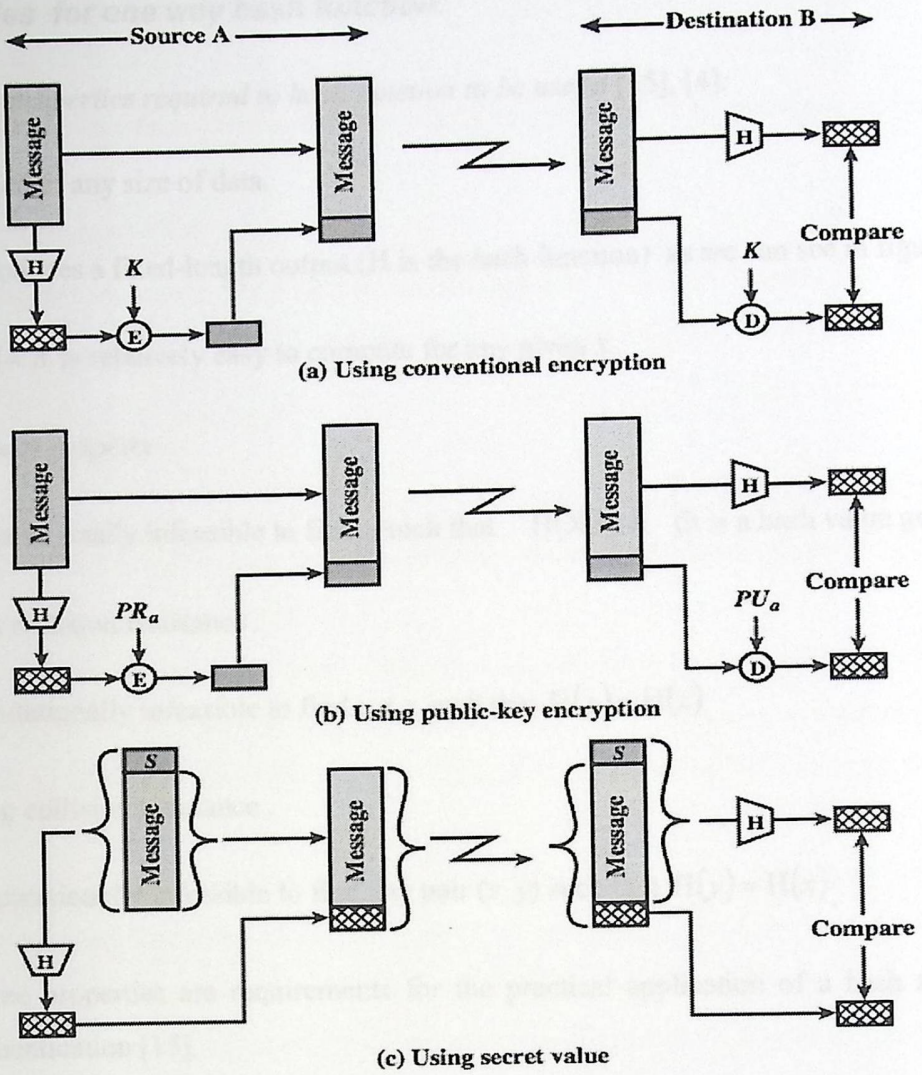


FIGURE 2.6: MESSAGE AUTHENTICATION[15]

One way hash function is an alternative to message authentication code (MAC) which accepts a variable size input and produces a fixed size message. Uunlike the MAC hash code, which does not require a key as input to authenticate the message but a message digest is sent with a message in authenticated way. the message digest can be encrypted by using symmetric key if only sender and receiver share the key in this way the authenticity is satisfied or using public key encryption that does not require the keys to be distributed to the parties [15].

- ***Properties for one way hash function:***

The following properties required to hash function to be useful [15], [4]*:*

1. Applied to any size of data.
2. H Produces a fixed-length output.(H is the hash function) as we can see in figure 2.7.
3. $\mathrm{H}(\mathrm{X}) = h$ Is relatively easy to compute for any given x.
4. One-way property .

   Computationally infeasible to find x such that $\mathrm{H}(\mathrm{X}) = h$ (h is a hash value generated).
5. Weak collision resistance .

   Computationally infeasible to find $y \neq x$ such that $\mathrm{H}(y) = \mathrm{H}(x)$.
6. Strong collision resistance .

   Computationally infeasible to find any pair (x, y) such that $\mathrm{H}(y) = \mathrm{H}(x)$.

The first three properties are requirements for the practical application of a hash function to message authentication [15].

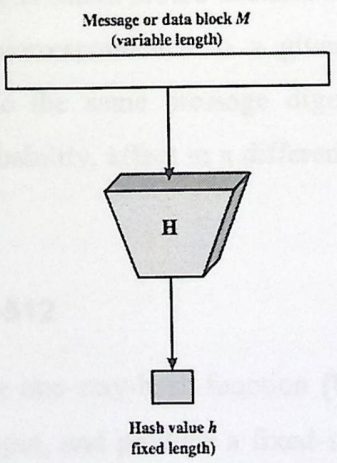


FIGURE 2.7: SECURE HASH FUNCTIONS[15].

2.8 Comparison Between MD5, SHA1

- **MD5**

MD5 is an enhanced version of md4. It's more complex, but its similar in design and produces a 128-bit hash.

- **Description of MD5**

After some initials processing, MD5 processes the input text in 512-bit blocks. The output of the algorithm is a set of four 32-bit blocks, which focused to form a single 128-bit hash value [15].

- **SHA1**

SHA-1 (Secure Hash Algorithm) is one of the most commonly used algorithm from SHA series of the cryptographic hash functions, it is designed by USA the National Security Agency o and published as their government standard [47].

SHA-1 produces the 160-bit hash value. Original SHA (or SHA-0) also produces 160-bit hash value, but SHA-0 was withdrawn by the NSA shortly after publication and was substituted by the revised version commonly referred to as SHA-1. The rest functions of SHA series produce 224-, 256-, 384- and 512-bit[47] .

This Standard specifies the Secure Hash Algorithm (SHA), which is very important to ensure the security of the Digital Signature Algorithm (DSA). When a message of any length < 264 bits is input, the SHA produces a 160-bit output called a message digest. The same message digest should be obtained by the verifier of the signature when the received version of the message is used as an input to SHA. The SHA is called secure because it is designed to be computationally infeasible to recover a message correspondence to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transmission, with a very high probability, affect in a different message digest, and the signature will fail to verify[47].

2.9 Secure Hash Algorithm-512

Hash Function and specifically the one-way-hash-function *(OWHF)* is a function that accepts a variable-size message M as an input, and produce a fixed-size output called hash code $H(M)$, digest or check value [4], [46]. Hash code algorithm doesn't use key, but instead of that it produces the output by using only the input message.

2.8 Comparison Between MD5, SHA1

- **MD5**

MD5 is an enhanced version of md4. It's more complex, but its similar in design and produces a 128-bit hash.

- **Description of MD5**

After some initials processing, MD5 processes the input text in 512-bit blocks. The output of the algorithm is a set of four 32-bit blocks, which focused to form a single 128-bit hash value [15].

- **SHA1**

SHA-1 (Secure Hash Algorithm) is one of the most commonly used algorithm from SHA series of the cryptographic hash functions, it is designed by USA the National Security Agency o and published as their government standard [47].

SHA-1 produces the 160-bit hash value. Original SHA (or SHA-0) also produces 160-bit hash value, but SHA-0 was withdrawn by the NSA shortly after publication and was substituted by the revised version commonly referred to as SHA-1. The rest functions of SHA series produce 224-, 256-, 384- and 512-bit[47] .

This Standard specifies the Secure Hash Algorithm (SHA), which is very important to ensure the security of the Digital Signature Algorithm (DSA). When a message of any length < 264 bits is input, the SHA produces a 160-bit output called a message digest. The same message digest should be obtained by the verifier of the signature when the received version of the message is used as an input to SHA. The SHA is called secure because it is designed to be computationally infeasible to recover a message correspondence to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transmission, with a very high probability, affect in a different message digest, and the signature will fail to verify[47].

2.9 Secure Hash Algorithm-512

Hash Function and specifically the one-way-hash-function *(OWHF)* is a function that accepts a variable-size message M as an input, and produce a fixed-size output called hash code $H(M)$, digest or check value [4], [46]. Hash code algorithm doesn't use key, but instead of that it produces the output by using only the input message.

Generally a hash function H must have the following properties, or in other words requirements of hash function[4]:

1. H *can be applied to a block of data of any size.*
2. H *produces a fixed-length output.*
3. $H(M)$ *is relatively easy to compute for any given message* M*, making both hardware and software implementations practical.*
4. *For any given digest* h*, it is computationally infeasible to find* M *such that* $H(M) = h$*, this property that makes hash function a one-way function.*
5. *For any given block* M_1*, it is computationally infeasible to find another block* M_2*, where* $(M_2 \neq M_1)$*, such that* $H(M_1) = H(M_2)$*, this referred to be a weak collision resistance.*
6. *It is computationally infeasible to find any pair of blocks* (M_1, M_2) *such that* $H(M_1) = H(M_2)$*, this is referred to be strong collision resistance.*

From previous properties, it is concluded that hash function can't be used in encryption plaintext; since we need the inverse of hash function and decryption cipher text, this property that makes hash algorithm a one way function, this is clear from point four, but hash function can be used in many other application such as authentication[47].

SHA-512 was developed by the National Institute of Standards and Technology (NIST), SHA-512 can be described briefly in two stages: first one is preprocessing and hash computation. Preprocessing involves padding a message, parsing the padded message into $1024 - bit$ blocks, and setting initialization values to be used in the hash computation. The second stage generates a message schedule from the padded message and uses that schedule, along with functions, constants, and word operations to iteratively generate a series of hash values. The final hash value generated by the hash computation is used to determine the message digest, SHA-512 takes any message M of any size less than 2^{128} bits, at preprocessing stage the input message M shall be padded before hash computation begins. The purpose of this padding is to ensure that the padded message is a multiple of $1024 - bits$ (size of block), second step of preprocessing stage is parsing the message, the padded message replace unpadded message and save in M, the

padded message must be parsed into $N \times (128 - bit)$ where N number of blocks, third step on preprocessing stage set initial hash value $H^{(0)}$ [63].

These words were obtained by taking the first sixty-four bits of the fractional parts of the square roots of the first eight prime numbers.

SHA-512 uses sequence of eighty constant 64-bit words, these words represent the first sixty-four bits of the fractional parts of the cube roots of the first eighty prime numbers. In hex.

SHA-512 Processing computation stage, summarize in the following algorithm steps after first stage is completed.

For $i = 1$ *to* N:

{

1. Prepare the message schedule, $\{W_t\}$:

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{\{512\}}(W_{t-2}) + (W_{t-7}) + \sigma_0^{\{512\}}(W_{t-15}) + (W_{t-16}) & 16 \le t \le 79 \end{cases} \quad (2.8)$$

2. Initialize the eight working variables, a, b, c, d, e, f, g, and h, with the $(i-1)^{st}$ hash value:

$$= H_0^{(i-1)}$$
$$= H_1^{(i-1)}$$
$$= H_2^{(i-1)}$$
$$= H_3^{(i-1)}$$
$$= H_4^{(i-1)}$$
$$= H_5^{(i-1)}$$
$$= H_6^{(i-1)}$$
$$= H_7^{(i-1)}$$

3. *For* $t = 0$ *to* 79:

{

$$T_1 = h + \sum_1^{\{512\}}(e) + Ch(e,f,g) + K_t^{\{512\}} + W_t \quad (2.9)$$

$$T_2 = \sum_0^{\{512\}}(a) + Maj(a,b,c) \quad (2.10)$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = T_1 + d$$

$$d = c$$

$$c = b$$

$$b = a$$

$$a = T_1 + T_2$$

}

4. Compute the i^{th} intermediate hash value $H^{(i)}$:

$$H_0^{(i)} = a + H_0^{(i-1)} \quad (2.11)$$

$$H_1^{(i)} = b + H_1^{(i-1)} \quad (2.12)$$

$$H_2^{(i)} = c + H_2^{(i-1)} \quad (2.13)$$

$$H_3^{(i)} = d + H_3^{(i-1)} \quad (2.14)$$

$$H_4^{(i)} = e + H_4^{(i-1)} \quad (2.15)$$

$$H_5^{(i)} = f + H_5^{(i-1)} \quad (2.16)$$

$$H_6^{(i)} = g + H_6^{(i-1)} \quad (2.17)$$

$$H_7^{(i)} = h + H_7^{(i-1)} \quad (2.18)$$

}

After repeating all iteration of outer loop for all blocks, the resulting 512-bit message digest of the message, $\boldsymbol{M}$ is

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)} \quad (2.19)$$

This simple view of the algorithm of Secure Hash Algorithm-512; and for further details please refer to one of references on this function list in the thesis references list, just because explaining one example would require large number of pages[4], [15], [47].

Figure 2.8 depicts the overall processing of a message to produce a hash value using SHA-512 algorithm. "The processing consists of the following :

Step 1: Append padding bits: so that message length is congruent to 896 modulo 1024 [length ≡ 896 (mod 1024)]. The padding consists of a single 1-bit followed by the necessary number of 0-bits.

as a block of 128 bits being an unsigned 128-bit integer length of the (before padding).

hash buffer: to the specified 64-bit integer values.

the message in 1024-bit (128-word) blocks, which forms the core of the a module, labeled F in this figure, that consists of 80 rounds.

the final hash buffer value as the resulting hash"[15].

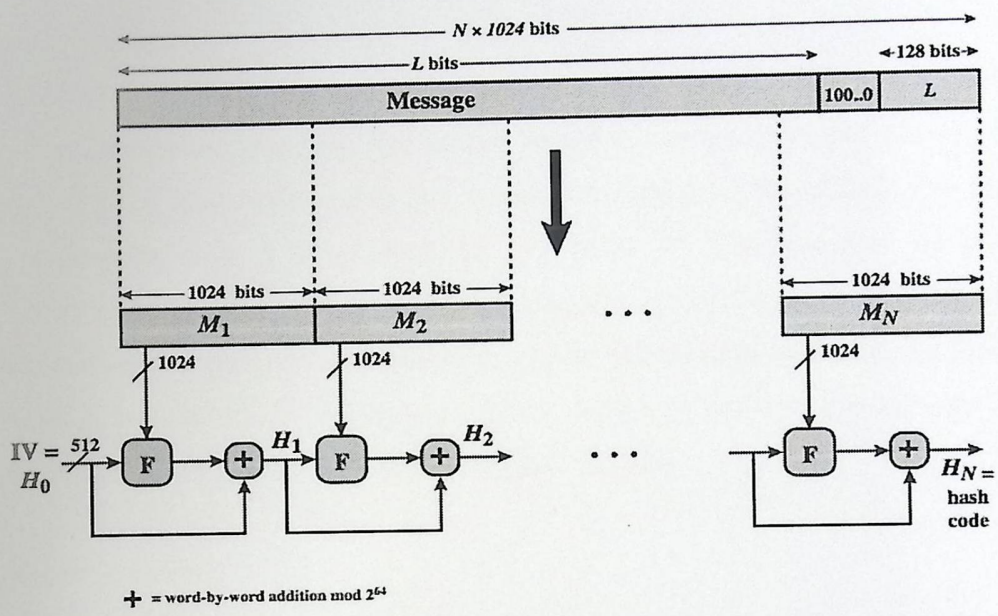


FIGURE 2.8: SHA-512 STRUCTURE[15].

Examples 2.4: The message is " *Palestine*" with *MD5* hash algorithm

The result: *b6d6ae0b8516440b56da14e3f2be5d95.*

Examples 2.5: The message is " *Palestine*" with *SHA1* hash algorithm

The result: *46eb8b3d433841e2ee1df540236ee0063aea551a.*

Examples 2.6: The message is " *Palestine"* with *SHA512* hash algorithm

The result:
1ecf229077df9a1ec9216013c20f66bc389039d1b0d3fd42bb9b4d00db1bc81cb45359c21d70a031 96c9b6f25eefb87162b650b610aa31d2dce7c65ffbfadf2d.

Examples 2.7 *The result of this message using* ***proposed algorithm*** *that will be described in chapter three will be:*

Step 2: Append length: as a block of 128 bits being an unsigned 128-bit integer length of the original message (before padding).

Step 3: Initialize hash buffer: to the specified 64-bit integer values.

Step 4: Process the message in 1024-bit (128-word) blocks, which forms the core of the algorithm, being a module, labeled F in this figure, that consists of 80 rounds.

Step 5: Output the final hash buffer value as the resulting hash"[15].

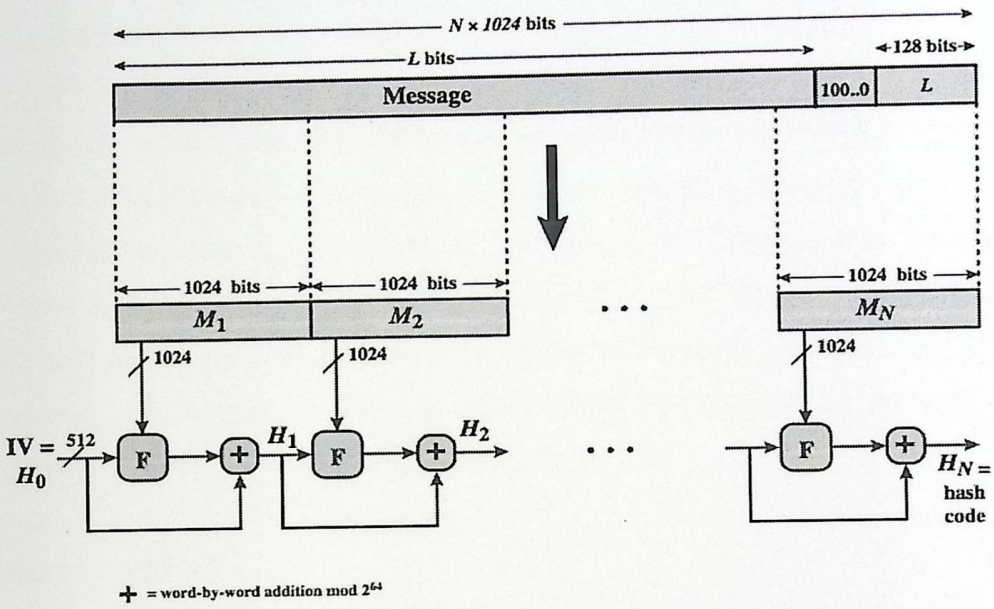


FIGURE 2.8: SHA-512 STRUCTURE[15].

Examples 2.4: The message is " *Palestine*" with *MD5* hash algorithm

The result: *b6d6ae0b8516440b56da14e3f2be5d95.*

Examples 2.5: The message is " *Palestine*" with *SHA1* hash algorithm

The result: *46eb8b3d433841e2ee1df540236ee0063aea551a.*

Examples 2.6: The message is " *Palestine"* with *SHA512* hash algorithm

The result:
1ecf229077df9a1ec9216013c20f66bc389039d1b0d3fd42bb9b4d00db1bc81cb45359c21d70a031 96c9b6f25eefb87162b650b610aa31d2dce7c65ffbfadf2d.

Examples 2.7 *The result of this message using* ***proposed algorithm*** *that will be described in chapter three will be:*

3f705d93020b14de334ff738a82bfec1 ***when matrix*** $R = \begin{bmatrix} 1 & 121 & 44 \\ 12 & 65 & 87 \\ 125 & 33 & 99 \end{bmatrix}$.

The final hash value that created using the (MD5, SHA1, and SHA-512) vary in length according to each algorithm steps and compression function in creating digest. In next chapter we will propose our algorithm POH.

Chapter 3

3.1 Introduction

In this chapter, we propose a new model to generate a hash value. We provide a math model and a proof requirement to satisfy the one way hash algorithm properties. In addition to that, we describe some examples about the algorithm.

3.2 Proposed Model

The main point of one-way hash function is that encrypted text cannot be decrypted. From this point, we need to choose the non-invertible matrix to use it inside a practical one-way hash algorithm. First we take the non-invertible matrix, and multiply it by the plaintext as column vector with the modular value n to generate the hash value H. The sender of the message calculates the hash value or digest of the message by using the model. Then the message and the hash value are sent to the receiver who makes the same calculations by using the model to generate a message hash value. After that, the receiver compares between the message digest from the sender and the hash value that he calculates[61], [9], [30].

3.3 Mathematical Model

Assume we have a plaintext message V represented as 1-dimentional matrix, a non-invertible key matrix R, and we have a modular value n as a secret value that is chosen between a sender and a receiver, then we can compute the hash value by using the following formula:

$H(V) = V \times R \bmod n$. Where : (3.1)

$H(V)$= generated hash value.

V =plaintext message as column vector.

R= non-invertible key matrix.(should between sender and receiver)

n= modular value. (should between sender and receiver)

We use the R as a non-invertible matrix that cannot be reversed to generate hash value by using this formula: $H(V) = [V][R] \bmod n$

It is trivial that the non-invertible matrix R cannot be reversed, if we calculate the determinant of this matrix d. If $d = |R|$ not relatively prime to n, R^{-1} will not exist and we cannot calculate the value of $(H(V))^{-1}$ [61]. Hamamreh, H. and Farajallah, M. in their research paper "*Design of a robust cryptosystem algorithm for non-invertible matrices based on Hill cipher*" have proved

that if the non-invertible matrix was used, the encrypted text will never be decrypted [57]. This is what we need to prove in our model to satisfy the one way property.

3.4 Algorithm Steps:

1. Convert the input text message into a 1-dimentional ASCII vector V.
2. Choose non-invertible matrix R and modular value n

   The algorithm needs a non-invertible matrix R, and modular value n. This value is chosen randomly between sender and receiver as secret value.
3. Make padding to a 1-dimentional ASCII vector V with non-invertible matrix R.
4. Make matrix multiplication to generate hash value H.
5. Make concatenation between the result of step 4 and the value V using Salt algorithm.
6. Add digest to generated H on step 5 to make the final Hash value that cannot be reversed.

It should be clear that if the key matrix R is invertible, then the security of our algorithm fails and the hash can be reversed. However we designed a new method to convert any invertible matrix into a non-invertible matrix as will be discussed in section 3.4.3.

3.4.1 Make Padding :

If the length of the message is more or less than the matrix size, we must make padding for the message as we see in the next algorithm in table 3.1:

TABLE 3.1: MODEL TO MAKE PADDING.

```
While  Mod (length of message And Matrix size) equal zero then
Next step
Else
New digit= previous digit+65
```

3.4.2 Digest Creation:

The result of step 5 is being an input to the last step to make digest. In this research we use the *RFC standard* [58] to create the message digest after we implement this model by using matlab and combine it with our algorithm.

The following five steps are performed to compute the digest of the message.

Step 1. Append Padding Bits.

The message is padded to ensure that its length in bits plus 64 is divisible by 512.

Step 2. Append Length.

A 64-bit binary representation of the original length of the message is concatenated to the result of step (1).(Least significant byte first).

Step 3. Initialize Buffer.

a four–word buffer (ABCD) used to compute the message digest.

Step 4. Process Message in 16-Word Blocks.

This is the core of the algorithm which includes four "rounds" of processing. By using

XOR, AND, OR and NOT operations respectively in each round .

Step 5. Output.

we first define four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word.

F(X,Y,Z) = XY v not(X) Z
G(X,Y,Z) = XZ v Y not(Z)
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X v not(Z))

In each bit position F acts as a conditional: if X then Y else Z. The function F could have been defined using + instead of v since XY and not(X)Z will never have 1's in the same bit position.) It is interesting to note that if the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z) will be independent and unbiased. The functions G, H, and I are similar to the function F, in that they act in "bitwise parallel" to produce their output from the bits of X, Y, and Z, in such a manner that if the corresponding bits of X, Y, and Z are independent and unbiased, then each bit of G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased. Note that the function H is the bit-wise "xor" or "parity" function of its inputs[58].

3.4.3 Model To Convert Invertible Matrix To Non- Invertible Matrix

The POH algorithm needs a non-invertible matrix R. One of our contributions in this research is design an algorithm that check if R invertible or not prime relative to modular value n, if R is invertible then the algorithm convert it to non –invertible (relatively not prime to modular value n. For example if determinant R=4 and n=12, in this case the matrix R is non-invertible; since it's not prime relative to modular value n).

The algorithm increases the value of diagonal until bound $=n^2$ (square of modular value n) as we can see in table3.2 .

TABLE 3.2: MODEL TO CONVERT INVERTIBLE MATRIX TO NON- INVERTIBLE MATRIX.

```
C_Matrix = Convert_To_Noninvertible( Invertible Matrix R ,Modular value n)

Begin{
      Diagonal= Extracted_Diag(Invertible Matrix R){
      // This will store the diagonal of invertible matrix in Diagonal
Array.
      Matrix_Diag= Diagonal
      }
      I=1
       while((gcd(mod(det(R),n),n)==1) &&(I>=0))
          {

            Diagonal(I,I)= Diagonal(I,I)+1
            If  Diagonal(I,I)= n * n
                  Diagonal(I,I)= Matrix_Diag(i)
                      I=I-1
                  }

           End
        Return
```

- As we see in equation (3.2), if the matrix R is inventible relative to n, the algorithm will convert it to non-invertible by using the algorithm "`Convert To Non-invertible(R,n)`" that receives R and converts it to non-invertible by increasing the value of the diagonal until another great common divisor rather 1 is found by using loop on this formula `(gcd(mod(det(R),n),n)==1)`,with bound function n^2.

$$R=\left\{\begin{array}{l}\left[\begin{array}{ccc} R(0,0) & \cdots\cdots & R(0,L-1) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & R(i,i) & \cdot \\ \cdot & & \cdot \\ & & \cdot\cdot \\ R(L-1,0) & \cdots\cdots & R(l-1,l-1) \end{array}\right] where R(i,i) < n^2, i > 0, \gcd(\text{mod}(\det(R),n) <> 1 \\ \\ R(i,i) = R(i,i)+1, where..\gcd(\text{mod}(\det(R),n),n) == 1 \\ \text{RESET R(i, i) , where. R(i, i)} = n^2 \end{array}\right. \quad (3.2)$$

- **Theorem:** *let A_{mm} be an mxm matrix and n be a natural number . Suppose that all the cofactors of the matrix $c_{11}, c_{22}, c_{33}, \ldots\ldots\ldots\ldots, c_{mm}$ Are not congruent $\not\equiv$ to 0 mod n. If $DetA = 1 \bmod n$, then $DetA^* \not\equiv 1 \bmod n$ where A^* is $a^*_{ij} = a_{ij}$ except $a^*_{mm} = a_{mm} + 1$*

- **Proof:**

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} & . & . & a_{1m} \\ a_{21} & a_{22} & a_{23} & . & . & a_{2m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{m1} & a_{m2} & a_{m3} & . & . & a_{mm} \end{bmatrix}$$

$$DetA = a_{m1}c_{m1} + a_{m2}c_{m2} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots + a_{mm}c_{mm} \quad (3.3)$$

$$A^*=\begin{bmatrix} a_{11} & a_{12} & a_{13} & . & . & a_{1m} \\ a_{21} & a_{22} & a_{23} & . & . & a_{2m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{m1} & a_{m2} & a_{m3} & . & . & a_{mm}+1 \end{bmatrix}$$

$$DetA^* = a_{m1}c^*_{m1} + a_{m2}c^*_{m2} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots + (a_{mm}+1)c^*_{mm} \quad (3.4)$$

$$DetA^* = a_{m1}c^*_{m1} + a_{m2}c^*_{m2} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots + a_{mm}c^*_{mm} + c^*_{mm} \quad (3.5)$$

But it is clear that $c^*_m = c_m \quad \forall i = 1, \ldots\ldots\ldots, m$
And so

$$DetA^* = DetA + c^*_{mm} \tag{3.6}$$

$$(DetA^*) \bmod n = (DetA + c^*_{mm}) \bmod n$$

$$\begin{aligned} &= DetA \bmod n + c^*_{mm} \bmod n \\ &= 1 + c^*_{mm} \bmod n \\ &\neq 1 \bmod n \ \ since \ \ c^*_{mm} \neq 0 \end{aligned} \tag{3.7}$$

3.4.4 Salt Algorithm

A **Salt** consists of a combination of data, creating one of the inputs to a one-way hash function. In a typical usage for password authentication, the salt is stored along with the output of the one-way hash function, sometimes along with the number of iterations to be used in generating the output Salt is closely related to the concept of nonce[15]. The benefit provided by using a salted password is preventing a lookup table that helped the dictionary attack against the stored values. In this research we designed an algorithm to make Salt value, and combined this algorithm with the POH algorithm to make it more secure against dictionary attack by making concatenation between the initial value V with the initial hash value.

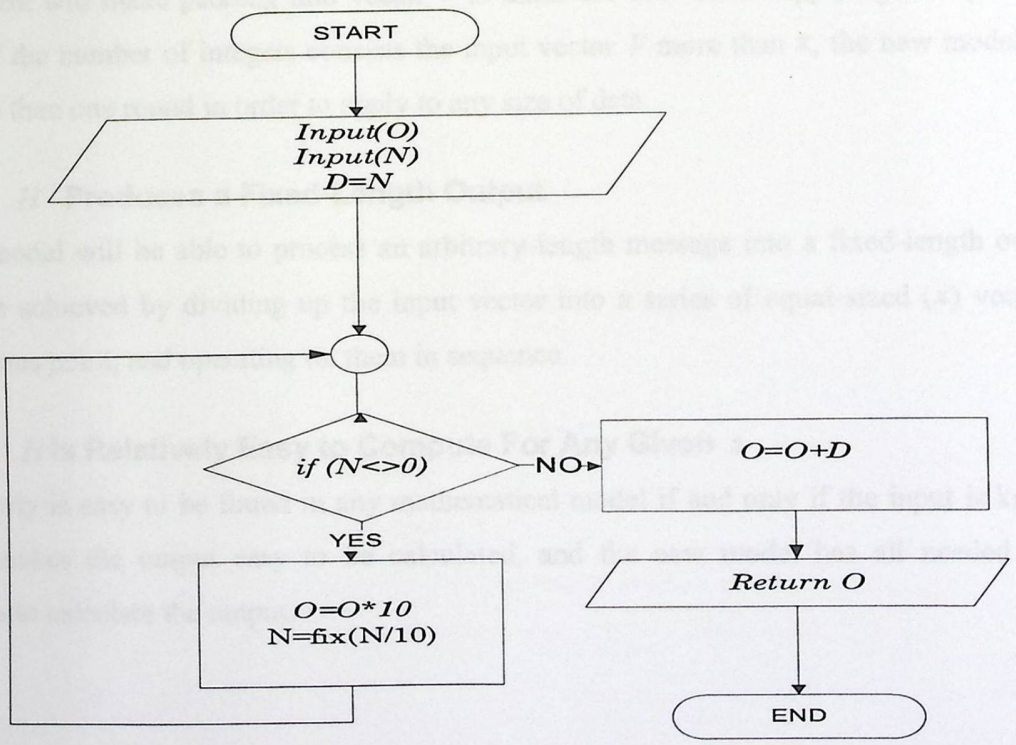


FIGURE 3.1: SALT ALGORITHM

In figure 3.1 we use the variable O to express one digit of the message V and the variable N to express one digit of initial hash value H. For example if the value of O=12 and the value of N=17, D=17. In first round of the algorithm N<>0 then O=120. In second round the result of $fix(N/10)$=1, N<>0, O will be 1200. In third round $fix(N/10)$=0, N=0 . The result of $O=O+D$ which is 1200+17=1217. The algorithm applied to all values of message V and initial hash H. This algorithm make the probability to get the matrix or message very hard for cryptanalysis.

3.5 Proof of Practical One Way properties For Hash Algorithm Requirements:

Any successful hash algorithm must satisfy some properties. In this section we provide a proof that our algorithm satisfies the one way hash algorithm requirements to generate a practical one way hash algorithm.

3.5.1 Applied To Any Size Of Data

For any input data V, let the square matrix has x dimension, then the system can convert the input data into vector(s) of x length. If the number of integers consists the input vector less than x, the system will make padding into vector v to make the new model applying to any size of data; but if the number of integers consists the input vector V more than x, the new model will make more than one round in order to apply to any size of data.

3.5.2 H Produces a Fixed-Length Output

The new model will be able to process an arbitrary-length message into a fixed-length output; this can be achieved by dividing up the input vector into a series of equal-sized (x) vector(s) from previous point, and operating on them in sequence.

3.5.3 H Is Relatively Easy to Compute For Any Given x

This property is easy to be found in any mathematical model if and only if the input is known, and that makes the output easy to be calculated, and the new model has all needed input parameters to calculate the output.

3.5.4 One-Way Property

The new model based on the following mathematical equation . $H(V) = V \times R \bmod n$, where V is the input vector at any round, R is the non-invertible matrix, and n is the modular value of the system, if any user has $H(v)$, R and n, he can only formulate the following model $V = H(v) \times R^{-1} \bmod n$. But since R is not invertible matrix, he can't solve this equation, so the proposed model is one way function.

3.6 POH Block Diagram

Figure 3.2 describes the block diagram of a practical one way hash algorithm . firstly, the algorithm reads the matrix R, modular value n, and the message V that we need to hash. Secondly, the algorithm converts the message to ASCII. Thirdly, the algorithm makes padding to make the size of the matrix and the message compatible . Fourthly, the algorithm checks whether the matrix is invertible or non- invertible. If the matrix is invertible, convert it to non- invertible. Fifthly, the algorithm computes the hash value by using matrix multiplications, and adds salt value and digest to the result of the last step to generate the final hash value.

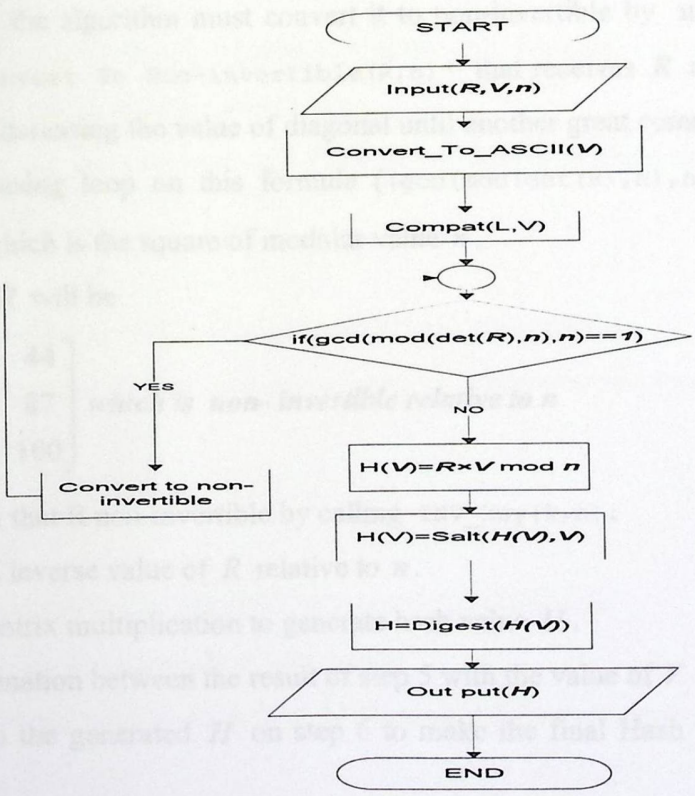


FIGURE 3.2: POH BLOCK DIAGRAM.

Example3.1: *we provide an example on the new algorithm* ***step by step****:*

If the message V that we want to hash is: ***"Palestine is our land from sea to river"*** *and the matrix* $R = \begin{bmatrix} 1 & 121 & 44 \\ 12 & 65 & 87 \\ 125 & 33 & 99 \end{bmatrix}$ ***with respect to n=12***

1. ***After calling*** `compat(l,V)`: the function receives the matrix size and V as string, then converts it to ASCII and makes padding to message to be compatible with size of matrix R, then builds V as column vector . The result will be :

Columns 1 through 14

80 97 108 101 115 116 105 110 101 32 105 115 32 111

Columns 15 through 28

117 114 32 108 97 110 100 32 102 114 111 109 32 115

Columns 29 through 39

101 97 32 116 111 32 114 105 118 101 114

2. Check weather matrix is non-invertible relative to n or not : if the matrix V is inventible relative to n, the algorithm must convert it to non-invertible by using the conversion algorithm "`Convert To Non-invertible(R,n)`" that receives R and converts it non-invertible by increasing the value of diagonal until another great common divisor rather 1 is found by using loop on this formula (`(gcd(mod(det(R),n),n)==1)`, with bound function n^2 which is the square of modular value n.
3. The value of R will be

$$R = \begin{bmatrix} 1 & 121 & 44 \\ 12 & 65 & 87 \\ 125 & 33 & 100 \end{bmatrix}$$ ***which is non- invertible relative to n***

4. We can check that R non-invertible by calling `INV_key(k,n)`:
   We can't find inverse value of R relative to n.
5. Then make matrix multiplication to generate hash value H.
6. Make concatenation between the result of step 5 with the value of V (Salt value).
7. Add digest to the generated H on step 6 to make the final Hash value that cannot be reversed.
8. The generated hash value = 4bfea7d02c738397040437f390ca192b.

Example 3.2: ***it is the*** *same as example 1 ,but it makes the matrix R 2X2*

If the message V that we want to hash it is: ***"Palestine is our land from sea to river"*** *and the*

matrix $\boldsymbol{R} = \begin{bmatrix} 1 & 121 \\ 12 & 65 \end{bmatrix}$ ***with respect to n=12***

- *The final hash value is:bdcee6f9b0af905d23870b03197e1885*

Example3.3: *same as example 2 with n=126*

If the message V that we want to hash is: ***"Palestine is our land from sea to river"*** *and the*

matrix $\boldsymbol{R} = \begin{bmatrix} 1 & 121 \\ 12 & 65 \end{bmatrix}$ ***with respect to n=126***

- *The final hash value is: 2c92dd33b82395af66adb5e76bfd6535*

Example3.4: ***it is the*** *same as example 3 with a change in one character of message* $\boldsymbol{P}$ *in Palestine to be* $\boldsymbol{F}$

If the message V that we want to hash is: ***"Falestine is our land from sea to river"*** *and the*

matrix $\boldsymbol{R} = \begin{bmatrix} 1 & 121 \\ 12 & 65 \end{bmatrix}$ ***with respect to n=126***

- *The final hash value is:* 913e1582a1965951f00953c7e126d0dd

Example3.5: ***it is the*** *same as example 4 with n=9*

If the message V that we want to hash is: ***"Falestine is our land from sea to river"*** *and the*

matrix $\boldsymbol{R} = \begin{bmatrix} 1 & 121 \\ 12 & 65 \end{bmatrix}$ ***with respect to n=9***

- *The final hash value is:* 298357505366ce3a570d1b2ab07db96e

Example3.6: *If the message V that we want to hash is:* ***"Palestine polytechnic university"*** *and*

the matrix $\boldsymbol{R} = \begin{bmatrix} 1 & 121 & 44 \\ 12 & 65 & 87 \\ 125 & 33 & 99 \end{bmatrix}$

with respect to n=109

- *The final hash value is:* 5e33420ff07247b844b0725580d9e16d

Example3.7: ***it is the same as example 6 with n=99***

If the message V that we want to hash it is: ***"Palestine polytechnic university"*** *and the matrix*

$$\boldsymbol{R} = \begin{bmatrix} 9 & 6 & 4 & 12 & 1 \\ 100 & 12 & 20 & 19 & 16 \\ 11 & 12 & 9 & 6 & 9 \\ 1 & 100 & 10 & 12 & 91 \\ 4 & 6 & 7 & 8 & 2 \end{bmatrix}$$

with respect to n=100

- *The final hash value is:* 332ad5860a1e2307207b86d07f8e92ca

Example3.8: it is the same as example 7 with n=26

*If the message V that we want to hash is: "**Palestine polytechnic university**" and the matrix*

$$R = \begin{matrix} 9 & 6 & 4 & 12 & 1 \\ 100 & 12 & 20 & 19 & 16 \\ 11 & 12 & 9 & 6 & 9 \\ 1 & 100 & 10 & 12 & 91 \\ 4 & 6 & 7 & 8 & 2 \end{matrix}$$

with respect to n=26

- *The final hash value is:* 8f09150bf9e483eca9b52551e1f4dac3

3.7 Summary

In this chapter, the practical one way hash algorithm which is the main contribution of our research with all algorithm steps is proposed and discussed. The mathematical model of POH algorithm is provided, a new model to convert invertible matrix to non-invertible matrix is provided which is a second contribution in our work, with a proof to all requirement needed to satisfy the properties of one way hash algorithm. The third contribution is Salt algorithm which used to solve the dictionary attack problem is described in next chapter. A list of examples are provided in the last section of this chapter.

Chapter 4

4.1 Simulation and Results Analysis:

In this chapter, the time per second of POH, MD5, SHA1and SHA-512 is measured for a different data size by using the Matlab simulation tool installed on dell inspiron with (I Core 5) central processing unit. The security level and the solution of dictionary attack will be discussed, and then the strength of our algorithm against brute force attack and collision is described.

- **Comparison Based on Matrix Size 1x1**

Using different data size, we measure the time per second for MD5, SHA1, SHA512, and our algorithm POH on matrix size 1x1. The matrix is $R = [12]$, with modular value = 128.

TABLE 4.1: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 1X1.

		Matrix Size 1x1			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*3.631*	*3.312*
	10KB	*38.563*	65.012	*11.679*	*11.121*
	256KB	*831.452*	1601.632	*221.829*	*214.441*
	1024KB	*3948.851*	6328.912	*1194.138*	*1188.127*

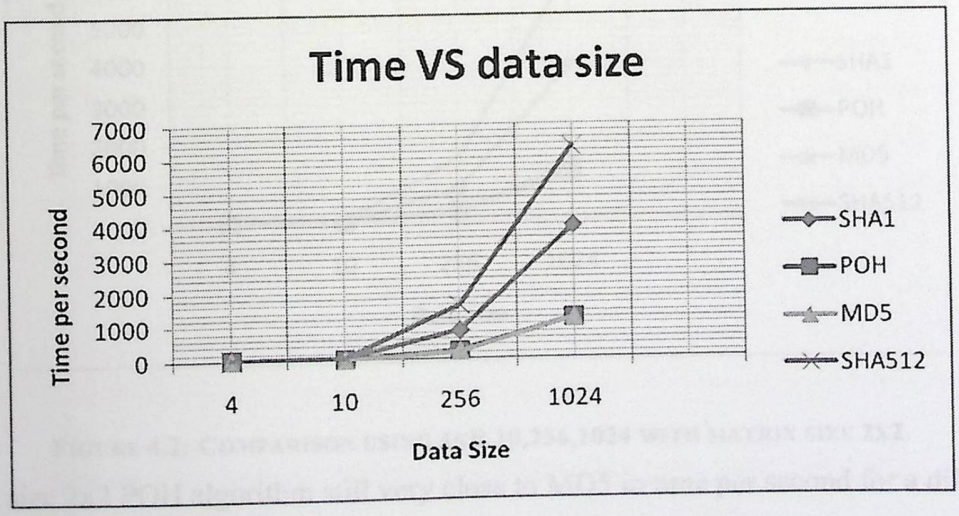


FIGURE 4.1: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 1X1.

As we see in figure 4.1, the performance of md5 and POH is almost the same for matrix size 1x1, and both MD5 and POH are faster than SHA1 and SHA512. MD5 and POH has four round in digest creation but SHA1 and SHA-512 has 80 round.

- **Comparison Based on Matrix Size 2x2**

Using different data size, we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 2x2. The matrix is $R = \begin{bmatrix} 12 & 10 \\ 2 & 12 \end{bmatrix}$, with modular value=128.

TABLE 4.2: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 2X2.

		Matrix Size 2x2			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*4.321*	*3.312*
	10KB	*38.563*	65.012	*12.839*	*11.121*
	256KB	*831.452*	1601.632	*277.1293*	*214.441*
	1024KB	*3948.851*	6328.912	*1314.714*	*1188.127*

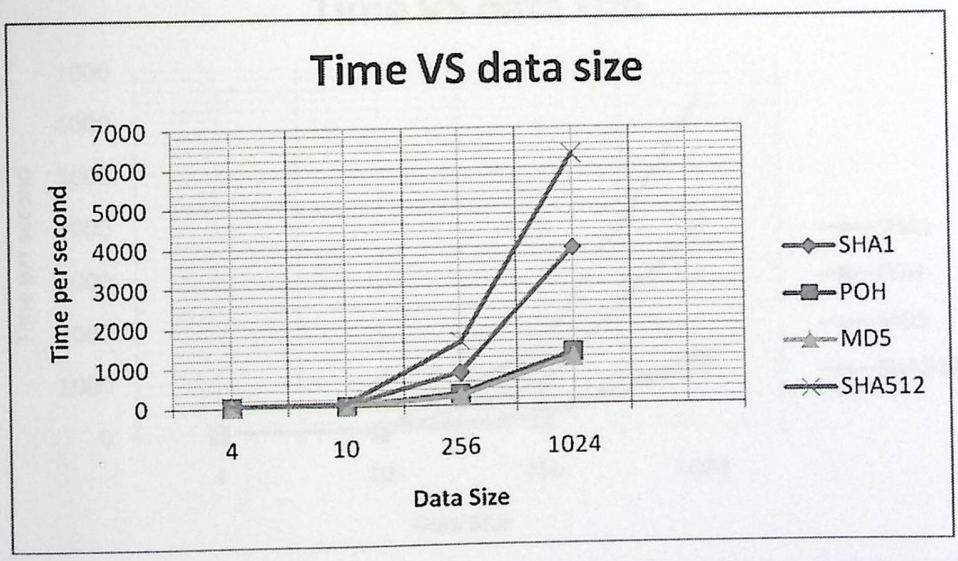


FIGURE 4.2: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 2X2.

On matrix size 2x2 POH algorithm still very close to MD5 in time per second for a different data size, and they are faster than SHA1 and SHA-512, because of the number of round that SHA1 and SHA-512 uses.

- **Comparison based on matrix size 3x3**

Using different data size we measure the time per second for MD5, SHA1, SHA512 and our algorithm POH on matrix size 3x3. The matrix is:

$$R = \begin{bmatrix} 12 & 14 & 16 \\ 18 & 20 & 22 \\ 24 & 26 & 28 \end{bmatrix}, \text{ with modular value= 128.}$$

TABLE 4.3: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 3X3.

		Matrix Size 3x3			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*4.477*	*3.312*
	10KB	*38.563*	65.012	*13.135*	*11.121*
	256KB	*831.452*	1601.632	*282.412*	*214.441*
	1024KB	*3948.851*	6328.912	*1345.024*	*1188.127*

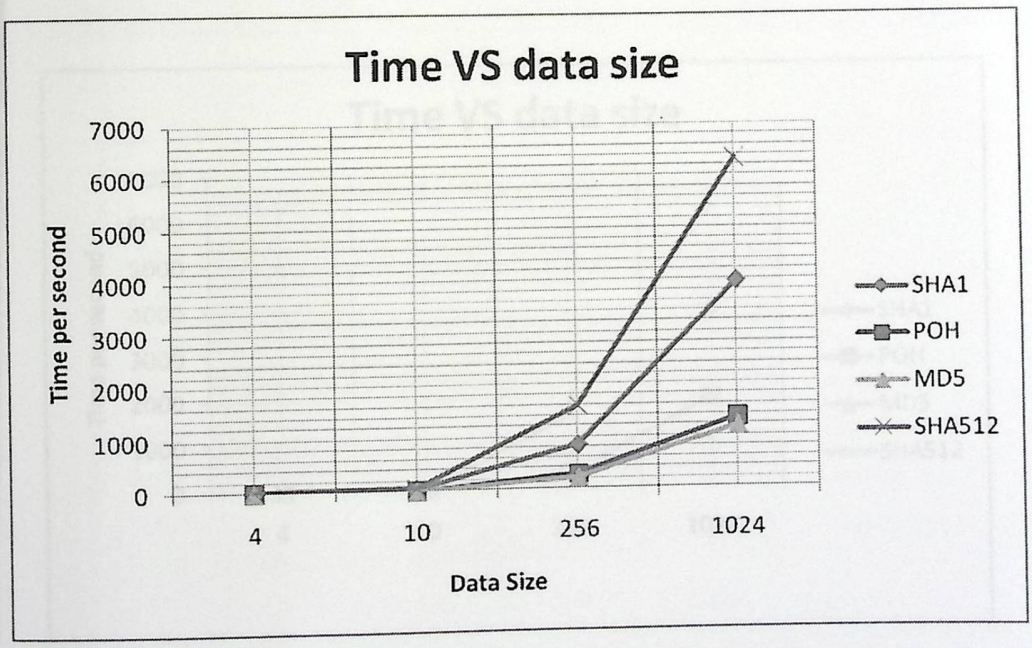


FIGURE 4.3: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 3X3.

On matrix size 3x3, POH algorithm still very close to MD5 in time per second on different data size and they are faster than SHA1 and SHA-512.

- **Comparison Based on Matrix Size 4x4**

Using different data size we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 4x4. The matrix is

$$R = \begin{bmatrix} 30 & 32 & 34 & 36 \\ 38 & 40 & 42 & 44 \\ 46 & 48 & 50 & 52 \\ 54 & 56 & 58 & 60 \end{bmatrix}$$, with modular value=128.

TABLE 4.4: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 4X4.

		Matrix Size 4x4			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*5.063*	*3.312*
	10KB	*38.563*	65.012	*19.212*	*11.121*
	256KB	*831.452*	1601.632	*312.101*	*214.441*
	1024KB	*3948.851*	6328.912	*1967.309*	*1188.127*

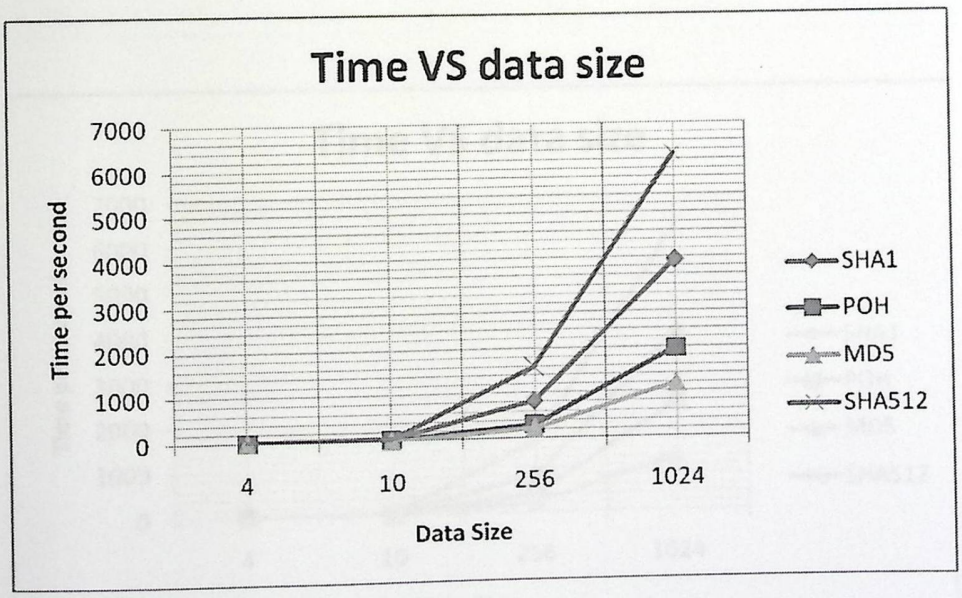


FIGURE 4.4: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 4X4.

On matrix size 4x4, the POH algorithm is slower than MD and still faster than SHA1 and SHA512.

Comparison Based on Matrix Size 5x5

Using different data size we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 5x5. The matrix is

$$R = \begin{bmatrix} 50 & 52 & 54 & 56 & 58 \\ 60 & 62 & 64 & 66 & 68 \\ 70 & 72 & 74 & 76 & 78 \\ 80 & 82 & 84 & 86 & 88 \\ 90 & 92 & 94 & 96 & 98 \end{bmatrix}$$, with modular value =128.

TABLE 4.5: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 5X5.

		Matrix Size 5x5			
		Hash algorithm			
		SHA1	SHA512	POH	MD5
Data Size	4KB	*12.964*	27.989	*5.616*	*3.312*
	10KB	*38.563*	65.012	*23.476*	*11.121*
	256KB	*831.452*	1601.632	*415.001*	*214.441*
	1024KB	*3948.851*	6328.912	*2403.942*	*1188.127*

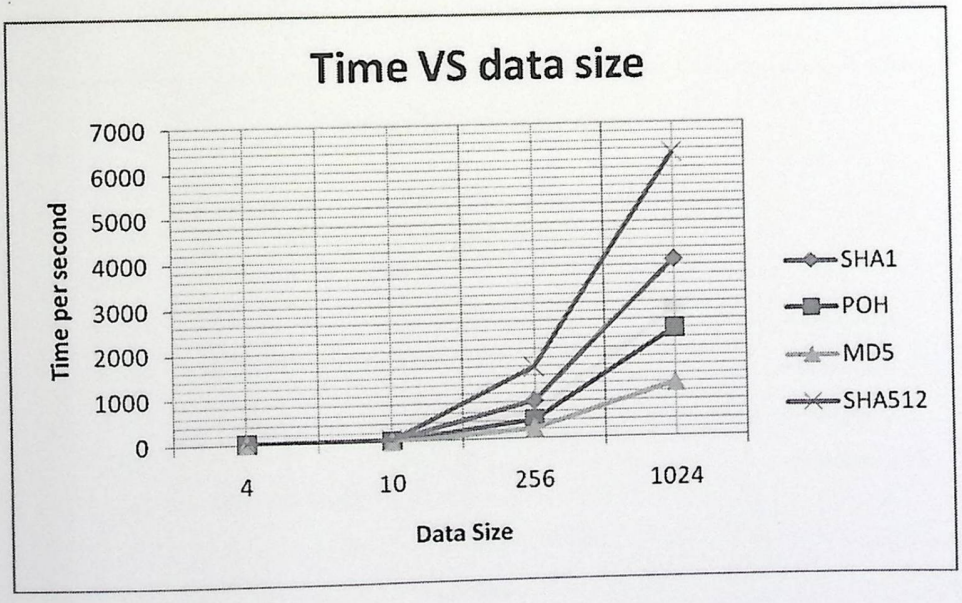


FIGURE 4.5: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 5X5.

On matrix size 5x5, the time per second of POH algorithm is larger than MD5, because the size of matrix increases.

- **Comparison Based on Matrix Size 6x6**

Using different data size we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 6x6.

The matrix is $R = \begin{bmatrix} 100 & 102 & 104 & 106 & 108 & 110 \\ 112 & 114 & 116 & 118 & 120 & 122 \\ 124 & 126 & 128 & 130 & 132 & 134 \\ 136 & 138 & 140 & 142 & 144 & 146 \\ 148 & 150 & 152 & 154 & 156 & 158 \\ 160 & 162 & 164 & 166 & 168 & 170 \end{bmatrix}$, with modular value=128.

TABLE 4.6 : TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 6X6.

		Matrix Size 6x6			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*6.0423*	*3.312*
	10KB	*38.563*	65.012	*27.162*	*11.121*
	256KB	*831.452*	1601.632	*509.397*	*214.441*
	1024KB	*3948.851*	6328.912	*2781.389*	*1188.127*

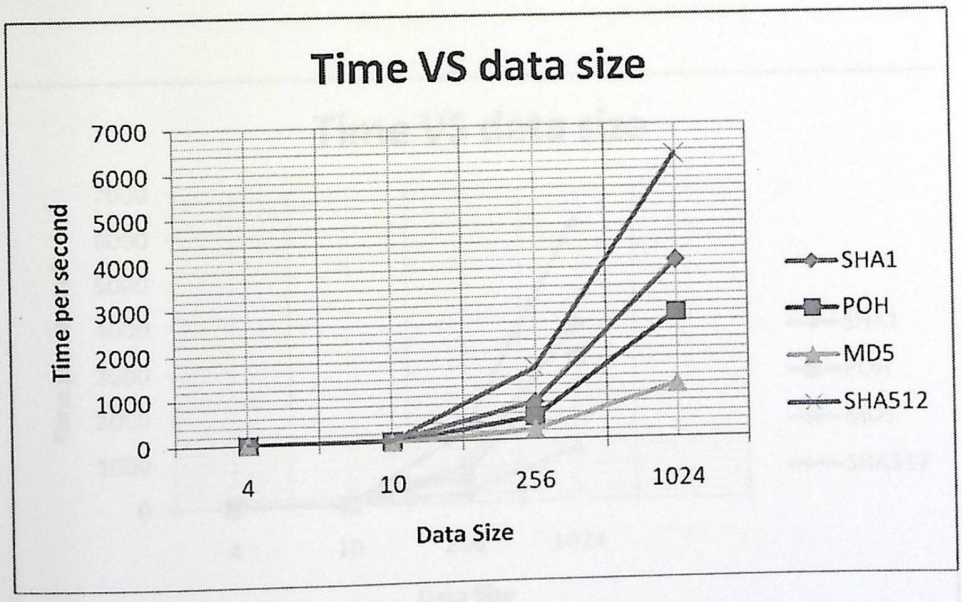


FIGURE 4.6: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 6X6

On the matrix size 6x6, the time per second of POH algorithm is larger than MD5, and smaller than SHA1, SHA-512 .

- **Comparison Based on Matrix Size 7x7**

Using different data size we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 7x7.

The matrix is $R = \begin{bmatrix} 166 & 168 & 170 & 172 & 174 & 176 & 178 \\ 180 & 182 & 184 & 186 & 188 & 190 & 192 \\ 194 & 196 & 198 & 200 & 202 & 204 & 206 \\ 208 & 210 & 212 & 214 & 216 & 218 & 220 \\ 222 & 224 & 226 & 228 & 230 & 232 & 234 \\ 236 & 238 & 240 & 242 & 244 & 260 & 262 \\ 264 & 265 & 267 & 269 & 271 & 273 & 275 \end{bmatrix}$, with modular value= 128.

TABLE 4.7: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 7X7.

		Matrix Size 7x7			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*7.223*	*3.312*
	10KB	*38.563*	65.012	*31.047*	*11.121*
	256KB	*831.452*	1601.632	*614.019*	*214.441*
	1024KB	*3948.851*	6328.912	*3179.213*	*1188.127*

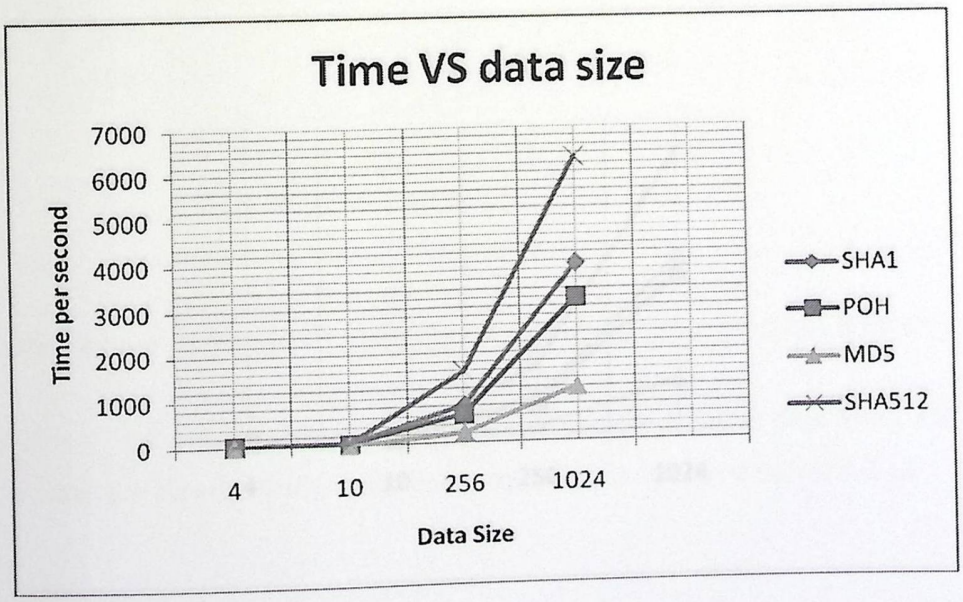


FIGURE 4.7: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 7X7

On the matrix size 7x7, the time per second of POH algorithm is larger than MD5, and smaller than SHA1, SHA-512 .

- **Comparison Based on Matrix Size 8x8**

Using different data size we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 8x8. The matrix is

$$R = \begin{bmatrix} 100 & 110 & 120 & 200 & 210 & 220 & 300 & 310 \\ 320 & 400 & 410 & 420 & 500 & 510 & 520 & 120 \\ 99 & 98 & 90 & 123 & 405 & 150 & 200 & 305 \\ 9 & 1100 & 1000 & 1010 & 1100 & 12 & 600 & 700 \\ 710 & 770 & 900 & 910 & 920 & 930 & 940 & 950 \\ 960 & 970 & 980 & 990 & 1000 & 1010 & 1020 & 1030 \\ 1040 & 1050 & 1060 & 1070 & 1080 & 1090 & 1100 & 1110 \\ 1120 & 1130 & 1140 & 1150 & 1160 & 1170 & 1180 & 1190 \end{bmatrix}$$, with modular value=128.

TABLE 4.8 : TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 8X8.

		Matrix Size 8x8			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*8.918*	*3.312*
	10KB	*38.563*	65.012	*34.701*	*11.121*
	256KB	*831.452*	1601.632	*706.017*	*214.441*
	1024KB	*3948.851*	6328.912	*3553.382*	*1188.127*

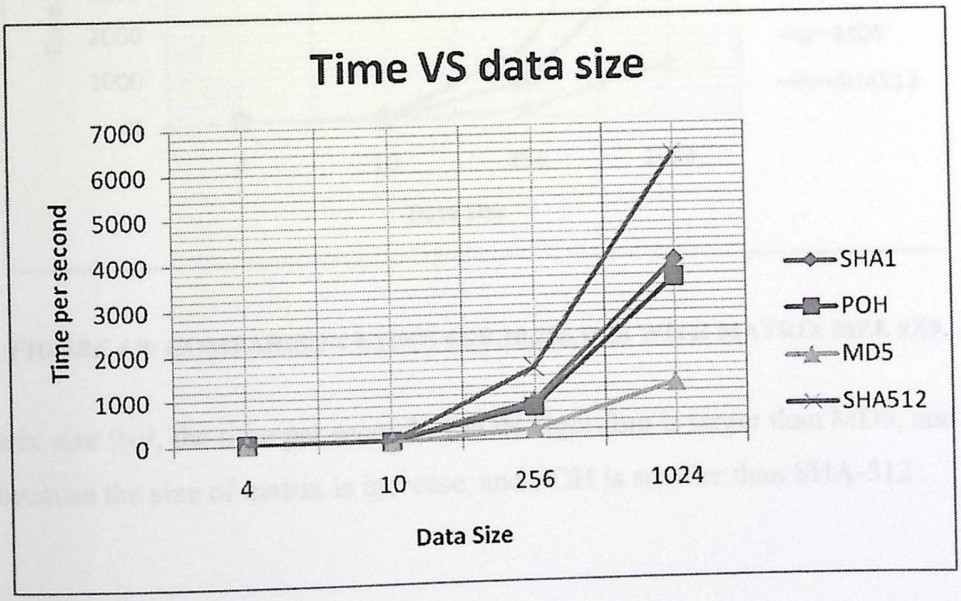


FIGURE 4.8: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 8X8.

On the matrix size 8x8, the time per second of POH algorithm is larger than MD5, and smaller than SHA1, SHA-512 .

- **Comparison Based on Matrix Size 9x9**

Using different data size we measure the time per second for MD5, SHA1, SHA512, and our algorithm POH on matrix size 9x9.

TABLE 4.9: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 9X9.

		Matrix Size 9x9			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*10.183*	*3.312*
	10KB	*38.563*	65.012	*37.981*	*11.121*
	256KB	*831.452*	1601.632	*813.748*	*214.441*
	1024KB	*3948.851*	6328.912	*3889.254*	*1188.127*

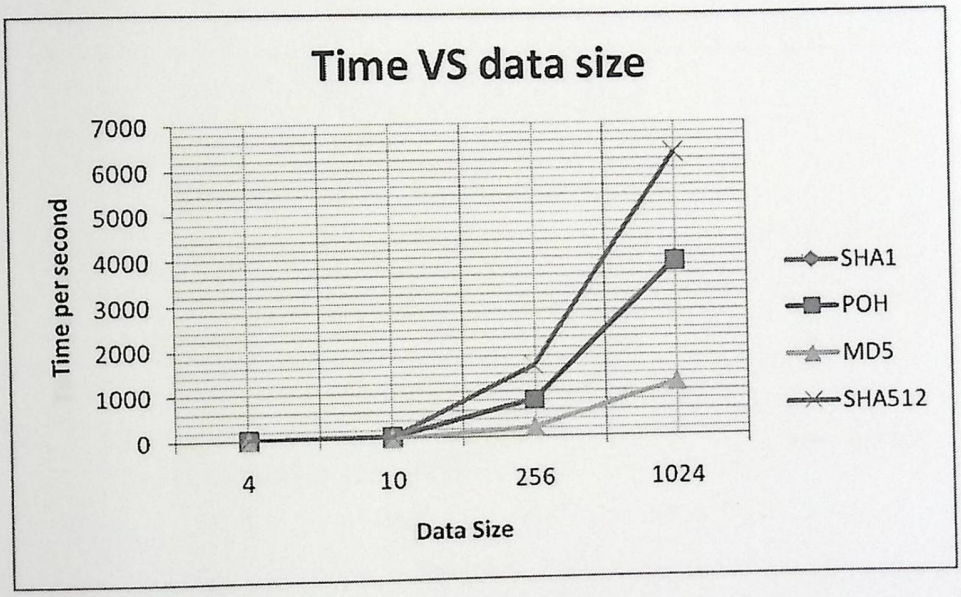


***FIGURE 4.9*: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 9X9.**

On the matrix size 9x9, the time per second of POH algorithm is larger than MD5, and very close to SHA1; because the size of matrix is increase, and POH is smaller than SHA-512 .

- **Comparison Based on Matrix Size 10x10.**

Using different data size we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 10x10.

TABLE 4.10: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 10X10.

		Matrix Size 10x10			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*11.989*	*3.312*
	10KB	*38.563*	65.012	*38.697*	*11.121*
	256KB	*831.452*	1601.632	*922.319*	*214.441*
	1024KB	*3948.851*	6328.912	*3962.573*	*1188.127*

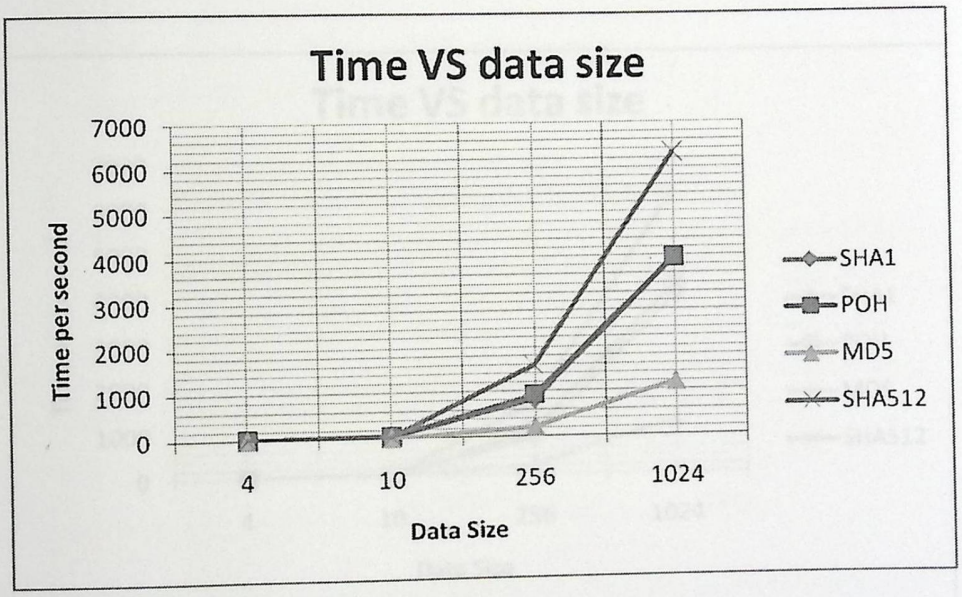


FIGURE 4.10: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 10X10

On the matrix size 10x10, the time per second for our POH algorithm is the same as SHA1, and smaller than SHA-512.

- **Comparison Based on Matrix Size 11x11.**

Using different data size we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 11x11.

TABLE 4.11: TIME PER SECOND USING 4KB,10,256,1024 WITH MATRIX SIZE 11X11.

		Matrix Size 11x11			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*13.989*	*3.312*
	10KB	*38.563*	65.012	*45.697*	*11.121*
	256KB	*831.452*	1601.632	*998.319*	*214.441*
	1024KB	*3948.851*	6328.912	*4213.643*	*1188.127*

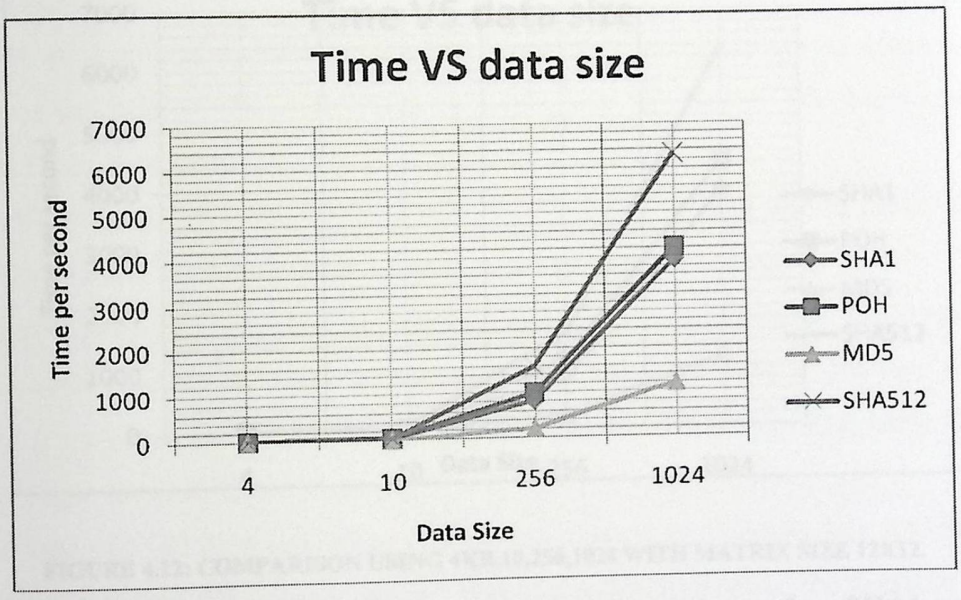


FIGURE 4.11: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 11X11

On matrix size 11x11, the time per second of POH algorithm is larger than SHA1.

- **Comparison Based on Matrix Size 12x12.**

Using different data size we measure the time per second for MD5, SHA1, SHA-512, and our algorithm POH on matrix size 12x12.

TABLE 4.12: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 12x12.

		Matrix Size 12x12			
		Hash algorithm			
Data Size		SHA1	SHA512	POH	MD5
	4KB	*12.964*	27.989	*16.989*	*3.312*
	10KB	*38.563*	65.012	*50.697*	*11.121*
	256KB	*831.452*	1601.632	*1124.319*	*214.441*
	1024KB	*3948.851*	6328.912	*4450.101*	*1188.127*

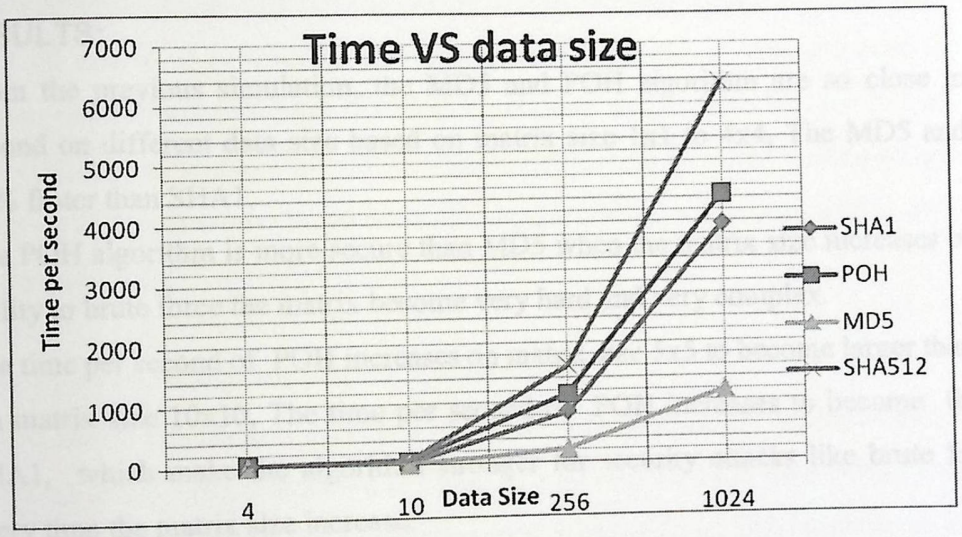


FIGURE 4.12: COMPARISON USING 4KB,10,256,1024 WITH MATRIX SIZE 12X12.

On matrix size 12x12, the time per second of POH algorithm is larger than SHA1 and less than SHA-512 .

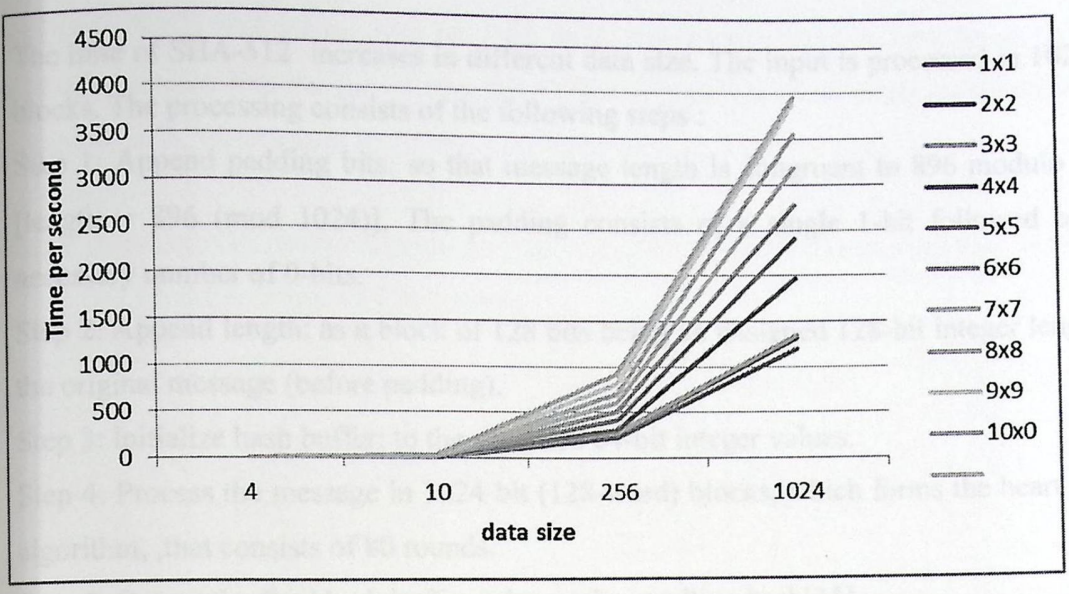


FIGURE 4.13: Time per second of POH in different matrix size and different data size.

4.2 RESULTS:

1. From the previous simulation, the MD5 and POH algorithm are so close in time per second on different data size based on matrix size 1x1 to 4x4, The MD5 and POH are 30% faster than SHA1.
2. The POH algorithm is more secure than MD5 when the matrix size increases because the ability to brute force the matrix become very hard and very complex.
3. The time per second of POH increases on matrix size 5x5 to become larger than MD5.
4. On matrix size 10x10, The time per second of POH increases to become the same as SHA1, which make the algorithm stronger for security attacks like brute force attack every time the matrix size increase.
5. The time of POH increases when matrix size increase in different data size but the main factor that increase the time is the last step of the algorithm when the digest of message is created.
6. The time of SHA1 increases when the size of the data increase because of 80 round used in the algorithm. A sequence of logical functions f(0), f(1),..., f(79) are used. Each f(t), 0 <= t <= 79, operates on three 32-bit words B, C, D and produces a 32-bit word as an output. f(t;B,C,D) is defined as follows: for words B, C, D,

```
f(t;B,C,D) = (B AND C) OR ((NOT B) AND D)          ( 0 <= t <= 19)
f(t;B,C,D) = B XOR C XOR D                         (20 <= t <= 39)
f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D)   (40 <= t <= 59)
f(t;B,C,D) = B XOR C XOR D                         (60 <= t <= 79).
```

7. The time of SHA-512 increases in different data size. The input is processed in 1024-bit blocks. The processing consists of the following steps :

   Step 1: Append padding bits: so that message length is congruent to 896 modulo 1024 [length ≡ 896 (mod 1024)]. The padding consists of a single 1-bit followed by the necessary number of 0-bits.

   Step 2: Append length: as a block of 128 bits being an unsigned 128-bit integer length of the original message (before padding).

   Step 3: Initialize hash buffer: to the specified 64-bit integer values.

   Step 4: Process the message in 1024-bit (128-word) blocks, which forms the heart of the algorithm, ,that consists of 80 rounds.

   Step 5: Output the final hash buffer value as the resulting hash[15].

   The SHA-512 algorithm has the property that every bit of the hash code is a function of every bit of the input. The complex repetition of the basic function F produces results that are well mixed, it is unlikely that two messages chosen at random, even if they show similar regularities, will have the same hash code, unless there is some hidden weakness. The difficulty of coming up with two messages having the same message digest is on the order of 2^{256} operations, while the difficulty of finding a message with a given digest is on the order of 2^{512} operations[15].

4.3 Security of POH Against Brute Force Attack:

The proposed model generates a 128-bit hash value, by using matrix multiplication in the first step to create an initial hash, and then add digest to generate the final hash value. This algorithm is strongly secure against brute force attack because it has a fixed 128-bit string as an output. If the modular value =128, and matrix size *nxn* the attacker needs $2^{(7)^{n*n}}$ to calculate the origin of the matrix as we can see in Table 4.13.

TABLE 4.13: SECURITY OF POH.

	Matrix size			
POH	1x1	2x2	3x3	4x4
	$2^{(7)}$	$2^{(7)^4}$	$2^{(7)^9}$	$2^{(7)^{16}}$

If we use $(n = 128)$, and matrix size $(s = 6)$, then the security level is very large. A huge number of tries needed. The number of tries is7.2370e+075, table 4.14 summarizes security level of POH against brute force attack in different modular value and matrix size.

TABLE 4.14 : NUMBER OF TRIES NEEDED BY THE HACKER TO CRYPTANALYSIS THE MATRIX.

		Modular Value of The System				
		26	34	35	71	128
Matrix Size	2x2	456976	1336336	1500625	25411681	268435456
	3x3	5.4295e+012	6.0717e+013	7.8816e+013	2.7799e+015	9.2234e+018
	4x4	4.3609e+022	3.1891e+024	5.0709e+024	2.8579e+027	5.1923e+033
	5x5	2.3677e+035	1.9363e+038	3.9967e+038	7.9448e+042	4.7890e+052
	6x6	8.6904e+050	1.3591e+055	3.8588e+055	5.9720e+061	7.2370e+075
	7x7	2.1562e+069	1.1027e+075	4.5639e+075	1.2138e+084	1.7918e+103
	8x8	3.6165e+090	1.0343e+098	6.6123e+098	6.6714e+109	7.2684e+134
	9x9	4.1005e+114	1.1215e+124	1.1736e+125	9.9145e+138	4.8307e+170

number of tries required to brute force the MD5=3.6207e+40.

number of tries required to brute force the SHA1=1.4133e+55.

number of tries required to brute force the SHA-512=9.8727e+172.

From this comparison the number of tries to brute force the MD5 its close to the proposed model in number of tries when matrix size =5x5 with modular value=71, and close to SHA1 when matrix size=6x6 with modular value=128, and close to SHA-512 when matrix size=9x9 with modular value=128.

If we take the matrix size 2x2 on modular value=26 the number of tries = 456976. If each try need $6.2*10^{-6}$ sec, then the attackers theoretically need $456976*6.2*10^{-6} = 2.83\,\text{sec}$. We design an algorithm to brute force the matrix of size 2x2 with modular value=26 using Matlab. The time the attackers practically need = 2.745 sec . The difference between the theoretical value and practical value is very small.

If we take the matrix size 2x2 on modular value=71 the number of tries = 25411681. If each try need $6.2*10^{-6}$ sec, then the attackers theoretically need $25411681*6.2*10^{-6} = 157.468\,\text{sec}$.

To brute force the matrix of size 2x2 with modular value=71. The time the attackers practically need $=153.5362\,\text{sec}$. The difference between the theoretical value and practical value is very small.

The previous calculation with respect to maximum value of any element in the matrix = modular value n, this tries will be very big if we use the maximum value of any element in the matrix $=n^2$ (the square of modular value).

4.4 Security of POH Against Dictionary Attack:

In this research, we use a salt value added to input message. The benefit provided by using a salted password is that a dictionary attack against the stored values becomes impractical if the salt is large enough, an attacker would not be able to create a pre computed lookup table (a rainbow table) of hashed values (password + salt); because it would take too much space.

4.5 Security of POH Against Collisions:

hash algorithms must have the following properties:

- *Preimage resistance*

Given a hash h it should be hard to find any message m such that $h = hash(m)$. This concept is related to one-way function. Any functions that lack this property are vulnerable to preimage attacks [4]. POH satisfy this property. We can prove that the non-invertible matrix R cannot be reversed, if we Calculate the determinant of this matrix d. If $d = |R|$ not relatively prime to n, R^{-1} will not exist and we cannot calculate the value of $(H(V))^{-1}$ [61].

- *Collision resistance*

It should be hard to find two different messages $m1$ and $m2$ such that $hash(m1) = hash(m2)$ [4]. If it happens it is called a cryptographic hash collision. This property is sometimes called a *strong collision resistance*[4]. After applying Hamming distance algorithm as we see in figure 4.15 between two string into 50 samples of hash value for MD5, SHA1 and POH, the result of hamming distance on average :

MD5=81,SHA1=91,POH=94.

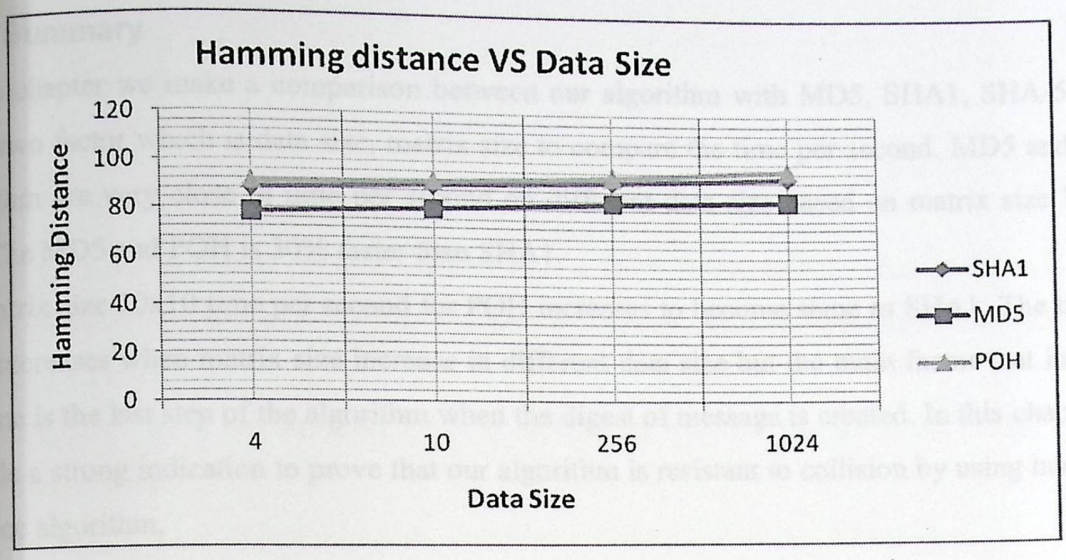


FIGURE 4.14: HAMMING DISTANCE BETWEEN TWO HASH VALUE STRING IN DIFFERENT DATA SIZE FOR MD5, POH, SHA.

From this result we can say that the POH has a low similarity of the final hash value . This is a strong indication to collision resistance as we see in fig 4.14.

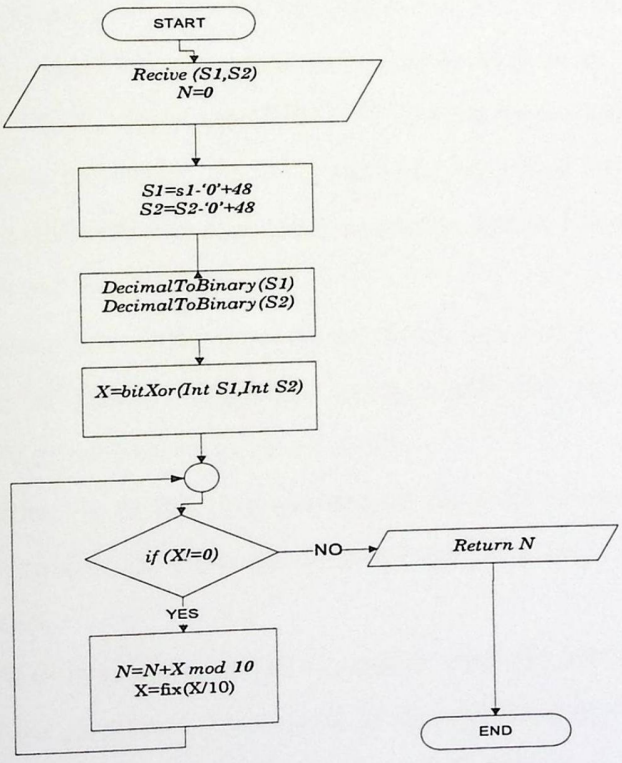


FIGURE 4.15: HAMMING DISTANCE ALGORITHM.

4.6 Summary

In this chapter we make a comparison between our algorithm with MD5, SHA1, SHA-512 by using two factor which is data size, matrix size to compare the time per second. MD5 and POH algorithm are very close in time per second on different data size based on matrix size 1x1 to 4x4, The MD5 and POH is 30% faster than SHA1.

On matrix size 10x10 time per second for POH increases to become same as SHA1. The time of POH increases when matrix size increase in different data size but the main factor that increase the time is the last step of the algorithm when the digest of message is created. In this chapter we provide a strong indication to prove that our algorithm is resistant to collision by using hamming distance algorithm.

Chapter 5

5.1 Conclusions

Cryptography is considered as one of the top hot research topics, so does exactly the security of encryption algorithm, it is well known, that there is a tradeoff between the quality and the time of encryption algorithm.

- In this research, we introduced a new method to convert the invertible matrix to non-invertible matrix .
- We proposed a new one way hash algorithm based on matrix multiplications.
- We prove that POH satisfies the requirement of the one way hash algorithm.
- We provide a solution for dictionary attack problem by using salt algorithm.
- After implementing hamming distance algorithm between two strings the POH has a strong indication to collision resistance.
- From this research, we notice that when we increase the matrix size, the level of security will be increased notably, regardless of the modular value of the system.
- We compared POH with MD5 , SHA1, and SHA-512 and we have some result:
    - From the previous simulation, the MD5 and POH algorithm are very close in time per second in a different data size based on matrix size 1x1 to 4x4, The MD5 and POH is 30% faster than SHA1.
    - The time per second for POH increases on matrix size 5x5 to become larger than MD5 because of matrix complexity which makes the algorithm stronger in security against attacks like brute force attack.
    - On matrix size 10x10, the time per second for POH increases to become the same as SHA1 which make the algorithm stronger in security against attacks like brute force attack.

As we mentioned in chapter four the time of SHA1 increases when the size of data is increase; because of 80 round used in the algorithm .A sequence of logical functions f(0), f(1),..., f(79) is used. Each f(t), 0 <= t <= 79, operates on three 32-bit words B, C, D and produces a 32-bit word as output. The time of SHA-512 increases for different data size because; the input is processed in 1024-bit blocks. The SHA-512 algorithm has the property that every bit of the hash code is a function of every bit of the input. The complex repetition of the basic function F produces results that are well mixed, it is unlikely that two messages chosen at random, even if they show similar

regularities, will have the same hash code, unless there is some hidden weakness[15]. The difficulty of coming up with two messages having the same message digest is on the order of 2^{256} operations, while the difficulty of finding a message with a given digest is on the order of 2^{512} operations[15].

As we can see in figure 4.13 the proposed model time per second increases when matrix size and modular value increase. But the increment in time have a strong security power against attacks.

5.2 Future Work

This research proposes a technique to design a practical one-way hash algorithm by using non-invertible matrix which cannot be reversed to produce a hash value. We prove the four requirements which a practical one way hash algorithm needs. In the future work:

- We can develop by the Gods permission our algorithm to be a parallel practical one way hash algorithm; this new technique may decrease the time in hashing with a strong level of security.
- We can develop a complete cryptosystem web based application based on the POH algorithm.
- We can develop a POH algorithm for any matrix not only square matrix.
- Also we can build an algorithm that try to find a collision in the proposed algorithm.

Reference

[1] Abutaha. M, et.al,(2011)."survey : a cryptography is the science of information secuirty". International Journal of Computer Science and Security (IJCSS) Volume 5, Issue 3.

[2] Abutaha. M, Mousa farajalla and Radwan tahboub(2011)."Practical one way hash algorithm based on matrix multiblications". *International Journal of Computer Applications* 23(2):33–37. Published by Foundation of Computer Science.

[3] Ahmed. Y. M and Alexander. G. C, (2009): " Hill cipher modification based on eigenvalues HCM-EE ". Second International Conference on Security of Information and Networks, 6-10 – October – 2009, Salamis Bay Conti Resort, Gazimagusa, North Cyprus pp: 164-167.

[4] Andru P. T, (2007): " Romantic Tantalizers Cipher as an Improved Version of Hill Cipher ". IF3058 Cryptography Technical Report Assignment, pp: 1-7.

[5] Badeau. J. S, (1983): " The Genius of Arab Civilization ", Second Edition. MIT Press, USA.

[6] Bao. N. T, and Thuc. D. N, (2008): " Modular Matrix Cipher and Its Application in Authentication Protocol ". Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 6-8 – August – 2008, IEEE Computer Society, Phuket, pp: 318-323.

[7] Bibhudendra. A and et al, (2008): " Image Encryption by Novel Cryptosystem Using Matrix Transformation " First International Conference on Emerging Trends in Engineering and Technology, 16-18 – July – 2008, IEEE Computer Society, Maharashtra, pp: 77-81.

[8] Bibhudendra. A, and et al, (2006): " Novel Methods Of Generating Self-Invertible Matrix For Hill Cipher Algorithm ". International Journal of Security, Vol (1), pp: 14-21.

[9] Bibhudendra. A, and et al, (2008): " A Novel Cryptosystem Using Matrix Transformation ". SPIT-IEEE Colloquium and International Conference, Vol (4), pp: 92-95.

[10] Bibhudendra. A. C, and et al, (2008): " Novel Modified Hill Cipher Algorithm ". Proceedings of the International Conference on Emerging Technologies and Applications in Engineering Technology and Sciences, 13-14 – January – 2008, Gujarat, India, pp: 126-130.

[11] Carl. F. G, (1986): " Disquisitiones arithmeticae", English Edition Translated By Arthur A. C. Springer-Verlag, USA.

[12] Chapple. M, and Solomon. M, (2005): " Information Security Illuminated " First Edition. Jones and Bartlett Publishers, USA.

[13] Charlie, O, and Behzad, S, (2007): " A Parallel Algorithm for determining the inverse of a matrix for use in block cipher encryption/decryption ". The Journal of Supercomputing, Vol (39), pp: 113-130.

[14] Cheng-qing, and et al, (2008): " Cryptanalysis of an image encryption scheme based on the Hill cipher ". Journal Of Zhejiang University Science A, Vol (9), pp: 1118-1123.

[15] Childs, J.R. (2000): " General Solution of the ADFGVX Cipher System ". Aegean Park Press, USA.

[16] Chu-Hsing, L, and et al, (2004): " Comments On Saeednia's Improved Scheme For The Hill Cipher ". Journal of the Chinese Institute of Engineering, Vol (27), pp: 743-746.

[17] Daemen. J, and Rijmen. V, (2002): " The Design of Rijndael: The Wide Trail Strategy Explained ". Springer-Verlag, USA.

[18] Daemen. J, and Rijmen. V, (March, 2001): " Rijndael: The Advanced Encryption standard ". Dr. Dobb's Journal, Vol (26), pp: 137-139.

[19] Delfs. D, and Helmut. K, (2007): " Introduction To Cryptography: Principles and applications ", Second Edition. Springer Science & Business Media, Germany.

[20] Dieter. G, (2005): " Computer Security ", Second Edition. John Wiley & Sons, UK.

[21] Essenberg .M, (1999) ,"linear algebra project with Mathematica", in Proceedings of the Twelfth Annual International Conference on Technology in CollegiateMathematics, Gail Goodell, editor, Addison-Wesley, Boston, Mass, 100–104.

[22] Farajallah. M (2010): "Self Generating Multi Key Cryptosystem For Non-Invertible Matrices Based on Hill Cipher" Master Thesis, Al-Quds University, Jerusalem, Palestine.

[23] Forouzan. A. B, (2007): " Cryptography and Network Security ", First Edition. McGraw-Hill, USA.

[24] Gligoroski .D, Smile Markovski, and Svein J. Knapskog,(2006):"A Secure Hash Algorithm with only 8 Folded SHA-1 Steps", IJCSNS International Journal of 194 Computer Science and Network Security, VOL.6 No.10

[25] Hadi S. A, and Ali. H. M, (2009): " Encrypted Block Code ". Australian Journal of Basic and Applied Sciences, Vol (3), pp: 1315-1318.

[26] Hamamreh. R, and Farajallah. M, (May, 2009): " Design of a Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher ". International Journal of Computer Science and Network Security, Vol (9), pp: 12-21.

[27] Hill S. L, (1929):" Cryptography in an algebraic alphabet ". American Math. Monthly, Vol (36), pp: 306-312.

[28] Hill, S. L, (1931): " Concerning Certain Linear Transformation Apparatus of Cryptography ". American Math. Monthly, Vol (38), pp: 135-154.

[29] Hoffstein. J, and et al, (2008): " An Introduction to Mathematical Cryptography ", First Edition. Springer Science & Business Media, Germany.

[30] Ismail, I. A, and et al, (2006): " How to repair the Hill cipher ". Journal of Zhejiang University SCIENCE, co-published with Springer-Verlag GmbH, Vol (7), pp: 2022-2030.

[31] Jeffrey, O, and et al, (January, 2005): " On the Keyspace of the Hill Cipher ". Cryptologia, Vol (29), pp: 59-72.

[32] Jiqiang. L and et al, (2008): " New Impossible Differential Attacks on AES ". Proceedings of the 9th International Conference on Cryptology in India: Progress in Cryptology-Indocrypt, Springer, Kharagpur, India PP: 279-293.

[33] Johannes M, (2005): " One-Way Encryption and MessageAuthentication".International Journal of Computer Science and Network Security.

[34] Jrvinen .K, Matti Tommiska and Jorma Skytt (2005): "Hardware Implementation Analysis of the MD5 Hash Algorithm", Proceedings of the 38th Hawaii International Conference on System Sciences.

[35] Jyotirmayee. M, (2009): " Modified Hill- Cipher And CRT Methods In Galois Field Gf(2M) For Cryptography", National Institute Of Technology University, Rourkela.

[36] Kenneth, H. R. (1992): " Elementary Number Theory and Its Applications " Third Edition. Addison-Wesley, Germany.

[37] Kumar. Y. N, and Narendra. B, (2008): " Modification Of Hill Cipher Technique Using Self Repetitive Matrix (Modular Arithmetic) And Correlation Of Eigen Values Of Matrix With The Exponent N". National Institute of Technology, Rourkela.

[38] Landau.S, (Febraury, 2004): " Polynomials in the Nation's Service: Using Algebra to Design the Advanced Encryption Standard ". American Mathematical Monthly pp: 89-117.

[39] Lewand. R. E, (2000): " Cryptological Mathematics ", First Edition. Mathematical Association of America Washington, USA.

[40] Lucas. M. A, (1995): " Thomas Jefferson wheel cipher ", Monticello Research Department, Thomas Jefferson Foundation, Charlottesville, VA.

[41] Maret. S, (1999): " Cryptography Basics PKI ", First Edition. Dimension Data SA, Switzerland.

[42] NIST, (2008): " Secure Hash Standard ", Cryptography, FIPS PUB 180-3, U.S. Department of Commerce, USA.

[43] Panigrahy. S K, and et al, (2008): " Image Encryption Using Self-Invertible Key Matrix of Hill Cipher Algorithm". First International Conference on Advances in Computing, 21-22 – February – 2008, Chikhli, India.

[44] Panigrahy. S. K, and et al, (2009): " On the Privacy Protection of Biometric Traits: Palmprint, Face, and Signature ". Proceedings of Second International Conference, IC3 17-19 – August – 2009, Noida, India, pp: 182-193.

[45] Ralph. E, and Weierud, F. (1987): " Naval Enigma: M4 and Its Rotors ". Cryptologia, Vol (11), pp: 235-244.

[46] Ramchandra. S. M, and Pallavi. V. C, (May, 2009): "Encrypting Informative Image by Key Image using Hill Cipher Technique". International Journal of Recent Trends in Engineering, Vol (1), pp: 568-570.

[47] Rangel-Romero. R, and Vega-Garcia, (2008): " Comments on " How to repair the Hill cipher "". Journal of Zhejiang University SCIENCE A, Vol (9), pp: 211-214.

[48] Reinhard. W, (2007): " Cryptology Unlocked ", Translation Edition By Angelika Shafir. John Wiley & Sons, UK.

[49] Rodríguez, H. F. and et al, (2006): " Cryptographic Algorithms on Reconfigurable Hardware ", First Edition. Springer, USA.

[50] RSA Data Security , (1992) retrieved from http://www.faqs.org/rfcs/rfc.html.

[51] Rushdi A. Hamamreh, Mousa Farajallah(2009), "Design of a Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher", International Journal of Computer Science and Network Security; pp 11-16.

[52] Salomon. D, (2003): " Data Privacy and Security " First Edition. Springer-Verlag New York, Inc. USA.

[53] Sastry. U, and Janaki. V, (2008): " A Modified Hill Cipher with Multiple Keys ". International Journal of Computational Science, Vol (2), pp: 815-826.

[54] Sastry. U, and Ravi. S, (November, 2007): " Modified Hill Cipher with Interlacing and Iteration ". Journal Of Computer Science, Vol (3), pp: 854-859.

[55] Schneier. B, and et al, (1996): " Applied cryptography ", Second Edition. John Wiley & Sons, USA.

[56] Shahrokh. S, (October, 2000): " How To Make The Hill Cipher Secure". Cryptologia, Vol (24), pp: 353-360.

[57] Shannon, C. (1949): " Communication Theory of Secrecy Systems ". Bell Syst, Tech. J., Vol (28), pp: 656-715.

[58] Shoups .V(2000) :"A composition theorem for universal one-way hash functions", in Proc. Eurocrypt this is a revised version of IBM Research Report RZ 3147 .

[59] Stallings. W, (2006): " Cryptography and network security, Principles and practices ", Fourth Edition. Pearson Prentice Hall, USA.

[60] Thomas. K, (July , 1998): " The Myth Of The Skytale ". Taylor & Francis, Vol (33), pp: 244-260.

[61] Tiwari. H.(2010), "Cryptographic Hash Function: An Elevated View".European Journal of Scientific Research. ISSN 1450-216X Vol.43 No.4 .pp.452-465.

[62] Yi-Shiung. Y, and et al, (1991): " A New Cryptosystem Using Matrix Transformation ". Proceedings of IEEE International Canahan Conference on Security Technology, 1-3 – October – 2008, Taipei, Taiwan pp: 131-138.

[63] Zheng .Y, Josef Pieprzyk and Jennifer Seberry: (1993).HAVAL " A One-Way Hashing Algorithm with Variable Length of Output" Advances in Cryptology -- AusCrypt'92, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Vol. 718, pp. 83-104.

Appendix A

Appendix A: Digest Creation

```
    /*" Then perform the following additions. (That is increment each
       of the four registers by the value it had before this block
       was started.) "[58]*/
    A = A + AA
    B = B + BB
    C = C + CC
    D = D + DD

  end /* of loop on i */
Output

" The message digest produced as output is A, B, C, D. That is, we
 begin with the low-order byte of A, and end with the high-order byte
 of D"[58].
```

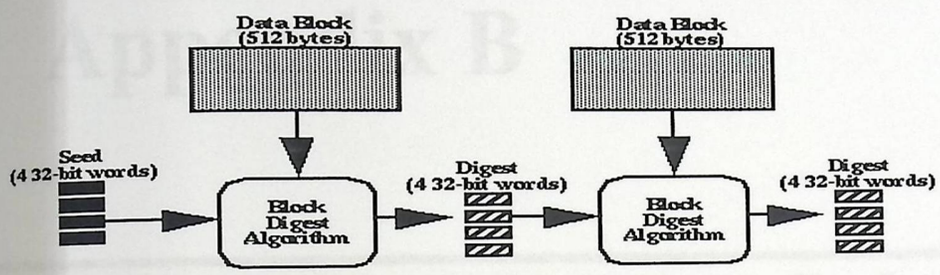


Figure appendix.A:Block diagram of digest creation

Appendix B

Appendix B: Matlab Code

- Sample of Matlab Code

//

```
function D=Hashing(R,V,n)
fid=fopen(V);
M = fread(fid);
fclose(fid);

V=M;
t1=clock;

l=size(R);% size of matrix
l=l(1);% number of rows =number of column
V=compat(l,V)
%R=salt(R)
while(gcd(mod(det(R),n),n)==1)% invertible since the  matrix det hasnt common
factor with n

    R=gen_mat(R,n)
%R=R+eye(l);
end
i=1;
j=0;
u=0;
V=V';
sv=size(V);
j2=0;
while(u<sv(1))%divide the original message into stream of size l
    i=1;
    while(i<=l)
       j2=j2+1;
       r(i)=V(j2);
       i=i+1;

    end
    size(r);
    size(R);
    r=double(r);

    H=mod(R*r',n)

    i=1;
    while(i<=l)
       j=j+1;
       H(i)=concat(H(i),V(j));
       FH(j)=H(i) ;
       i=i+1;

    end

    u=u+l;

end
FH


dlmwrite('myfile.txt',FH);
D=dig('myfile.txt');
```

```
t2=clock;
disp(etime(t2,t1))

return ;


//////////////////////////////////////////

function d=compat(l,V)

len=length(V);% number of char on message
i=1;
while(i<=len)
    d(i)=uint16(V(i));% convert to Ascii array.
    i=i+1;
end

while(mod(len,l)~=0)%padding
   d(i)=65+d(i-1);
   len=i;
   i=i+1;
end
return ;

////////////////////////////////////////////////////////////


////////////////////////////////////////////////////////////////////
function R=gen_mat(R,n)
l=size(R);% matrix size
l=l(1);

i=1;
while(i<=l)% to store the main diagonal matrix in one dimensional array (m)
m(i)=R(i,i);
i=i+1;
end
i=1;
while((gcd(mod(det(R),n),n)==1) &&(i>=0))%
    R(i,i)=R(i,i)+1;
    if(R(i,i)==n*n);
        R(i,i)=m(i);
        i=i-1;
    end
end
%R=R+eye(l);

return ;

//////////////////////////////////////////////////////////////////////////


function sv=concat(d,sv)
dd=d;
while (fix(d)~=0)
sv=sv*10;
d=fix(d/10);
```

Appendix C

Appendix C: Papers

- **Paper 1:Published(IJCA)**
- **Paper 2:published(ACS)**
- **Paper 3:published(IJCSS)**

International Journal of Computer Applications (0975 – 8887)
Volume 23– No.2, June 2011

A Practical One Way Hash Algorithm based on Matrix Multiplication

Mohammed Abu Taha
College of Administrative Sciences and Informatics
Palestine Polytechnic University
Hebron, Palestine

Mousa Farajallah
College of Engineering and Technology
Palestine Polytechnic University
Hebron, Palestine

Radwan Tahboub
College of Engineering and Technology
Palestine Polytechnic University
Hebron, Palestine

ABSTRACT
It is well known that Hash algorithm works in one way, and it cannot be reversed. We can build a new algorithm by using Hill cipher technique .Since its invention in 1929, Hill cipher algorithm which is one of the most famous symmetric cryptosystems .Hill cipher requires the inverse of the key matrix for decryption. This inverse not always exists, so we can use non-invertible matrices to propose a model for our new hash algorithm, and we proof the four requirements that needed to design a practical one way hash algorithm.

General Terms
Cryptographic algorithm, Practical One Way Function

Keywords
Hill cipher technique, Non-invertible matrix, hash algorithm, One-way hash function, plaintext, integrity.

1. INTRODUCTION
Cryptography is a mixture of mathematics and computer science. It is the study and the ability of hiding data. It is also used in other technologies and business applications such as payment. Cryptography has increasingly been used to secure information, But secure data of today could be broken in the future. Cryptography is the science of codes and ciphers. It includes many algorithms and techniques that transfer data safely. It is also inaccessible for non-permitted readers or writers. Computer Security aims to protect the automated information system in order to provide the goals of preserving the integrity, availability and confidentiality of information system resources and services. There are many attacks that harm computers and information security. There are two general methods that attack a symmetric encryption scheme. The first one , known as cryptanalysis, and it relies on the nature of algorithm and some information of the general characteristics of the plaintext message or even some plaintext samples. This kind of attacks exploits the characteristics of the algorithm in order to attempt to infer a specific plaintext or to infer the encryption used key. If the attack succeeds in inferring the key, all the messages plaintext and key are compromised. The second method is the brute-force, and it is used to try every possible key in apart of cipher text until plaintext translation is obtained [16].

Many approaches and countermeasures are set to protect systems' security. A countermeasure is an any mean or any technique that is used to prevent security attack. Ideally, a countermeasure can be devised to prevent a particular type of attacks from harming a computer or information security. If prevention is impossible, or fails in some cases, the goal is to detect the attack, then to recover from the effects of the attack. the countermeasure itself may infer new vulnerabilities. In such cases, residual vulnerabilities may remain after the imposition of countermeasures. Such vulnerabilities may be exploited by threat agents representing a residual level of risk to the assets. Systems owners will seek to minimize that risk by giving other constraints [16].

This paper contains five sections. Firstly, it compares between Symmetric and a symmetric encryption. Secondly, it talks about hash algorithm definition, description, applications and the requirements for one-way hash functions. thirdly, it handles the hill cipher encryption technique. Fourthly, contains our proposed model for one-way hash algorithm with mathematical representations and proves that the inverse of noninvertible matrix does not exist. Finally, is the conclusion and suggestions for future work. [16][7].

2. SYMMETRIC AND A SYMMETRIC ENCRYPTION
Asymmetric encryption uses two separate keys, in the contrary with the symmetric encryption which uses one key only. Symmetric encryption has five components [16], [7]:

- Plaintext: It is the message or any data inserted to algorithm as input.
- Encryption algorithm: It makes some of transformations and substitutions to the message.

• Secret key: One of the inputs of the algorithm and the encryption algorithm itself depend on key in the transformation and substitution process.

• Cipher text: The message that had been scrambled.

• Decryption algorithm: It reverses the encryption algorithm in execution.

A public-key encryption technique components [16],[7]:

• Plaintext: It is the message or any data inserted to algorithm as input.

• Encryption algorithm: It makes some transformations and substitutions to the message.

• Public and private key: A selected pair of keys. if one of the keys is used for encryption, the other one is used for decryption. All transformations and substitutions that are made by the encryption algorithm rely on the public or private key which submitted as input.

• Cipher text: The message that had been scrambled.

• Decryption algorithm: It produces the plaintext from the cipher by reversing the encryption algorithm.

The public key of the pair is made public for others to handle . But the private key is handled only by the owner. A public key cryptographic algorithm depends on a key for encryption and a different key for decryption.

3. HASH ALGORITHM

3.1 Definition, Descriptions and Applications

Algorithm changes messages and text into a fixed string of digits. It usually does so for systems security integrity, confidentiality and availability. The one-way means that it is hard to recover the original text from the hash value string. A one-way hash function is used to create digital signatures. Which in its turn identify and authenticate the sender and the distributed message digitally. One-way Hash functions have an important primitive cryptographic, and it can be used to solve any problems including authentication and integrity. Hash function is a well-defined procedure that converts a large data into a small one. The returned value from hash function is called hash code [6]. One-way hash function is a function that converts a variable string length into a fixed length binary sequence that cannot be reversed, [10],[15],[13]. The Microsoft cryptographic providers support three hash algorithms: MD4, MD5 and SHA [6]. An important element in many computer security services and applications is the usage of cryptographic algorithms .The first type is symmetric encryption like DES algorithm, which is used primarily in the widest variety of contexts. to provide confidentiality. Another type is a secure hash functions like SHA512, MD5 which are used in message authentication. The third type is public-key encryption like RSA. Asymmetric encryption and secure hash functions are combined together to produce an extremely useful tool; Hash functions are used in cryptography with digital signatures for ensuring data integrity when hash used with digital signatures. A public available hash function hashes the message and signs the resulting hash value. The part that receives the message hashes the message and checks whether the block size is authentic for the given hash value [16],[7].

One-way hash function is an alternative to the message authentication code (MAC), which accepts a variable size input and produces a fixed size message. Unlike the MAC, hash code does not require a key as an input to authenticate the message, but a message digest is sent with the message in an authenticated way. The message digest can only be encrypted by using symmetric key if the sender and the receiver share the key. In this way, the authenticity is satisfactory by using public key encryption that does not require the keys to be distributed to the parties [16].

3.2 Requirements for one-way hash function

The following properties required for hash function to be useful:

- Applied to any size of data.
- Hash function (H) produces a fixed-length output.
- $H(X) = h$ is relatively easy to compute for any given x.
- One-way property.

Computationally infeasible to find x such as $H(X) = h$ (h is a hash value generated).

- Weak collision resistance.

Computationally infeasible to find $y \neq x$ such that $H(y) = H(x)$

- strong collision resistance

Computationally infeasible to find any pair (x, y) such as $H(y) = H(x)$

The first three properties are requirements for a practical application of a hash function to message authentication [16].

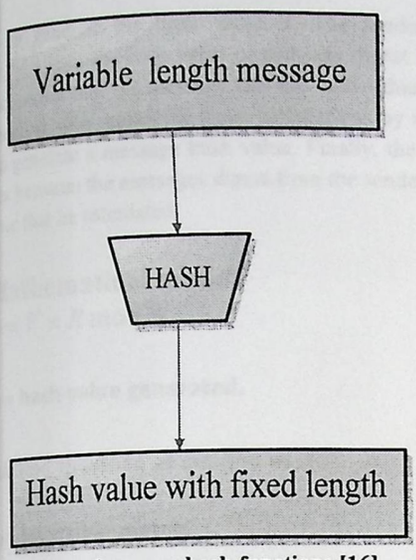


Figure1: secure hash functions [16].

4. HILL CIPHER

Hill cipher, invented by Lester S. Hill in 1929, uses matrix multiplication for mixing the plaintext [3]. The Hill cipher works on groups of letters in different ways [8], [9]. It works by displaying a group of letters as a vector. And encryption is done by matrix multiplication [7] [5].

Hill cipher satisfies properties that good cryptosystems would Have:

Diffusion: One change in plaintext character should affect as many characters as possible in cipher text. We know that hill cipher converts any plaintext character to number, and then inserts it in a matrix of column vector. If we take – be - as plaintext characters then it will be – 1, 4 -, the matrix of column vector will be $\begin{bmatrix}1\\4\end{bmatrix}$ then any change in plaintext must affect cipher text characters.

Confusion: The key should not relate to the cipher text. Hill cipher uses key matrix to encrypt the message and key inverse to decrypt it.

Hill Cipher example with Key Matrix 2×2 uses math equation with Condition: The key matrix has to be invertible relative to 26.

Given Plaintext: $p1p2p3p4.....pn-1pn$ Given Key Matrix: $k=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}$

Encryption:

1. Form vectors as follows:

$$\begin{pmatrix}p_1\\p_2\end{pmatrix}\begin{pmatrix}p_3\\p_4\end{pmatrix}\begin{pmatrix}p_5\\p_6\end{pmatrix}\cdots\begin{pmatrix}p_{n-1}\\p_n\end{pmatrix}$$

2. Multiply each vector by k to obtain a pair of cipher text letters:

$$k\begin{pmatrix}p_1\\p_2\end{pmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{pmatrix}p_1\\p_2\end{pmatrix}=\begin{pmatrix}c_1\\c_2\end{pmatrix}(\text{mod } 26)$$

$$k\begin{pmatrix}p_3\\p_4\end{pmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{pmatrix}p_3\\p_4\end{pmatrix}=\begin{pmatrix}c_3\\c_4\end{pmatrix}(\text{mod } 26)$$

....

$$k\begin{pmatrix}p_{n-1}\\p_n\end{pmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{pmatrix}p_{n-1}\\p_n\end{pmatrix}=\begin{pmatrix}c_{n-1}\\c_n\end{pmatrix}(\text{mod } 26)$$

3. The cipher text message is: $c1,c2...cn$

Decryption:

1. Calculate k^{-1}

2. For each pair of cipher text find a plaintext by:

$$k^{-1}\begin{pmatrix}c_1\\c_2\end{pmatrix}=\begin{pmatrix}p_1\\p_2\end{pmatrix}(\text{mod}\,26)$$

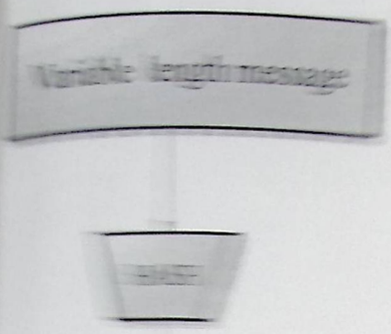
$$k^{-1}\begin{pmatrix}c_3\\c_4\end{pmatrix}=\begin{pmatrix}p_3\\p_4\end{pmatrix}(\text{mod}\,26)$$

.....

$$k^{-1}\begin{pmatrix}c_{n-1}\\c_n\end{pmatrix}=\begin{pmatrix}p_{n-1}\\p_n\end{pmatrix}(\text{mod}\,26)$$

5. PROPOSED MODEL

The main point of one-way hash function is that any encrypted text cannot be decrypted. From this point ,we need to choose the noninvertible matrix from the hill cipher to use it inside the practical one- way hash algorithm. First we take non-invertible matrix, multiply it by plaintext as column vector with modular


Variable length message
Hash

value n to generate the hash value H. The sender of the message calculates the hash value or message digest by using the model. After that the message and the hash value are sent to the receiver who makes the same calculations by using the model to generate a message hash value. Finally, the receiver compares between the messages digest from the sender and the hash value that he calculated.

5.1 Mathematical model

$H(V) = V \times R \bmod N$.

$H(V)$ = hash value generated.

V =plaintext message as column vector.

R = non-invertible matrix.

N = modular value.

We use the R as a non-invertible matrix that cannot be reversed, which is used to generate hash value using this formula:

$$H(V) = V \times R \bmod N.$$

R cannot be reversed, If we Calculate the determinant of this matrix d where $d = |R|$, then doesn't relatively prime to N, so R^{-1} doesn't exist and we can't calculate the value of $(H(V))^{-1}$.

Rushdi A. Hamamreh and Mousa Farajallah in their research paper "Design of a robust cryptosystem algorithm for non-invertible matrices based on Hill cipher" have proved that if the non-invertible matrix was used then the encrypted text can never be decrypted [5] and this what we need in our model in order to this satisfy the one way property .

5.2 Proof of practical one way property hash algorithm requirements:

5.2.1 *Applied To Any Size of Data*

For any input data v, let the square matrix has x dimension, then the system can convert the input data into vector(s) of x length, so if the number of integers consists the input vector less than x , then the system will make padding into vector v to make the new model applying to any size of data, but if the number of integers consists the input vector v more than x , the new model will make more than one round in order to applying to any size of data.

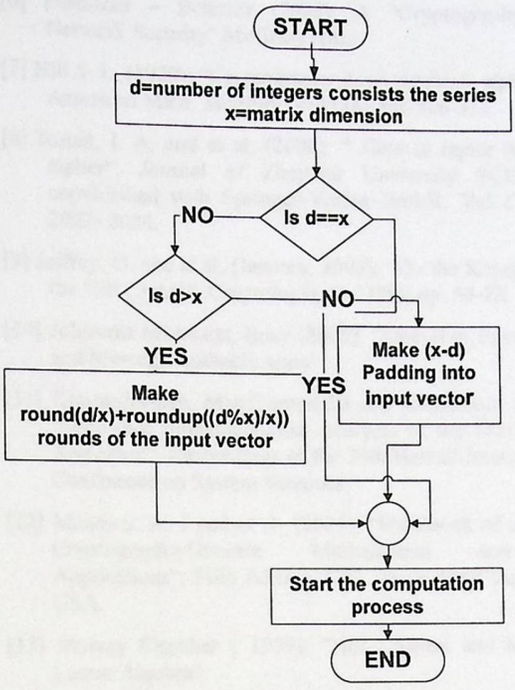


Figure2: steps to make new model applying to any data size

5.2.2 *H Produces a Fixed-Length Output*

The new model will be able to process an arbitrary-length message into a fixed-length output. this can be achieved by dividing up the input vector into a series of equally sized x vector(s) from a previous point, and operating on them in sequence.

5.2.3 *H Relatively Easy To Compute for Any Given X*

This property is easy to be found in any mathematical model If and only if the input is known and that makes the output easy to be calculated, and the new model has all the needed input parameters in order to calculate the output.

5.2.4 *One-Way Property*

The new model based on the following mathematical equation $H(V) = V \times R \bmod N$, where V is the input vector at any round, R is the non-invertible matrix, and N is the modular value of the system, if any user has $H(V)$, and let us assume also he has R and N then he can only formulate the following model $V = H(V) \times R^{-1} \bmod N$, but since R is not invertible matrix he can't solve this equation, so the proposed model is one way function.

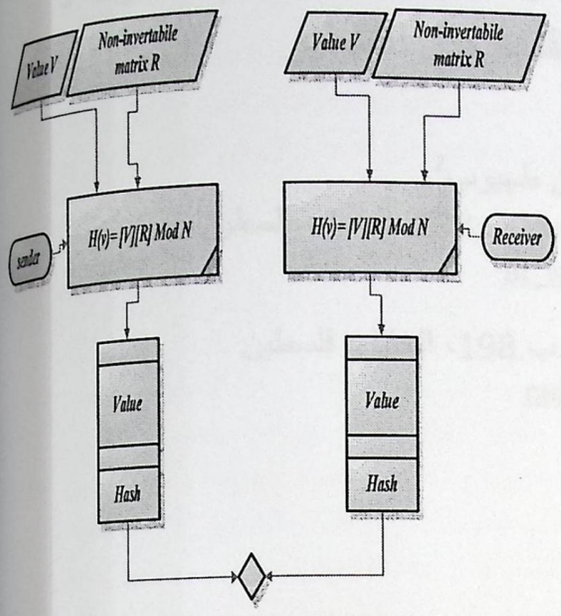


Figure3: practical one way hash algorithm

6. CONCLUSIONS AND FUTURE WORK

This paper proposes a technique to design a practical one-way hash algorithm by using non-invertible matrix that cannot be reversed to produce hash value. We proof the four requirements that a practical one way hash algorithm need. In Future works we need to design an algorithm that convert any invertible matrix into non-invertible one, and then design one way hash algorithm to generate hash value.

7. REFERENCES

[1] Bibhudendra. A, and et al, (2006): " Novel Methods Of Generating Self-Invertible Matrix For Hill Cipher Algorithm ". International Journal of Security, Vol (1), pp:14-21.

[2] Charlie, O, and Behzad, S, (2007): "A Parallel Algorithm for determining the inverse of a matrix for use in block cipher encryption/decryption". The Journal of Supercomputing, Vol (39), pp: 113-130.

[3] Chu-Hsing, L, and et al, (2004): " Comments On Saeednia's Improved Scheme For The Hill Cipher". Journal of the Chinese Institute of Engineering, Vol (27), pp: 743- 746.

[4] DaniloGligoroski†, Smile Markovski†† and Svein J. Knapskog†(2006):"A Secure Hash Algorithm with only 8 Folded SHA-1 Steps", IJCSNS International Journal of 194 Computer Science and Network Security, VOL.6 No.10

[5] Farajallah. M and Hamamreh. R (2009) " Self Generating Multi Keys Cryptosystem Model for Non-Invertible Matrices based on Hill Cipher ", Security and Management 2009, IET SAM09 Conference, Las Vegas, Nevada, USA, 665-672.

[6] Forouzan – Behrouz (2008) .A "Cryptography and Network Security",McGraw Hill..

[7] Hill S. L, (1929): "Cryptography in an algebraic alphabet". American Math. Monthly, Vol (36), pp: 306-312.

[8] Ismail, I. A, and et al, (2006): " How to repair the Hill cipher". Journal of Zhejiang University SCIENCE, copublished with Springer-Verlag GmbH, Vol (7), pp: 2022- 2030.

[9] Jeffrey, O, and et al, (January, 2005): "On the Keyspace of the Hill Cipher". Cryptologia, Vol (29), pp: 59-72.

[10] Johannes Mittmann, (may ,2005): "One-Way Encryption and Message Authentication".

[11] KimmoJrvinen, MattiTommiska and JormaSkytt (2005): "Hardware Implementation Analysis of the MD5 Hash Algorithm", Proceedings of the 38th Hawaii International Conference on System Sciences.

[12] Menezes. A. J and et al, (2001): "Handbook of Applied Cryptography-Discrete Mathematics and Its Applications", Fifth Edition. CRC Press, New York, Inc. USA.

[13] Murray Eisenber (1999): "Hill Ciphers and Modular Linear Algebra".

[14] Rushdi A. Hamamreh, Mousa Farajallah(2009), "Design of a Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher", International Journal of Computer Science and Network Security; pp 11-16.

[15] Victor Shoup, (2000): " A Composition Theorem for Universal One-WayHash Functions".

[16] William Stallings (2006.), "Cryptography and Network Security Principles and Practices", Prentice Hall.

[17] Yi-Shiung. Y, and et al, (1991): "A New Cryptosystem Using Matrix Transformation". Proceedings of IEEE International Canahan Conference on Security Technology, 1-3 – October – 2008, Taipei, Taiwan pp: 131-138.

[18] Yi-Shiung. Y, and et al, (1993): " HAVAL — A One-Way Hashing Algorithm with Variable Length of Output". Appeared in "Advances in Cryptology — AUSCRYPT'92," Lecture Notes Computer Science, Vol.718, pp.83-104, Springer-Verlag

Mousa Farajallah received the B.S. degree in Computer Systems Engineering from Palestine Polytechnic University (PPU) in 2006, and his M.S degree from AL-Quds University in Jerusalem, now he is lecturer at College of Engineering and Technology at PPU, his research interests are in cryptography and algorithms.

Mohammed Abutaha received the B.S. degree in information Technology from Palestine Polytechnic University (PPU) in 2007, and his M.S degree from PPU University in Hebron, now he is lecturer at College of Applied Professions at PPU, his research interests are in cryptography and information security applications.

© Communications of the Arab Computer Society, Vol. 4 No.1, August, 2011

خوارزمية البعثرة \التجزئة العملية ذات الاتجاه الواحد باستخدام مصفوفة لا معكوس لها اعتماداً على تقنية هيل للتشفير

محمد ابو طه[1] ،رضوان طهبوب[2]
[1]جامعة بوليتكنيك فلسطين، كلية المهن التطبيقية، ص.ب.198، الخليل، فلسطين
m_abutaha@ppu.edu

[2]جامعة بوليتكنيك فلسطين، كلية الهندسة، ص.ب.198، الخليل، فلسطين
radwant@ppu.edu

الخلاصة:

ان قضية امن المعلومات تعد من اهم قضايا العصر الحديث حيث ان الاعتماد الحالي على الانترنت اصبح في جميع مجالات الحياة ;فأصبحت قضية حماية هذه المعلومات من الاختراق قضية مهمة .

من المعروف ان خوارزمية التشفير ذات الاختزال تعمل بالاتجاه الواحد ولا يمكن الوصول الى اصلها او استرجاعه ويمكن تصميم خوارزمية مشابهة تعمل بالاتجاه الواحد وذلك باستخدام احدى الطرق المشهورة في عالم التشفير المسمية بخوارزمية هيل والتي تم اختراعها في سنة 1929 . تقنية هيل تعتمد على التشفير باستخدام مفتاح على شكل مصفوفة واستخدام معكوس هذه المصفوفة كمفتاح لفك التشفير .وتعتبر من الخوارزميات المهمة التي قدمت حلا لمشكلة تحليل تكرارية الحروف داخل نص معين باستخدام التشفير لأكثر من ثلاثة حروف دفعة واحدة.

معكوس المصفوفة المستخدم في تقنية هيل غير متاح دائما ومن هذا المنطلق يمكن استخدام المصفوفات التي لا يوجد لها معكوس لاشتقاق تقنية جديدة لخوارزمية بعثرة عملية ذات الاتجاه الواحد .

الخوارزمية المقترحة ستقدم نموذجا افضل لإيجاد قيمة مبعثرة حيث يتميز هذا النظام بقدرته على مواجهة خطر هجوم القاموس وذلك عن طريق اضافة القيمة الملحية على البيانات باستخدام مولد الارقام العشوائية مما يجعل من الصعب اعداد قاموس من القيم المبعثرة .في هذه الورقة تم عمل مقارنة بين النموذج المقترح وانظمة MD5,SHA1,SHA512.

الكلمات الجوهرية: خوارزمية البعثرة ذات الاتجاه الواحد، قاموس الهجوم، التصادم القوي.

1. مقدمة:

عملية التشفير هي عملية تجمع بين علم الرياضيات وعلم الحاسوب، هي العلم والمقدرة على حماية البيانات من الاختراق حيث تستخدم في معظم المجالات والتطبيقات التقنية والعملية .الطلب على عملية التشفير في تزايد لحماية البيانات لكن ما يتم حفظه وتأمينه اليوم من الممكن ان يتعرض للاختراق في الغد .

يشمل علم التشفير مجموعة من الخوارزميات والتقنيات لتحويل البيانات الى شكل اخر بحيث تظهر محتوياتها بشكل غير مقروء وغير قابل للتفسير لأي شخص ليس لديه الصلاحيات للقراءة او الكتابة على هذه البيانات.

الهدف الرئيسي من استخدام خوارزميات التشفير حماية المعلومات والبيانات بهدف تحقيق الخصوصية، التكاملية وامكانية الوصول للمصادر والخدمات التي يقدمها نظام المعلومات. هناك مجموعة من المخاطر التي من الممكن أن تؤذي نظم المعلومات الحاسوبية ومنها ما يكون موجه لخوارزمية التشفير نفسها مثال على ذلك خوارزمية التشفير التماثلية قد تواجه خطر المهاجمين لمحتويات الخوارزمية وتكوينها او خصائصها و معرفة أجزاء من الرسالة الاصلية التي يريد المستخدم لهذه الخوارزمية استخدامها مما يؤدي الى معرفة الرسالة كاملة اوقد يعتمد المهاجمين على تجربة مجموعة من مفاتيح التشفير المحتملة على جزء النص المشفر ومن الممكن في هذه الحالة ان يصل الى الرسالة الاصلية قبل التشفير.

مع التزايد الكبير في المخاطر التي قد تضر بالبيانات تم وضع مجموعة من القواعد والتقنيات والمضادات لحماية انظمة المعلومات . المضادات هي أي طريقة أو وسيلة ممكن استخدامها لمنع أي هجوم يهدد امن البيانات والمعلومات. المضادات ممكن ان تكون بمنع الهجوم على البيانات إن أمكن ، أو اكتشاف الهجوم وعملية الاختراق في حال فشلت عملية منع الهجوم ثم ارجاع النظام الى الوضع الطبيعي قبل عملية الاختراق او الهجوم[15].

يواجه النظام المقترح مشكلة الحسابات العملية التي يقد يستهلكها في ضرب المصفوفات الكبيرة ولكن هذه المشكلة قد تنعكس كفائدة لتعقيد التوصل الى اصل القيمة المبعثرة.

1.1. خوارزميات التشفير التماثلية وغير التماثلية:

عملية التشفير التماثلية تعتمد على استخدام مفتاح واحد في التشفير من طرف المرسل وفي فك التشفير من طرف المستقبل اما عمليات التشفير غير التماثلية فتعتمد على استخدام مفتاحين مختلفين في التشفير حيث يتم استخدام مفتاح عام للتشفير في طرف المرسل ومفتاح اخر خاص لفك التشفير في طرف المستقبل [16].

2. خوارزميات التجزئة أو البعثرة :

2.1. التعريف، الوصف، التطبيقات:

الخوارزمية التي تقوم بتغيير النص أو الرسالة الى مجموعة من الرموز والأحرف لحمياتها من امكانية التعرف على اصلها أو استرجاعها من قبل المهاجمين للنظام. خاصية الاتجاه الواحد التي تحققها خوارزميات التجزئة تجعل من الصعب بل من المستحيل استرجاع النص الاصلي من القيمة الناتجة عن هذه الخوارزمية. ممكن استخدام هذا النوع من الخوارزميات في إنشاء التوقيع الالكتروني الذي يحقق خاصية الأصالة للبيانات أي أنها من مصدر موثوق .خوارزميات البعثرة تتكون من إجراء يقوم بتحويل البيانات الضخمة الى بيانات صغيرة أو من بيانات ذات حجم متغير إلى بيانات ذات حجم ثابت كما هو مبين في الشكل رقم 1 .حيث أن البيانات الناتجة عن هذا النوع من الخوارزميات تسمى رموز البعثرة[5].

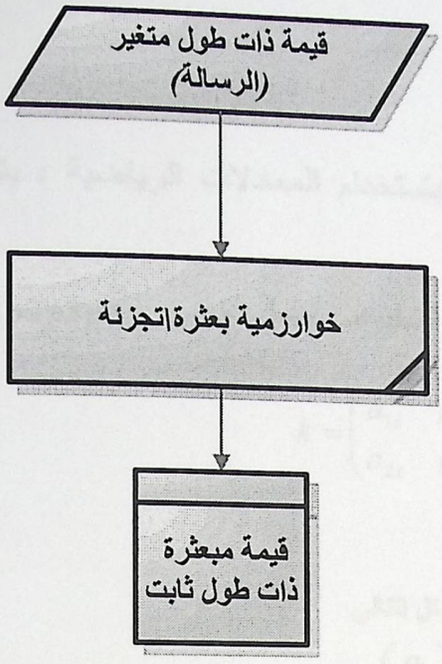
تعد خوارزميات البعثرة ذات الاتجاه الواحد من أهم الخوارزميات المستخدمة في علم التشفير حيث تستخدم في حل الكثير من المشاكل المتعلقة بالتكاملية والأصالة حيث يعتبر بديل عن خوارزمية التشفير رمز المصادقة للرسالة حيث انه لا يحتاج لوجود مفتاح في عملية التشفير حيث يكون الناتج عبارة عن قيمة مبعثرة لا يمكن استرجاع اصلها [8].

2.2. متطلبات خوارزميات البعثرة ذات الاتجاه الواحد:

تحتاج أي خوارزمية اختزال باتجاه واحد الى مجموعة من المتطلبات حتى يقوم بعمله ودوره بشكل صحيح[16]:

- يجب ان يكون صالح لأي حجم من البيانات: بحيث يتم ادخال بيانات بأحجام متغيرة ثم يتم تحويلها إلى حجم ثابت.
- يقوم بإنتاج مجموعة من البيانات بحجم ثابت
- $H(X) = h$ ممكن حسابها لأي قيمة .
- خاصية الاتجاه الواحد : من الصعب ايجاد قيمة x $H(X) = h$.
- ممانعة التصادم الضعيف: من الصعب ايجاد $H(y) = H(x)$ حيث $(y \neq x)$.
- ممانعة التصادم القوي: من الصعب ايجاد $H(y) = H(x)$ لأي قيمة (x, y).

اذا تم تحقيق أول ثلاثة متطلبات فأن خوارزمية البعثرة تكون خوارزمية عملية وفي حال تم إضافة الخاصية الرابعة ففي هذه الحالة تسمى خوارزمية البعثرة ذات الاتجاه الواحد وفي حال تم اضافة المتطلب الخامس والسادس تسمى خوارزمية البعثرة الامنة ذات الاتجاه الواحد[16].



شكل 1 : خوارزمية البعثرة ذات الاتجاه الواحد.

3. تقنية هيل للتشفير:

هي من أشهر خوارزميات التشفير التماثلية منذ اختراعها والتي تستخدم في تشفير البيانات ومنع الأشخاص غير المسموح لهم من الاطلاع على البيانات. بحيث تعتمد على استخدام مفتاح التشفير على شكل مصفوفة ومعكوس هذا المفتوح لفك التشفير والمعكوس هنا ليس المعكوس العادي للمصفوفة وانما المعكوس نسبة الى أي رقم معين[1].

ومن الممكن عدم وجود معكوس لبعض المصفوفات عندما تكون محددة المصفوفة ليست أولية نسبة إلى ذلك الرقم[3].

تعتبر تقنية هيل في التشفير من التقنيات الجيدة لاعتمادها على الجبر الخطي في الحسابات اضافة إلى السرعة العالية والانتاجية في التشفير[4],[2].

تقنية هيل تحقق الخصائص التي يجب أن يتمتع بها أي نظام تشفير متمز:

- **خاصية النشر:** أي تغيير على النص الاصلي للرسالة قبل التشفير سيؤدي الى تغيير في النص بعد التشفير.
- **خاصية الارباك:** المفتاح المستخدم في التشفير غير مرتبط في عملية فك التشفير.

3.1 مثال على تقنية هيل

باستخدام مفتاح تشفير 2X2 باستخدام المعادلات الرياضية ، بشرط مفتاح التشفير مصفوفة قابلة للانعكاس بقيمة باقي القسمة 26.

على اعتبار أن النص الأصلي المراد تشفيره : $p1p2p3p4.....pn-1pn$

وعلى اعتبار أن مفتاح التشفير $k=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$

عملية التشفير:

- **تحويل النص الاصلي على الشكل التالي**

$$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix}\begin{pmatrix} p_3 \\ p_4 \end{pmatrix}\begin{pmatrix} p_5 \\ p_6 \end{pmatrix}....\begin{pmatrix} p_{n-1} \\ p_n \end{pmatrix}$$

- **ضرب المقدار السابق بالمفتاح لإيجاد النص المشفر**

$c1c2.....cn$

$$k\begin{pmatrix} p_1 \\ p_2 \end{pmatrix}=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} p_1 \\ p_2 \end{pmatrix}=\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}(\text{mod } 26)$$

$$k\begin{pmatrix} p_3 \\ p_4 \end{pmatrix}=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} p_3 \\ p_4 \end{pmatrix}=\begin{pmatrix} c_3 \\ c_4 \end{pmatrix}(\text{mod } 26)$$

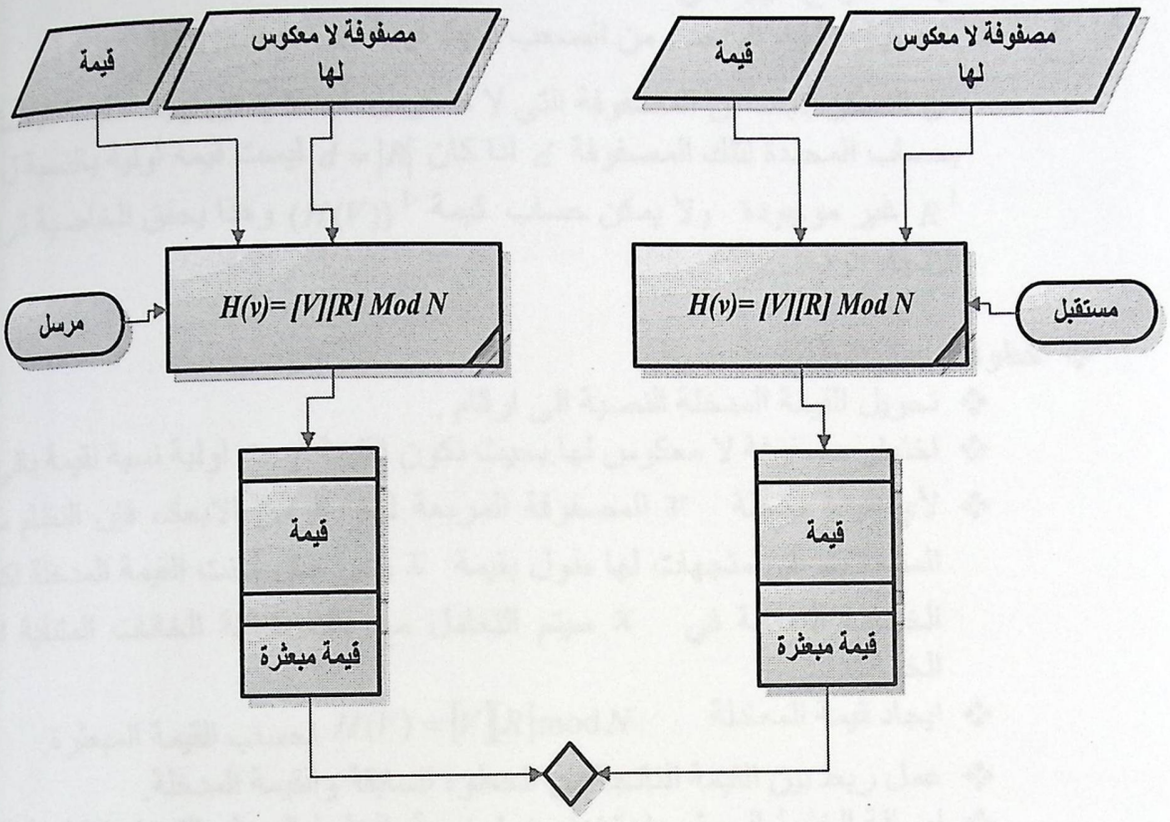
.....

$$k\begin{pmatrix} p_{n-1} \\ p_n \end{pmatrix}=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} p_{n-1} \\ p_n \end{pmatrix}=\begin{pmatrix} c_{n-1} \\ c_n \end{pmatrix}(\text{mod } 26)$$

عملية فك التشفير:

- **حساب معكوس المفتاح** k^{-1}
- **ايجاد النص الاصلي من النص المشفر**

من الممكن إثبات إن المصفوفة التي لا معكوس لها لا يمكن ايجاد القيمة الاصلية منها وذلك بحساب المحدد لتلك المصفوفة d اذا كان $d = |R|$ ليست قيمة أولية بالنسبة ل N وبذلك R^{-1} غير موجودة ولا يمكن حساب قيمة $(H(V))^{-1}$ وهذا يحقق الخاصية الرابعة ذات الاتجاه الواحد.



شكل 2: خوارزمية البعثرة العملية ذات الاتجاه الواحد

❖ **اثبات المتطلبات:**

- يجب ان يكون صالح لأي حجم من البيانات: بحيث يتم ادخال بيانات بأحجام متغيرة ثم يتم تحويلها إلى حجم ثابت: لأي قيمة مدخلة V المصفوفة المربعة لها x من الابعاد، فإن النظام سيقوم بتحويل المدخلات الى متجهات لها طول بقيمة x وفي حال كانت القيمة المدخلة اكبر او اقل من الخانات المتاحة في x سيتم التعامل مع ذلك بتعبئة الخانات المتبقية او زيادة عدد الخانات.

- يقوم بإنتاج مجموعة من البيانات بحجم ثابت
ويتم ذلك بتقسيم المدخلات الى اجزاء تتوافق مع عمل نظام البعثرة.

- $H(X)=h$ ممكن حسابها لأي قيمة .

ويتم تحقيقها بسهولة بوجود نموذج رياضي مدخلاته معلومة فمن المؤكد وجود مخرجات لهذا النموذج الرياضي

- خاصية الاتجاه الواحد : من الصعب ايجاد قيمة x . $H(X)=h$

من الممكن إثبات إن المصفوفة التي لا معكوس لها لا يمكن ايجاد القيمة الاصلية منها وذلك بحساب المحددة لتلك المصفوفة d اذا كان $d=|R|$ ليست قيمة أولية بالنسبة ل N وبذلك R^{-1} غير موجودة ولا يمكن حساب قيمة $(H(V))^{-1}$ وهذا يحقق الخاصية الرابعة ذات الاتجاه الواحد.

❖ خطوات عمل النظام:

- ❖ تحويل القيمة المدخلة النصية الى ارقام .
- ❖ اختيار مصفوفة لا معكوس لها.بحيث تكون القيمة ليست اولية نسبة لقيمة باقي القسمة N.
- ❖ لأي قيمة مدخلة v المصفوفة المربعة لها x من الابعاد، فإن النظام سيقوم بتحويل المدخلات الى متجهات لها طول بقيمة x وفي حال كانت القيمة المدخلة اكبر او اقل من الخانات المتاحة في x سيتم التعامل مع ذلك بتعبئة الخانات المتبقية او زيادة عدد الخانات.
- ❖ ايجاد قيمة المعادلة $H(V)=[V][R] \bmod N$ لحساب القيمة المبعثرة.
- ❖ عمل ربط بين القيمة الناتجة من الخطوة السابقة والقيمة المدخلة.
- ❖ اضافة الخليط المبعثر باستخدام خوارزمية الخليط المبعثر التي تم انشاؤها باستخدام الماتلاب.

❖ مثال على استخدام خوارزمية البعثرة المقترحة :

V="Palestine"

$$R=\begin{pmatrix}7 & 3\\ 2 & 2\end{pmatrix}$$

N=12

d(R)=8

لقيمة ليست اولية نسبة لقيمة باقي القسمة N وبذلك فإن المصفوفة لا معكوس لها.

"Polestine" is :6 5 10 9 6 1 10 3 10 7 *بعد عمل خطوة تعبئة الخانات الناقصة لتصبح عشر خانات بدل تسعة*

القيمة المبعثرة الناتجة من الخوارزمية بعد تطبيق خطوات الخوارزمية كاملة:

- $H(X)=h$ ممكن حسابها لأي قيمة .

ويتم تحقيقها بسهولة بوجود نموذج رياضي مدخلاته معلومة فمن المؤكد وجود مخرجات لهذا النموذج الرياضي

- خاصية الاتجاه الواحد : من الصعب ايجاد قيمة x . $H(X)=h$.

من الممكن إثبات إن المصفوفة التي لا معكوس لها لا يمكن ايجاد القيمة الاصلية منها وذلك بحساب المحددة لتلك المصفوفة d اذا كان $d=|R|$ ليست قيمة أولية بالنسبة ل N وبذلك R^{-1} غير موجودة ولا يمكن حساب قيمة $(H(V))^{-1}$ وهذا يحقق الخاصية الرابعة ذات الاتجاه الواحد.

❖ **خطوات عمل النظام:**

- ❖ تحويل القيمة المدخلة النصية الى ارقام .
- ❖ اختيار مصفوفة لا معكوس لها.بحيث تكون القيمة ليست اولية نسبة لقيمة بافي القسمة N.
- ❖ لأي قيمة مدخلة v المصفوفة المربعة لها x من الابعاد، فإن النظام سيقوم بتحويل المدخلات الى متجهات لها طول بقيمة x وفي حال كانت القيمة المدخلة اكبر او اقل من الخانات المتاحة في x سيتم التعامل مع ذلك بتعبئة الخانات المتبقية او زيادة عدد الخانات.
- ❖ ايجاد قيمة المعادلة $H(V)=[V][R]\bmod N$ لحساب القيمة المبعثرة.
- ❖ عمل ربط بين القيمة الناتجة من الخطوة السابقة والقيمة المدخلة.
- ❖ اضافة الخليط المبعثر باستخدام خوارزمية الخليط المبعثر التي تم انشاؤها باستخدام الماتلاب.

❖ مثال على استخدام خوارزمية البعثرة المقترحة :

V="Palestine"

$$R=\begin{pmatrix}7 & 3\\ 2 & 2\end{pmatrix}$$

N=12

d(R)=8

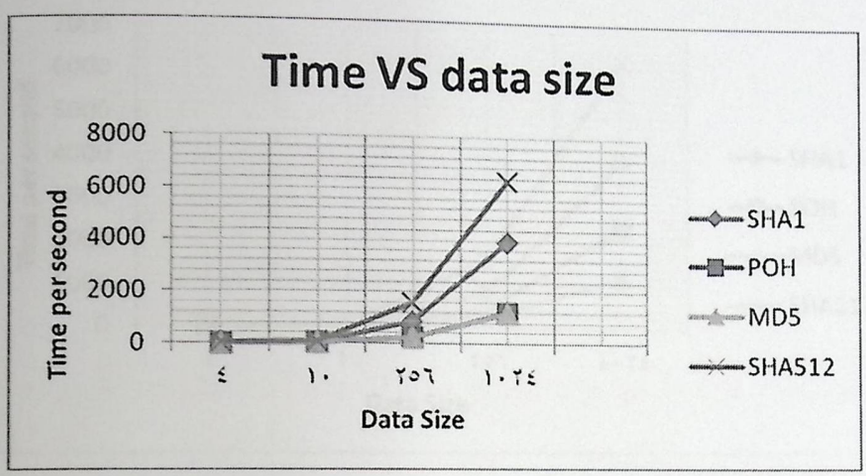
لقيمة ليست اولية نسبة لقيمة باقي القسمة N وبذلك فإن المصفوفة لا معكوس لها.

بعد عمل خطوة تعبئة الخانات الناقصة لتصبح عشر خانات بدل تسعة 7 10 3 10 1 6 9 10 5 6: "Palestine" is

القيمة المبعثرة الناتجة من الخوارزمية بعد تطبيق خطوات الخوارزمية كاملة:

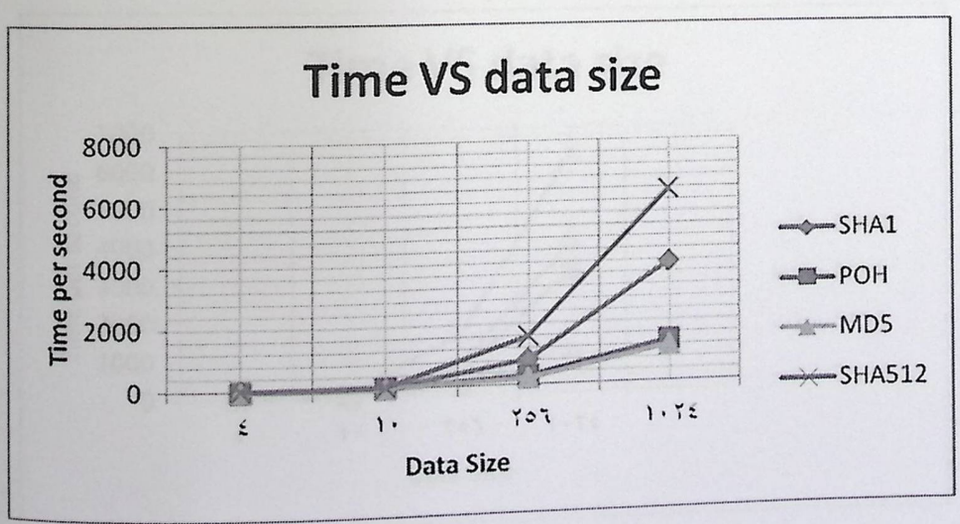
6. مقارنة مع الانظمة الاخرى:

تم عمل مقارنة بين النظام المقترح وثلاثة انظمة اخرى مشهورة من حيث الوقت وحجم المصفوفة وحجم بيانات متغير



شكل3: مقارنة بين النظام المقترح (POH)، MD5، SHA1، SHA512 على حجم مصفوفة 1x1

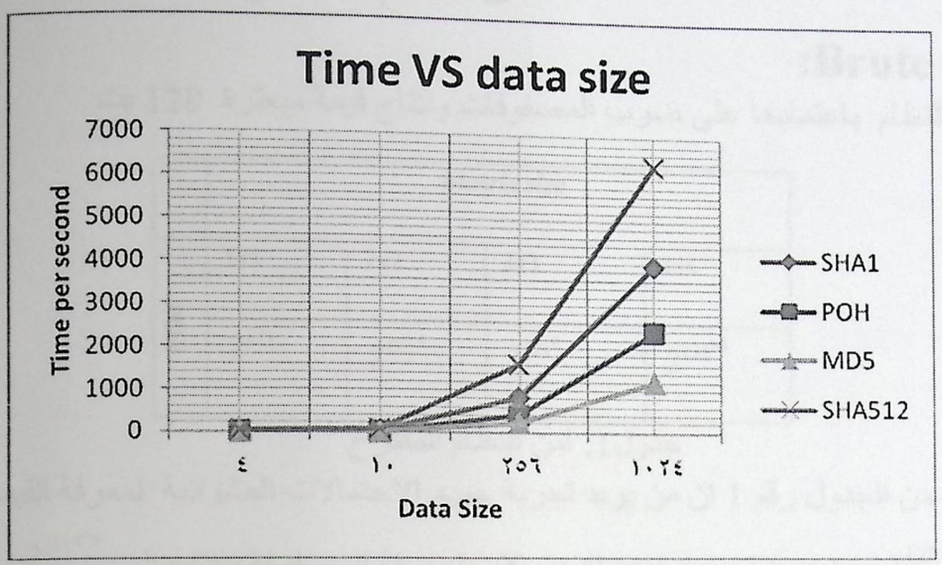
كما نشاهد في الشكل 3 ان سرعة النظام الجديد تعادل سرعة MD5 عندما يكون حجم المصفوفة 1x1 وايضا اسرع من SHA512, SHA1.



شكل4: مقارنة بين النظام المقترح (POH)، MD5، SHA1، SHA512 على حجم مصفوفة 3x3

كما نشاهد في الشكل 4 ان سرعة النظام الجديد تعادل سرعة MD5 عندما يكون حجم المصفوفة 3x3

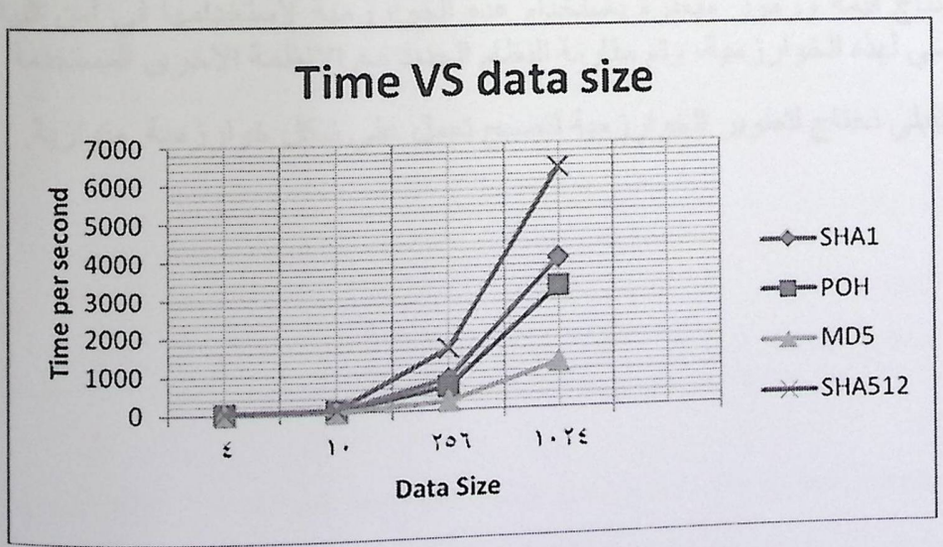
وايضا اسرع من SHA512, SHA1



شكل5: مقارنة بين النظام المقترح (POH)، MD5، SHA1، SHA512 على حجم مصفوفة 5x5

كما نشاهد في الشكل 5 ان سرعة النظام الجديد اقل من MD5 عندما يكون حجم المصفوفة 5x5

ويكون اسرع من SHA512, SHA1



شكل6: مقارنة بين النظام المقترح (POH)، MD5، SHA1، SHA512 على حجم مصفوفة 7x7

كما نشاهد في الشكل 6 ان سرعة النظام الجديد اقل من MD5 عندما يكون حجم المصفوفة 7x7

ويكون اسرع من SHA1 ,SHA512

7. أمن النظام المقترح ضد هجوم القوة الغاشمة

Brute force:

تكمن قوة النظام باعتمادها على ضرب المصفوفات وانتاج قيمة مبعثرة 128 بت

	Matrix size			
POH	1x1	2x2	3x3	4x4
	$2^{(7)}$	$2^{(7)^4}$	$2^{(7)^9}$	$2^{(7)^{16}}$

جدول1: امن النظام المقترح

كما نلاحظ من الجدول رقم 1 ان من يريد تجربة جميع الاحتمالات العشوائية لمعرفة القيمة المبعثرة يحتاج الى 2^{128} ومن يريد تجربة جميع الاحتمالات لمعرفة المصفوفة يحتاج الى $2^{(7)^{n*n}}$

8. الخاتمة:

في هذه الورقة تم تقديم نموذج وتقنية لإيجاد خوارزمية تجزئة عملية ذات اتجاه واحد باستخدام مصفوفة لا معكوس لها لإنتاج قيمة ورموز مبعثرة باستخدام هذه الخوارزمية لاستخدامها في أمن البيانات. وتم تقديم النموذج الرياضي لهذه الخوارزمية، وتم مقارنة النظام الجديد مع الانظمة الاخرى المستخدمة .

في العمل المستقبلي نحتاج لتطوير الخوارزمية لتصبح تعمل على شكل خوارزمية متوازية.

9. المصادر والمراجع:

[هيل ، 1929]

[1] Hill S. L, (1929): " **Cryptography in an algebraic alphabet** ". American Math. Monthly, Vol (36), pp: 306-312.

[اسماعيل ومن معه ، 2006]

[2] Ismail, I. A, and et al, (2006): " **How to repair the Hill cipher** ". Journal of Zhejiang University SCIENCE, co-published with Springer-Verlag GmbH, Vol (7), pp: 2022-2030.

[مينز ومن معه ، 2001]

[3] Menezes. A. J and et al, (2001): **"Handbook of Applied Cryptography-Discrete Mathematics and Its Applications"**, Fifth Edition. CRC Press, New York, Inc. USA.

[ياي شينج ومن معه، 1991]

[4] Yi-Shiung. Y, and et al, (1991): **"A New Cryptosystem Using Matrix Transformation "**. Proceedings of IEEE International Canahan Conference on Security Technology, 1-3 – October – 2008, Taipei, Taiwan pp: 131-138.

[ياي شينج ومن معه،1993]

[5] Yi-Shiung. Y, and et al, (1993): " **HAVAL — A One-Way Hashing Algorithm with Variable Length of Output** ". Appeared in "Advances in Cryptology — AUSCRYPT'92," Lecture Notes Computer Science, Vol.718, pp.83-104, Springer-Verlag

[شو هسنج ومن معه ، 2004]

[6] Chu-Hsing, L, and et al, (2004): " **Comments On Saeednia's Improved Scheme For The Hill Cipher** ". Journal of the Chinese Institute of Engineering, Vol (27), pp: 743-746.

[جيفري ومن معه ، 2005]

[7] Jeffrey, O, and et al, (January, 2005): **"On the Keyspace of the Hill Cipher"**. Cryptologia, Vol (29), pp: 59-72.

[جوهانس ماتمان، 2005]

[8] Johannes Mittmann, (may ,2005): " **One-Way Encryption and Message Authentication"**.

[بيبهاندرا ومن معه ، 2006]

[9] Bibhudendra. A, and et al, (2006): " **Novel Methods Of Generating Self-Invertible Matrix For Hill Cipher Algorithm** ". International Journal of Security, Vol (1), pp: 14-21.

[شارلي ومن معه ، 2007]

[10] Charlie, O, and Behzad, S, (2007): " **A Parallel Algorithm for determining the inverse of a matrix for use in block cipher encryption/decryption"**. The Journal of Supercomputing, Vol (39), pp: 113-130.

[فيكتور شوب ، 2000]

[11] Victor Shoup, (2000): " **A Composition Theorem for Universal One-WayHash Functions**".

[كيمو جرفينين ومن معه ، 2005]

[12] Kimmo Jrvinen, Matti Tommiska and Jorma Skytt (2005): **"Hardware Implementation Analysis of the MD5 Hash Algorithm"**, Proceedings of the 38th Hawaii International Conference on System Sciences.

[دانيلو جليحورستك ومن معه ، 2006]

[13] *Danilo Gligoroski†, Smile Markovski†† and Svein J. Knapskog†*(2006):**"A Secure Hash Algorithm with only 8 Folded SHA-1 Steps"**, IJCSNS International Journal of 194 Computer Science and Network Security, VOL.6 No.10

[موراي اسنبرج، 1999]

[14] Murray Eisenber (1999): **" Hill Ciphers and Modular Linear Algebra"**.

[افورزان بهروز، 2006]

[15] Forouzan – Behrouz (2008) .A **"Cryptography and Network Security"**,McGraw Hill..

[وليم ستالينج، 2006]

[16] William Stallings (2006.), **"Cryptography and Network Security Principles and Practices"**, Prentice Hall.

10. الخلاصة باللغة الانجليزية

Practical one way hash algorithm using non-invertible matrix based on hill cipher technique

Keywords: *One way hash algorithm, dictionary attack, collision resistance .*

Nowadays, Cryptography plays a major role in protecting technology applications information. Hash function is a well-defined procedure that converts a large data into a small one, The returned value from hash function is called hash code, It is well known that Hash algorithm works in one way and cannot be reversed, In this paper, The new one way hash algorithm will be designed by using two steps. Firstly, we will convert the input data into matrix system by using all necessary conversions to generate the initial hash value . Secondly, use the output of the first step to make a digest for these data and finally generate the secure hash value. The first step can never be reversed in the new proposed system since the matrix system requires the inverse of the matrix to retrieve the input data, and the first round of the proposed algorithm depends on the multiplication of non-invertible matrix with the plaintext message in column vector. It's a new method to make hash value that depends on matrix multiplication.

11. المصطلحات:

انجليزي	عربي
One way hash algorithm	خوارزمية البعثرة ذات الاتجاه الواحد
Non-invertible matrix	مصفوفة لا معكوس لها
Hill cipher technique	تقنية هيل
Diffusion	خاصية النشر
Confusion	خاصية الارباك
Message authentication code	رمز المصادقة للرسالة
Hash value	قيمة مبعثرة
Weak collision resistance	ممانعة التصادم الضعيف
Strong collision resistance	ممانعة التصادم القوي
Digital signature	التوقيع الالكتروني
Authenticity of data	اصالة البيانات
Message digest	خليط مبعثر للرسالة

Survey Paper: Cryptography Is The Science Of Information Security

Mohammed AbuTaha
College of Administrative Sciences and Informatics
Palestine Polytechnic University
Hebron, Palestine
m_abutaha@ppu.edu

Mousa Farajallah
College of Engineering and Technology
Palestine Polytechnic University
Hebron, Palestine
mousa_math@ppu.edu

Radwan Tahboub
College of Engineering and Technology
Palestine Polytechnic University
Hebron, Palestine
radwant@ppu.edu

Mohammad Odeh
IT and Communications Dept
Al-Quds Open University
Hebron, Palestine
mhmdodeh@qou.edu

Abstract

Cryptography in the past was used in keeping military information, diplomatic correspondence secure and in protecting the national security. However, the use was limited. Nowadays, the range of cryptography applications have been expanded a lot in the modern area after the development of communication means; cryptography is essentially required to ensure that data are protected against penetrations and to prevent espionage. Also, cryptography is a powerful mean in securing e-commerce. Cryptography is used to ensure that the contents of a message are confidentiality transmitted and would not be altered. Confidentiality means nobody can understand the received message except the one who has the decipher key, and data cannot be changed means the original information would not be changed or modified; this is done when the sender includes a cryptographic operation called a hash function in the original message. A hash function is a mathematical representation of the information, when any information arrives at its receiver; the receiver calculates the value of this hash function. If the receiver's hash function value is equivalent to the sender's, the integrity of the message is assured .

Keyword: Symmetric Encryption, A Symmetric Encryption ,Hash Algorithm, Caesar Table.

1. INTRODUCTION

Nowadays, cryptography plays a major role in protecting the information of technology applications. Information security is an important issue, for some applications. Have the top priority such as e-commerce, e-banking, e-mail, medical databases, and so many more, all of them require the exchange of private information. For example, let us consider a person named Alice a sender who wants to send a data message which has a length of m characters to a receiver called Bob. Alice uses an unsecure communication channel. Which could be a telephone line , computer network, or any other channel. If the message contains secret data, they could be intercepted and read by hackers. Also they may change or modify the message during its transmission in such a way that Bob would not be able to discover the change. In this survey a various ways of encryption is viewed and have been compared ,a lot of examples have been provided .

1.1 Cryptography Goals

By using cryptography many goals can be achieved, These goals can be either all achieved at the same time in one application, or only one of them, These goals are:

1. Confidentiality: it is the most important goal, that ensures that nobody can understand the received message except the one who has the decipher key.

2. Authentication: it is the process of proving the identity, that assures the communicating entity is the one that it claimed to be, This means that the user or the system can prove their own identities to other parties who don't have personal knowledge of their identities. (The primary form of host to host authentication on the Internet today is name-based or address-based; and both of them are notoriously weak).

3. Data Integrity: its ensures that the received message has not been altered in any way from its original form, This can be achieved by using hashing at both sides the sender and the recipient in order to create a unique message digest and compare it with the one that received.
4. Non-Repudiation: it is mechanism used to prove that the sender really sent this message, ,and the message was received by the specified party, so the recipient cannot claim that the message was not sent [2].

5. Access Control: it is the process of preventing an unauthorized use of resources. This goal controls who can have access to the resources, If one can access, under which restrictions and conditions the access can be occurred, and what is the permission level of a given access.

1.2 Basic Terminology of Cryptography

Computers are used by millions of people for many purposes. such as banking, shopping, military, student records, etc.... . Privacy is a critical issue in many of these applications, how are we need to make sure that an unauthorized parties cannot read or modify messages.

Cryptography is the transformation of readable and understandable data into a form which cannot be understood in order to secure data. cryptography refers exactly to the methodology of concealing the content of messages, the word cryptography comes from the Greek word "Kryptos", that means hidden, and "graphikos" which means writing [3].

The information that we need to hide, is called **plaintext** (P), It's the original text, It could be in a form of characters, numerical data, executable programs, pictures, or any other kind of information, The plaintext for example is the first draft of a message in the sender before encryption, or it is the text at the receiver after decryption.

The data that will be transmitted is called **cipher text** (C), it's a term refers to the string of "*meaningless*" data, or unclear text that nobody must understand, except the recipients. it is the data that will be transmitted Exactly through network, Many algorithms are used to transform plaintext into cipher text [4].

Cipher is the algorithm that is used to transform plaintext to cipher text, This method is called encryption or enciphers (encode), in other words, it's a mechanism of converting readable and understandable data into "*meaningless*" data, and it is represented as follows:

$$C = E_{(K)}(P) \quad (1)$$

Where $E_{(K)}$ is the encryption algorithm using key k.

The opposite of cipher mechanism is called **decipher (decode)** that is the algorithm which recovers the cipher text, this method is called decryption, in other words it's the mechanism of converting "*meaningless*" data into readable data.

$$P = D_{(K^{-1})}(C) \quad (2)$$

The Key is an input to the encryption algorithm, and this value must be independent of the plaintext, This input is used to transform the plaintext into cipher text, so different keys will yield different cipher text, In the decipher side, the inverse of the key will be used inside the algorithm instead of the key.

Computer security it's a generic term for a collection of tools designed to protect any data from hackers, theft, corruption, or natural disaster while allowing these data to be available to the users at the same time. One example of these tools is the A-vast antivirus program [1].

Network security refers to any activity designed to protect the usability, integrity, reliability, and safety of data during their transmission on a network, Network security deals with hardware and software, The activity can be one of the following anti-virus and anti-spyware, firewall, Intrusion prevention systems, and Virtual Private Networks [4].

Internet Security is measures and procedures used to protect data during their transmission over a collection of interconnected networks .while **information security** is about how to prevent attacks, and to detect attacks on information-based systems [2].

Cryptanalysis (code breaking) is the study of principles and methods of deciphering cipher text without knowing the key, typically this includes finding and guessing the secrete key, It's a complex process involving statistical analysis, analytical reasoning, math tools and pattern-finding, The field of both cryptography and cryptanalysis is called **cryptology** [4,15].
Symmetric encryption refers to the process of converting plaintext into cipher text at the sender with the same key that will be used to retrieve plaintext from cipher text at the recipient. while **asymmetric encryption** refers to the process of converting plaintext into cipher text at the sender with different key that will be used to retrieve plaintext from cipher text at the recipient [15].

Passive attacks mean that the attackers or the unauthorized parties just monitoring on the traffic or on the communication between the sender and the recipient, but not attempting to breach or shut down a service, This kind of attacks is very hard to discover, since the unauthorized party doesn't leave any traces. On the other hand **active attacks** mean that the attackers are actively attempting to cause harm to the network or the data. The attackers are not just monitoring on the traffic, but they also attempt to breach or shut down the service [4,15].

Authentication is the process of determining whether someone is the same person who really is, such as login and password in login pages while authorization is the process of ensuring that this person has the ability to do something [4, 9, 15].

Brute force is the attacker who is trying all of the possible keys that may be used in either decrypt or encrypt information [15].

1.3 A Brief History of Cryptography

The encryption process is as old as writing itself, Through this short historical combo, the most important stations in the progress of data encryption will be reviewed. It is believed that the first texts used or contained any encryption techniques were known 4000 years ago at the Veterans Egyptian where the hieroglyphic inscriptions on the tomb of the nobleman Khnumhotep II, They were written with a number of unusual symbols to confuse or obscure the meaning of the inscriptions [6].

2000 years ago, the Greek knew cylinder device called Scytale, which was the sender's part very similar to the recipient part, where a narrow strip of parchment or leather, was wounded around the Scytale and the message was written across it, so if anyone tries to read the text he will find meaningless letters, The only one that can read this text is the one who has the Scytale, This technique is similar to the transposition technique which will be later discussed in symmetric encryption section [5].
The Arab role in the data encryption, was since ancient times, Through the analysis of the text of the holy Qur'an text, Muslim scholars were able to invent frequency analysis technique for breaking monoalphabetic substitution ciphers about 1200 years ago, by Sheikh AL-Kindi in his famous book "Risalah fi Istikhraj al-Mu'amma (Manuscript for the Deciphering Cryptographic Messages)", which it was

the most advanced in cryptography since that time, until the World ware two, Figure 1 shows the first page of AL-Kindi's book, After AL-Kindi's invention, all cipher text became vulnerable to this cryptanalytic technique, until the development of the polyalphabetic cipher by Leone Battista Alberti, who is known as "The Father of Western Cryptology" in 1465 [6].

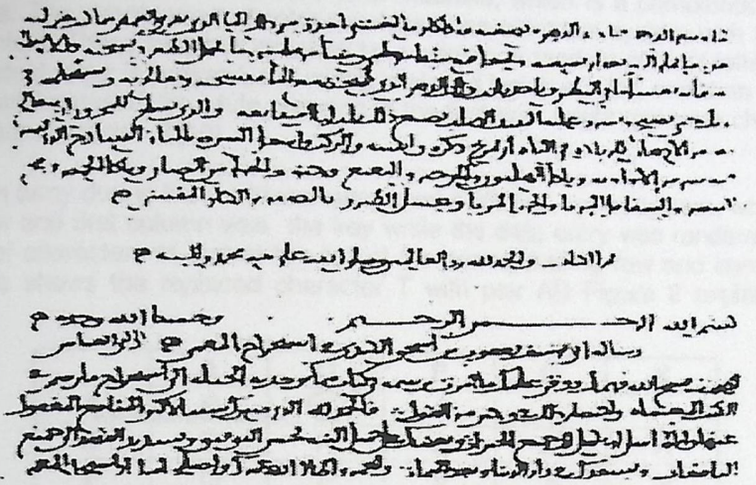


FIGURE.1: The first page of al-Kindi's manuscript On Deciphering Cryptographic Messages

The next step was in 1518 by Trithemius, a German monk, who wrote a table of Twenty-six column and Twenty-six row. Each row duplicate the above row but shifted by one letter.

In 1585, Blaise de Vigenere developed a Trithemius table by changing the way that the keywords system works. One of his used techniques is the plaintext as its own key.

Forty-three years later, a Frenchman named Antoine Rossignol helped his army to defeat the Huguenots, by deciphering a captured message. After that victory, Antoine was deciphering messages for the benefit of the French government many times. He used two lists to solve his ciphers: "one in which the plain elements were in alphabetical order and the code elements randomized, and one to facilitate decoding in which the code elements stood in alphabetical or numerical order while their plain equivalents were disarranged" [7].

The wheel cipher is a cylinder composed of Twenty six cylindrical piece of wood, The alphabetical letters inscribed randomly on each piece the [8].
The development in data encryption has begun to accelerate after the discovery of the telegraph, simply sending messages by the telegraph is not secure; therefore they had to provide means of data encryption before transmission.

In 1854, Charles Wheatstone and Lyon Playfair invented the Playfair system, which was consisted from 5X5 rectangle key, while the plaintext message divided into adjacent pairs, This system will be discussed later.

Before 1883, the encrypt ion process often depended on hiding of algorithm to protect data, Of course, This is not practical, but the first major advances in cryptography were made in the year 1883 by Kerkhoff by developing a set of principles which is now known as Kerkhoff principle, The major principle is, hidding the key of algorithm instead of hiding the algorithm itself [5].

Kerkhoff Principles [9]

1. Ciphertext should be unbreakable.
2. The cryptosystem should be convenient for the correspondent.
3. The key should be easily remembered and changeable.

4. The Ciphertext should be transmitted by the telegraph.
5. The cipher apparatus should be easily portable,
6. The cipher machine should be relatively easy to use.

In 1915, two Dutch navy officers invented the rotor machine; which is a combination of electrical and mechanical systems. The simple view of rotor machine is an electrical system with twenty-six switches pressed by the plaintext, These switches attached by a wire to a random contact letter on the output, for example if the plaintext letter is pressed, the wiring is placed inside a rotor, and then rotated with a gear every time a letter was pressed. So while pressing ***A*** the first time might generate character ***D***, the next time it might generate character ***S*** [10].

In 1918, the german army during the world war one, used ADFGVX cipher system, which consisted from a table, the first row and first column was the key while the data entry was randomly replaced by the plaintext with pair of characters of text at the top of the corresponding row and corresponding column, The following figure shows the replaced character T with pair AD Figure 2 explain ADFGVC cipher system [11].

	A	D	F	G	X
A	B	T	A	L	P
D	D	H	O	Z	K
F	Q	F	V	S	N
G	G	J	C	U	X
X	M	R	E	W	Y

FIGURE 2 : Example of Using ADFGVX cipher system.

Lester Hill is one of the few scientists who had concluded that mathematics inevitably necessary for the success of encryption, and the encryption remained the same until 1941 when Adrian Albert Benefited from Hill theorem and built an encryption system based on mathematics [12].

In 1948, Shannon published "A Communications Theory of Secrecy Systems", In this paper Shannon's analysis demonstrates several important features of the statistical nature of language that make nearly the solution of all previous ciphers very straight forward, One of the most important result in this paper is that Shannon developed a measure for cryptographic strength called the "unicity distance" [14].

During a collaboration between Whitfield Diffie and Martin Hellman in 1976 , the Diffie-Hellman key agreement was invented, The method was based on the selected three variables at the sender ***(x, a, P)*** and generating of ***s***, then sending ***(s, a, P)*** to the recipient, the recipient chooses ***y*** and uses ***y*** with ***(a, P)*** to generate ***r*** and sends ***r*** to the sender, the sender use ***r*** with ***(x, P)*** to generate the public key, The recipient also uses ***s*** with ***(y, P)*** to generate the same public key, Figure 3 explains this idea [15].

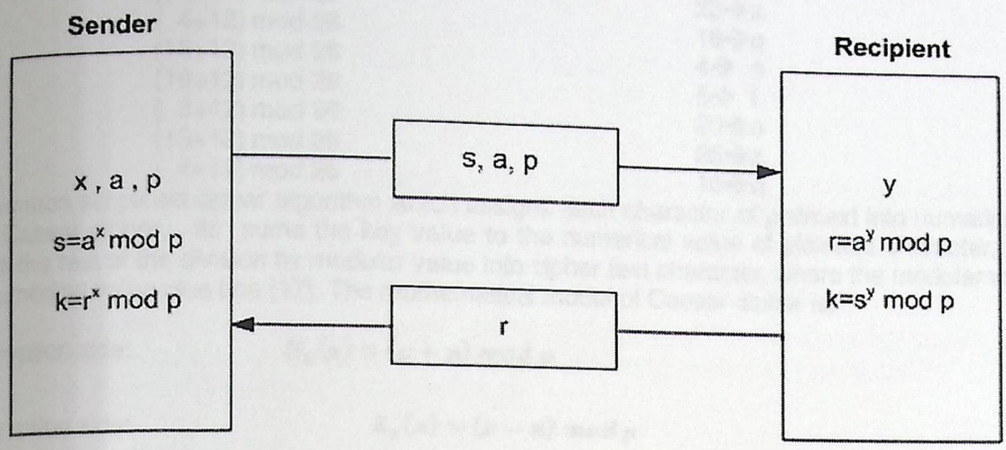


FIGURE .3 : Diffie-Hellman key generation

After Diffie-Hellman approach, the cryptography was divided into symmetric and asymmetric cryptography, and then many techniques and methods were developed. The next section is about the symmetric and asymmetric encryption [16].

2. SYMMETRIC AND ASYMMETRIC ENCRYPTION

Encryption is the strongest and the safest way in securing data. Certainly, it is the most common one. Encryption systems are divided into two major types or forms, symmetric and asymmetric.

Symmetric encryption is known as secret key or single key, The receiver uses the same key which the sender uses to encrypt the data to decrypt the message,. This system was the only system used before discovering and developing the public key., A safe way of data transfer must be used to moving the secret key between the sender and the receiver in symmetric encryption. Figure 4 shows how the system works. Symmetric encryption occurs either by substitution transposition technique, or by a mixture of both. Substitution maps each plaintext element into cipher text element, but transposition transposes the positions of plaintext elements.

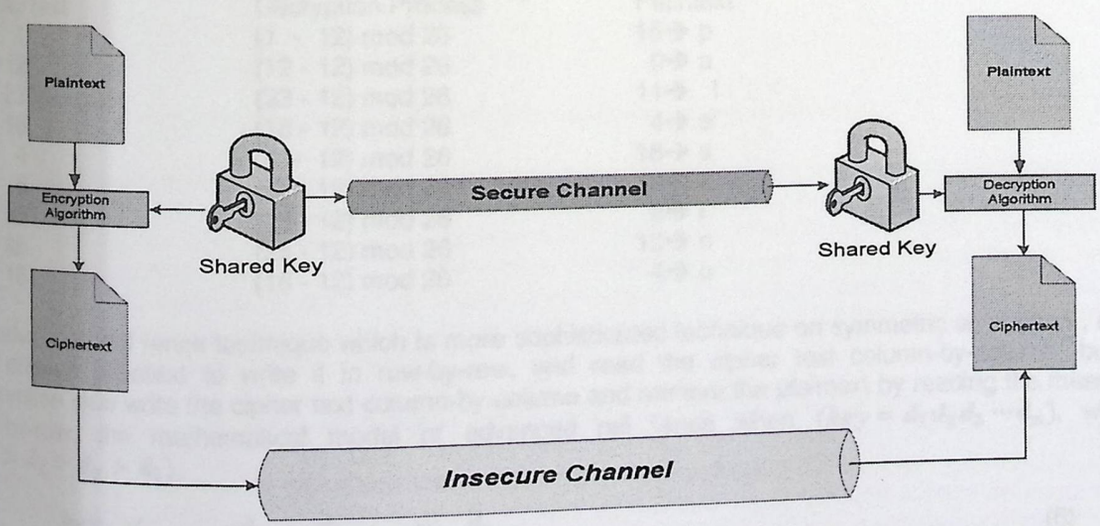


FIGURE .4 : Simplified model of conventional encryption

Plaintext	Encryption Process	Cipher text
p→15	(15+12) mod 26	1→ b
a→ 0	(0+12) mod 26	12→m
l →11	(11+12) mod 26	23→x
e→ 4	(4+12) mod 26	16→q
s→18	(18+12) mod 26	4→ e
t→19	(19+12) mod 26	5→ f
i→ 8	(8+12) mod 26	20→u
n→13	(13+12) mod 26	25→z
e→ 4	(4+12) mod 26	16→q

The common simplified cipher algorithm which assigns each character of plaintext into numerical value is called Caesar cipher, , its sums the key value to the numerical value of plaintext character, and then assigns the rest of the division by modular value into cipher text character, where the modular value is the max numerical value plus one [17], The mathematical model of Caesar cipher is:

At encryption side: $$E_n(x) = (x + n) \bmod p \quad (3)$$

At decryption side: $$E_n(x) = (x - n) \bmod p \quad (4)$$

Where x is the plaintext character and x is shift value, the following example illustrates Caesar cipher model:

Example 1:
Let the plaintext message is "Palestine" and the key value=12 , and use the simplest symmetric encryption algorithm ,which called "Caesar cipher", the Caesar table will be:

Table .1: Caesar Table

a	b	C	d	e	f	g	h	i	j	k	L	m	n	o	p	q	r	s	T	u	v	W	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

The cipher text which arrive to the receiver is "bmxqefuzq", and the cipher text is entered into decryption process in the receiver to decrypt the text as follow:

Cipher text	Decryption Process	Plaintext
b→ 1	(1 - 12) mod 26	15→ p
m→12	(12 - 12) mod 26	0→ a
x→ 23	(23 - 12) mod 26	11→ l
q→ 16	(16 - 12) mod 26	4→ e
e→ 4	(4 - 12) mod 26	18→ s
f→ 5	(5 - 12) mod 26	19→ t
u→ 20	(20 - 12) mod 26	8→ i
z→ 25	(25 - 12) mod 26	13→ n
q→ 16	(16 - 12) mod 26	4→ e

an advanced rail fence technique which is more sophisticated technique on symmetric encryption , uses the original plaintext to write it in row-by-row, and read the cipher text column-by-column, but at decryption side write the cipher text column-by-column and retrieve the plaintext by reading the message row-by-row, the mathematical model of advanced rail fence when $(key = d_1 d_2 d_3 \cdots d_n)$, where $(d_3 > d_1 > d_n > d_2)$:

$$\begin{array}{lccccc} key & d_1 & d_2 & d_3 & \cdots & d_n \\ & p_1 & p_2 & p_3 & \cdots & p_n \end{array} \quad (5)$$

$$
\begin{array}{lccccc}
key & d_1 & d_2 & d_3 & \cdots & d_n \\
 & C_{2\times l/n+1} & C_1 & C_{3\times l/n+1} & \cdots & C_{l/n+1} \\
 & C_{2\times l/n+2} & C_2 & C_{3\times l/n+2} & \cdots & C_{l/n+2} \\
 & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & C_{3\times l/n} & C_{l/n} & C_{4\times l/n} & \cdots & C_{2\times l/n}
\end{array} \qquad (6)
$$

Where d_1 is the smallest digit among digits of key that consist from n digits, l represent number of characters in plaintext message, p_i is the i^{th} character of plaintext message and C_i is the i^{th} character of cipher text output.

Example 2:

To understand and accommodate advance rail fence technique, let us consider $(key = 5236417)$, plaintext (p)"AES is a block cipher intended to replace DES for commercial application":

Using equation (5), the encryption message:

Key	5	2	3	6	4	1	7
Plaintext:	A	e	s	i	s	a	b
	L	o	c	k	c	i	p
	H	e	r	i	n	t	e
	N	d	e	d	t	o	r
	E	p	l	a	c	e	d
	E	s	f	o	r	c	O
	M	m	e	r	c	i	A
	L	a	p	p	l	i	C
	A	t	i	o	n	x	X

Output: Aitoeciixeoedpsmatscrelfepiscntcrclnalhneemlaikidaorpobperdoacx

Using equation (6), the decryption message (plaintext):

Key	5	2	3	6	4	1	7
Plaintext:	A	e	s	i	s	a	B
	i	o	c	k	c	i	P
	h	e	r	i	n	t	E
	n	d	e	d	t	o	R
	e	p	l	a	c	e	D
	e	s	f	o	r	c	O
	m	m	e	r	c	i	A
	l	a	p	p	l	i	C
	a	t	i	o	n	x	X

Output: Aesisablockcipherintendedtoreplacedesforcommercialapplication

From previous examples, the plaintext is translated into different cipher text and then transferred throw unsecured channel to the receiver, while the secrete key which is been used in encryption process will be transferred throw secured channel, At the receiver side the inverse of the secret key or/and the inverse of encryption process are used to decrypt the cipher text and to retrieve the original plaintext, Caesar mechanism is the core for all encryption model, from easy to very complicated one, in other word, the encryption process needs key to convert the plaintext into cipher text, but at the receiver the inverse of processes will retrieve the original plaintext.

Symmetric encryption has many advantages over asymmetric. Firstly, it is faster since it doesn't consume much time in data encryption and decryption. Secondly, it is easier than asymmetric encryption in secret key generation. However, it has some disadvantages, for example key distribution and sharing of the secret key between the sender and the receiver, also symmetric key encryption incompleteness, since some application like authentication can't be fully implemented by only using symmetric encryption [18].

In 1976 Diffie and Helman invented new encryption technique called public key encryption or asymmetric encryption; Asymmetric encryption is the opposite of symmetric encryption in safety, since it doesn't require sharing the secret key between the sender and the receiver. And this is the main difference between symmetric and asymmetric encryption, the sender has the public key of the receiver. Because the receiver has his own secret key which is extremely difficult or impossible to know through the public key, no shared key is needed; the receiver is responsible for establishing his private and public key, and the receiver sends the public key to all senders by any channel he needs, even unsecured channels to send his public key, asymmetric key can use either the public or secret key to encrypt the data. Also it can use any of the keys in decryption, asymmetric encryption can be used to implement the authentication and non-repudiation security services, and also it can be used for digital signature and other application that never be implemented using symmetric encryption. Figure.5 shows how the system works.

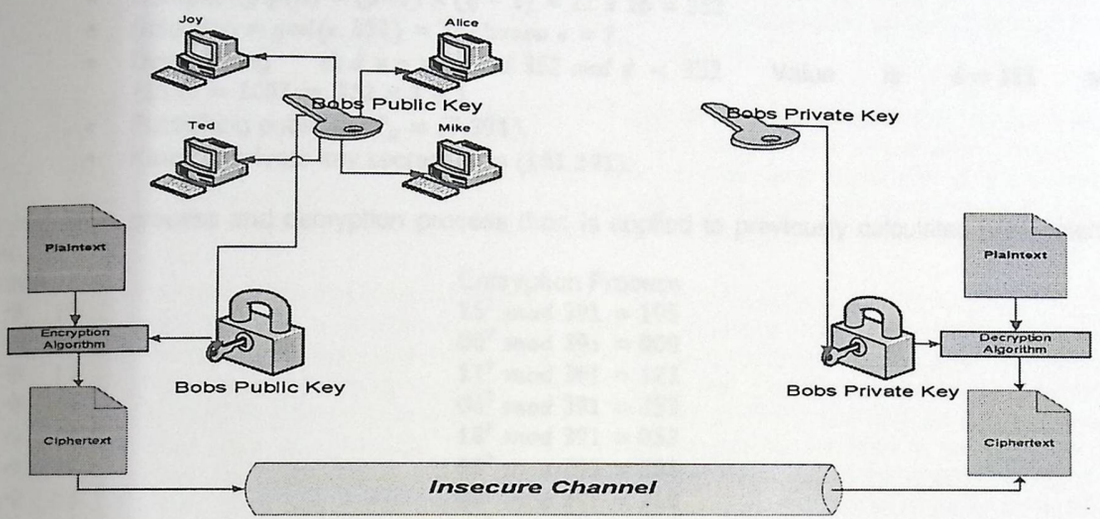


FIGURE 5 : Simplified model of asymmetric encryption

Asymmetric encryption is slower and very complicated in calculations than symmetric encryption . Therefore, asymmetric encryption deals with plaintext as a group of numbers which are manipulated in mathematics, while the plaintext in symmetric encryption deal as group of symbols and characters, the encryption process may permute these symbols, or may substitute one symbol by another.

So the nature of the data determines the system of encryption type. And every system has its own uses. For example, asymmetric encryption may be used in authentication or in sending secret key for decryption.

To understand asymmetric encryption, lets us take RSA model which is an example on asymmetric encryption, RSA model main steps:

RSA Model Steps:

- Each user generates a public/private key pair by selecting two large primes at random p, q.
- Computing modular value $n = p \times q$
- Calculating the Euler's function $\phi(n) = (p-1) \times (q-1)$
- Selecting at randomly the public encryption key e, where $1 < e < \phi(n)$, and e is prime relative to the $\phi(n)$.

- Solving the following equation to find private decryption key d, $e \times d = 1 \bmod \phi(n)$, and $0 \leq d \leq n$.
- Publishing their public encryption key: $P_K = (e, n)$.
- Keeping secret private decryption key: $P_R = (d, n)$.
- At the encryption side the sender uses encryption mathematical equation $C = P^e \bmod n$.
- At the decryption side the receiver uses decryption mathematical equation $P = C^d \bmod n$.

Example 3:

Let a part of the plaintext message be "Palestine", then the RSA key generation process is:

- Select two prime numbers: p=23 & q=17
- Computing $n = p \times q = 23 \times 17 = 391$
- Computing $\phi(n) = (p-1) \times (q-1) = 22 \times 16 = 352$
- Selecting e: $gcd(e, 352) = 1$; $choose\ e = 7$
- Determining d: $d \times e = 1 \bmod 352$ and $d < 352$ Value is $d = 151$ since $151 \times 7 = 1057 = 352 \times 3 + 1$
- Publishing public key $P_K = (7, 391)$.
- Keeping private key secrete $P_R = (151, 391)$.

The encryption process and decryption process then is applied to previously calculated parameters as follow:

Plaintext			Encryption Process
p	→	15	$15^7 \bmod 391 = 195$
a	→	00	$00^7 \bmod 391 = 000$
l	→	11	$11^7 \bmod 391 = 122$
e	→	04	$04^7 \bmod 391 = 353$
s	→	18	$18^7 \bmod 391 = 052$
t	→	19	$19^7 \bmod 391 = 383$
i	→	08	$08^7 \bmod 391 = 219$
n	→	03	$13^7 \bmod 391 = 055$
e	→	04	$04^7 \bmod 391 = 353$

The cipher text will arrive the receiver, and at the receiver the cipher text will be entered into decryption process to decrypt the text as follow:

Decryption Process	Plaintext		
$195^{151} \bmod 391 = 015$	015	→	p
$000^{151} \bmod 391 = 000$	000	→	a
$122^{151} \bmod 391 = 011$	011	→	l
$353^{151} \bmod 391 = 004$	004	→	e
$052^{151} \bmod 391 = 018$	018	→	s
$383^{151} \bmod 391 = 019$	019	→	t
$219^{151} \bmod 391 = 008$	008	→	i
$055^{151} \bmod 391 = 003$	003	→	n
$353^{151} \bmod 391 = 004$	004	→	e

The mathematical model for symmetric and asymmetric encryption consists of key, encryption and decryption algorithm and powerful secured channel for transmitting the secrete key or any channel for transmitting the public key from the sender to the receiver, the mathematical model similar to equations (1 - 2):

At encryption side: $C = E_K(P)$

At decryption side: $P = D_K(C)$

Where C is the cipher text to be sent, E is the encryption algorithm, P is the plaintext, D is the decryption algorithm, and K is the key used inside the encryption and/or decryption process.

3. RESULTS AND COMPARISON

When it comes to encryption, the latest isn't necessarily the best. You should always use the encryption algorithm that is right for the job and has been extensively publicly analyzed and tested, something the cryptographic community won't have had the chance to do with a brand new algorithm. Let's have a look at some of the most widely-used algorithms. For most people, encryption means taking plaintext and converting it to cipher text using the same key, or secret, to encrypt and decrypt the text. This is symmetric encryption and it is comparatively fast compared to other types of encryption such as asymmetric encryption. The most widely-used algorithm used in symmetric key cryptography is AES (Advanced Encryption Standard). It comprises three block ciphers, AES-128, AES-192 and AES-256, each of which is deemed sufficient to protect government classified information up to the SECRET level with TOP SECRET information requiring either 192 or 256 key lengths.

The main disadvantage of symmetric key cryptography is that all parties involved have to exchange the key used to encrypt the data before they can decrypt it. This requirement to securely distribute and manage large numbers of keys means most cryptographic services also make use of other types of encryption algorithms. Secure MIME (S/MIME) for example uses an asymmetric algorithm - public/private key algorithm - for non-repudiation and a symmetric algorithm for efficient privacy and data protection.

Asymmetric algorithms use two interdependent keys, one to encrypt the data, and the other to decrypt it. This interdependency provides a number of different features, the most important probably being digital signatures which are used amongst other things to guarantee that a message was created by a particular entity or authenticate remote systems or users. The RSA (Rivest, Shamir and Adleman) asymmetric algorithm is widely used in electronic commerce protocols such as SSL, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations. As RSA is much slower than symmetric encryption, what typically happens is that data is encrypted with a symmetric algorithm and then the comparatively short symmetric key is encrypted using RSA. This allows the key necessary to decrypt the data to be securely sent to other parties along with the symmetrically-encrypted data.

4. SUMMARY

Cryptography is used to ensure that the contents of a message are confidentiality transmitted and would not be altered. Confidentiality means nobody can understand the received message except the one that has the decipher key, and "data cannot be changed" means the original information would not be changed or modified; this is done when the sender includes a cryptographic operation called a hash function in the original message. A hash function is a mathematical representation of the information, when information arrives at its receiver; the receiver calculates the value of this hash function. If the receiver's hash function value is equivalent to the sender's, the integrity of the message is assured [15].in this survey paper we describe and compare between symmetric and asymmetric encryption technique ,provide many example to show the differences .

5. REFERENCES

[1] J. Badeau.,: " The Genius of Arab Civilization ", Second Edition. MIT Press,(1983), USA.

[2] M .Chapple., M Solomon,: " Information Security Illuminated " First Edition. Jones and Bartlett Publishers, (2005), USA.

[3] J.R Childs: " General Solution of the ADFGVX Cipher System ". Aegean Park Press, ,(2000), USA.

[4] D.Delfs., and K. Helmut.,: " Introduction To Cryptography: Principles and applications ", Second Edition. Springer Science & Business Media, (2007), Germany.

[5] G .Dieter: " Computer Security ", Second Edition. John Wiley & Sons, , (2005), UK.

[6] A. Forouzan.,: " Cryptography and Network Security ", First Edition. McGraw-Hill, (2007), USA.

[7] R Hamamreh., M Farajallah., " Design of a Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher ". International Journal of Computer Science and Network Security, (2009): Vol (9), pp: 12-21.

[8] J Hoffstein., et al, " An Introduction to Mathematical Cryptography ", First Edition. Springer Science & Business Media, (2008):, Germany.

[9] H.Kenneth, " Elementary Number Theory and Its Applications " Third Edition. Addison-Wesley, (1992): Germany.

[10] M.Lucas, " Thomas Jefferson wheel cipher ", Monticello Research Department, Thomas Jefferson Foundation, Charlottesville, (1995):, VA.

[11] S .Maret," Cryptography Basics PKI ", First Edition. Dimension Data SA, ., (1999):, Switzerland.

[12] E.Ralph., F Weierud" Naval Enigma: M4 and Its Rotors ". Crypologia, ,. (1987):, Vol(11),pp:235-244.

[13] W .Reinhard., " Cryptology Unlocked ", Translation Edition By Angelika Shafir. John Wiley & Sons, (2007):, UK.

[14] H. Rodríguez, et al,: " Cryptographic Algorithms on Reconfigurable Hardware ", First Edition. Springer, (2006), USA.

[15] D.Salomon" Data Privacy and Security " First Edition. Springer-Verlag New York, ., (2003):, Inc. USA.

[16] C .Shannon,. " Communication Theory of Secrecy Systems ". Bell Syst, (1949):, Tech. J., Vol (28), pp: 656-715.

[17] W .Stallings, " Cryptography and network security, Principles and practices ", Fourth Edition. Pearson Prentice Hall, (2006):, USA.

[16] K .Thomas, : " The Myth Of The Skytale ". Taylor & Francis, (1998), Vol (33), pp: 244-260.

RMISSION TO USE COPYRIGHTED MATERIAL IN A THESIS

n a graduate student at the University of Palestine polytechnic university and am preparing my final thesis. I requesting permission to include excerpt(s) from your thesis described below. The source(s) of the included terial will be fully identified in the Thesis.

of Thesis: practical one way hash algorithm based on non-invertible matrix using matrix multiplications

ee: master degree Graduating Year: 2011

ission is hereby granted to Mohammed said Ibrahim abutaha
(Author of thesis)

he University of Palestine polytechnic university reproduce the following in the thesis:

le of article/book: **Computer Security: Principles and Practice/ first edition**

e or page numbers: figure 2.5 /figure 2.6

pyright holder or representative of the copyright holder(s), the undersigned is aware that the author of Thesis will be granting irrevocable non-exclusive licenses .

ature of copyright holder or

e *William Stallings* Date: September 7, 2011

ess: p. o. box 2405

Brewster MA 02631 USA