REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

ANALYSIS OF NETWORK SECURITY USING CYPHER TECHNIQUE

Mohanad Ahmed Abdulrazzaq AL-ZAMILI

Master's Thesis

Supervisor

Prof. Dr. Osman Nuri UÇAN

Istanbul, 2022

ANALYSIS OF NETWORK SECURITY USING CYPHER TECHNIQUE

Mohanad Ahmed Abdulrazzaq AL-ZAMILI

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled ANALYSIS OF NETWORK SECURITY USING CYPHER TECHNIQUE prepared by MOHANAD AHMED ABDULRAZZAQ AL-ZAMILI and submitted on 23/12/2022 has been **accepted unanimously** for the degree of Master of Science in Information Technology.

Prof. Dr. Osman Nuri UÇAN

the Supervisor

Thesis Defense Committee Members:

Prof. Dr. Osman Nuri UÇAN	Department of Electrical and Electronics Engineering, Altınbaş University	_____
Asst. Prof. Dr. Oğuz KARAN	Department of Software Engineering, Altınbaş University	_____
Asst. Prof. Dr. Adil Deniz DURU	Department of Coaching Education, Marmara University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission date of the thesis to Institute of Graduate Studies: ___/___/___

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Mohanad Ahmed AL-ZAMILI

Signature

DEDICATION

In the name of Allah, the merciful. I would first like to thank God Almighty who gave me the opportunity and helped me to finish my Master degree. My dear Father To my angel in life.to the meaning of love and to the meaning of compassion and dedication .to the smile of life and the secret of existence to whom your invitation was the secret of my success and affection Dear Mother. To the one who has the largest and so I rely to a candle burning illuminates the darkness of my life to those who have gained strength and love without limits to whom did you know the meaning of life. To my dear Wife, the source of comfort, happiness, and affectionate heart is for you and for me, to the one who pleases my heart and blesses to the flower garden that sprouts flowers blossoms. To those who are closest to my soul to those who share with me the bosom of pain, and through them I draw my strength and my determination. My Sisters and My Son. Who paved the way in front of me to reach the height of science to My Professor "I was in my studies and shared my thoughts reminder and appreciation My Friends.

PREFACE

I would like to express my sincere gratitude to Prof. Dr. Osman Nuri UÇAN for all the knowledge and support he provided during my study for the Master Degree and throughout the work to complete this thesis, which have diluted the difficulties I have faced during the study. You have made my dream come true.

ABSTRACT

ANALYSIS OF NETWORK SECURITY USING CYPHER TECHNIQUE

Al-Zamili, Mohanad Ahmed Abdulrazzaq

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Prof. Dr. Osman Nuri UÇAN

Date: 12/2022

Pages: 147

Cryptographic procedures, such as encryption algorithms and cryptanalysis tools, necessitate a thorough understanding of mathematics. It's been used for a long time to teach these cryptographic concepts through hands-on experimentation supported by a theoretical framework. It's not enough to do theoretical hands-on trials if you're dealing with complex encryption approaches. There has been a huge improvement in educational programmes over the past few years, and cryptography is no exception. The goal of cryptographic programmes is to help students better comprehend the complex algorithms by presenting them visually in an understandable manner.

Keywords: Cryptography, Cipher, Encryption, Decryption, Algorithm.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

1. INTRODUCTION

Cryptographic procedures, such as encryption algorithms and cryptanalysis tools, necessitate a thorough understanding of mathematics. It's been used for a long time to teach these cryptographic concepts through hands-on experimentation supported by a theoretical framework. It's not enough to do theoretical hands-on trials if you're dealing with complex encryption approaches. There has been a huge improvement in educational programmes over the past few years, and cryptography is no exception. The goal of cryptographic programmes is to help students better comprehend the complex algorithms by presenting them visually in an understandable manner.

1.1 OVERVIEW

It was the duty of those who worked with essential and sensitive information, such as military secrets, to keep information safe in ancient civilizations. This function became a secure science. There are only a few persons who wanted to master these talents because of the specialised aspects of the work. However, the advent of electronic communications has grown in importance over time, particularly since the creation of the World Wide Web, which introduced an environment in which data transfer has become faster and more effective, but more vulnerable to penetration by unauthorised persons to introduce. Nowadays, network security is critical to safeguarding online data due to the prevalence of the internet and the use of wired and wireless connectivity. It's common for people to post personal information about their lives on sites like social networking and banking applications because they use the internet daily. These sites are trusted because they are believed to be secure and capable of protecting their users' personal information from unauthorised access. Hardware and software protection measures that are used to prevent unauthorised access and modification to system assets and information are called network security measures. It assures that only authorised users will be able to access the system. To attain the principles of security, you'll need these three unique keys to succeed:

i. It is critical to keep information private to prevent it from being misused.

ii. Data integrity must be protected from unauthorised tampering to be maintained.

iii. Availability: This refers to the fact that the data should be accessible and readable by the authorised user.

Cryptography must be at the heart of any secure communication or data exchange via a network in order to be effective. Cryptography has its origins in two Greek words: 'kryptós,' which means "hidden or secret," and 'gráphein,' which means "to write." It is a collection of techniques for encrypting and decrypting data that use a unique key to achieve their results. After the data has been encrypted, only the intended receiver will be able to decrypt it and read the true plaintext that was included within it.

The upshot is that communications can be sent without the sender having to worry about the contents of the messages being accessed by someone who is not authorised to view them. Because cryptography is built on challenging mathematical concepts, students with limited mathematical abilities will be unable to gain a thorough understanding of the subject matter through traditional theoretical instruction. Even the most complex cryptographic algorithms are easier to comprehend when they are taught through active learning, real-world examples, and practical applications.

Students will also benefit from this since they will be able to put what they have learned into practise, which will aid them in remaining focused and motivated throughout the semester. It is possible for students in any subject area to benefit from visual learning techniques, and network infrastructure is no exception. A range of tools have been developed to help people understand even the most sophisticated algorithms and protocols on the market.

1.2 GOALS AND PURPOSES

i. To learn the fundamentals of cryptography.

ii. To learn about the state-of-the-art in network cryptography.

iii. To have a thorough understanding of secure data transfer concepts.

iv. To learning and implementing cryptographic methods.

v. To make complex cryptography concepts understandable to a general audience.

vi. Design and describe experiments in the lab that demonstrate how the features of each algorithm vary based on the software tools given.

vii. To make it easier for people to understand these algorithms through some real-world examples and experiments.

viii. evaluating other cryptographic algorithms for efficiency.

1.3 THE PROBLEM'S BACKGROUND AND SOURCE OF MOTIVATION

Currently, a lot of systems and platforms are working to increase their efficiency to better satisfy the needs of their customers' applications. The integrity and validity of the communications they receive must be able to be relied upon by users and devices to preserve their personal information and prevent the messages from being tampered with. To achieve integrity, standard cryptographic procedures are both essential and beneficial instruments to have on hand. It is critical to have a diverse range of cryptographic techniques accessible to maintain the system's security. If this is not done, critical information may be leaked. To develop a solution that can be applied in all situations, it is necessary to consider the many application areas.

When it comes to Internet of Things (IoT) applications, sensors, which are frequently the core components, lack the power to incorporate complex security measures. Therefore, the procedure must be adaptive and feasible to be used in a variety of situations. Also crucial is an understanding of the network structure of the IoT system to ensure the security of the system. Dispersed networks, rather than relying on a centralised system with its inherent bottlenecks and constraints on scalability, are frequently the preferable option in many situations. Consequently, we may rest comfortable that relying on the authentication centre to validate the identities of users is a terrible idea.

The protocol should be able to detect other users since the authentication centre is no longer required in this scenario. Furthermore, wireless sensor networks are widely employed in industries such as manufacturing, weather monitoring, e-health, and everyday life as the principal Internet of Things (IoT) domain. Because each sensor primarily focuses on a specific sort of data, such as location information from a GPS sensor, the amount of data transmitted per transmission is quite small. The sensors will routinely communicate the source data at regular intervals, even though each session is only a few minutes long. When it comes to efficiency, the standard protocol does not meet the needs of these one-of-a-kind applications.

The management of keys and passwords must be given considerable consideration. Passwords must be used when dealing with sensitive information such as secret keys, private keys certificates, and so on. Most password breaches occur as a result of improper password hygiene. A similar set of concerns can occur in terms of system security due to the system's sensitivity to attack. This new strategy should be able to prevent this problem more effectively.

1.4 GOALS ARE CONCRETE AND VERIFIABLE

A series of concrete and verifiable goals are required for this thesis work to achieve the objectives. These objectives demonstrate, step by step, the way to the result, which can be seen below.

i. Conduct research on the Internet of Things and its security concerns, as well as peer-to-peer networks and cryptography. Summarize the notion of linked technologies and their execution. Then, discover the outcome of previous studies. Analyze the benefits and drawbacks of all viable options. Determine what has been done and what remains to be done.

ii. Conduct research on previous security protocols. Learn about existing security protocol concepts and flaws through focused assaults. Consider possible changes and enhancements to this project. After that, you should have a thorough knowledge of the Crypto tools's design philosophy as well as its practical issues.

Consider the protocol's required features next. These characteristics are as follows: scalable, lightweight, rapid, flexible, integrity, authentication, and availability. When the requirements are clear, it is time to create a security protocol prototype.

iii. Discover the Crypto tools, a context-aware distributed system designed for the Internet of Things. Investigate the system's principles and architecture, and then put the techniques into action using the source code. Determine each system layer and how to implement the security protocol in this platform.

iv. Create and implement a secure communication platform on the Crypto tools. The first and most critical step is to adhere to the platform's principles. The less changes made to the original system, the better. Each component should be modularized and less tightly connected. Basic cryptographic functions and key management for secure communication should be thoroughly evaluated for stability and reliability.

v. Use demo clients to test and assess secure communication. Using BAN logic, examine the security protocol. From a practical standpoint, secure communication should also be examined. Learn about the hazards and limitations of secure communication.

1.5 RESEARCH METHODOLOGY

Choosing the right techniques and procedures is very important to the success of a project. Security problems on the Internet of Things and peer-to-peer networks will be looked at with the help of the best research available, inspired by real-world problems and challenges. At the same time, it's important to brush up on the basics of cryptography and security, which include algorithms, protocols, and information found in books and other sources of information. You also need to understand the pros and cons of different cryptography and security systems. To make a new protocol, security flaws in other protocols must be considered and avoided. There are a lot of new real-world conceptual attacks on different methods or protocols that aren't clearly explained in books or used and summarised in papers, so it's important to list all the possible vulnerabilities and threats for each algorithm that could be used in real life.

1.6 RESEARCH HYPOTHESES

As part of this research, an Internet of Things security protocol is being created, and a cryptographic communication system is being deployed. Additionally, this protocol provides a fast and effective method of authenticating each client, in addition to protecting the reliability of the messages. To handle the transmission patterns of brief but regular communications, a single dynamic security level modification will be made specifically for wireless sensor networks. This thesis work includes a logic analysis of this protocol as well as a functional laboratory prototype of the security procedures for the Internet of Things, as well as all the necessary modules and features.

The safety of a system depends on many different pieces working together. A lot more is involved in this suggested security protocol than on this one portion of it, as you can see. To adhere to time constraints, this thesis will not discuss other security problems such as identity management, file permission control, or security attacks against this protocol.

1.7 THESIS LAYOUT

Chapter One: Introduction, where this chapter is a basic information about the

Chapter Two: Literature Review

Chapter Three: Methodology

Chapter Four: Result and Discussions

Chapter Five: Conclusion and Future works

2. LITERATURE REVIEW

2.1 INTRODUCTION

Because the information was intended for a limited number of people who needed to keep it secret, security procedures were simple and did not rely on sophisticated mathematical calculations in the past. These early techniques were assumed to be secure in comparison to today's standards, but when evaluated and cracked, they were found to be weak. More secure algorithms have become necessary as a result of the boom in online and electronic communication use in order to protect the large amounts of data being transported across networks from unauthorized access. As a result, cryptographic methods have evolved, making it more difficult to evaluate and break these algorithms.

Platform users must have faith in the security of their systems and networks in order for their use to continue. Before proceeding to the next level of education, a student must be able to comprehend and appreciate the complete extent of security issues. Because of the usage of complex formulas, cryptography is difficult to master for those who do not have a fundamental understanding of mathematics. As a result, utilizing experimentation to explain these difficult techniques can substantially shorten the amount of time it takes students to learn the material. Experiments should follow a predetermined teaching paradigm to provide students the best opportunity to learn.

The difficulty and intensity of the course increase as the student progresses through it, but engaged learning tactics help the learner stay focused on the subject matter. Students should first study about the earliest cryptographic methods, followed by an examination of how this discipline has grown into today's procedures. The student will have a better knowledge of the path taken by cryptographers and how contemporary approaches have evolved as a result of this method. Visual aids benefit students because they can see how the approaches are used in practice, which improves their learning.

The purpose of this style of learning is to use experiments to learn about a complicated subject matter on a practical level.

The authors stated that visual learning could be utilized to overcome some of the challenges associated with teaching cryptography. Using a simulator to teach cryptography provides a step-by-step approach to the process, allowing students to build a knowledge of algorithmic ideas. This teaching method has been acclaimed by students, and studies have shown that it helps them get higher results.

2.2 LITERATURE REVIEW

2.2.1 The Advancement Of Computer Network Security

i. Academic communities and the broader public media have only given its function in fostering viral assaults a passing amount of attention in recent years. On the other hand, these kinds of tactics are being exploited in an increasingly prevalent manner so that malware might be enabled. As the complexity of computer security has increased, authors of malicious software have started using security as a way to disguise their code and avoid being discovered (Cabaj et al., 2018). (Cabaj et al., 2018). This method was unearthed for the very first time in 2011, and ever since then, those who work in the field of digital forensic investigation have experienced a rise in both popularity and complexity, as well as an increase in their level of anxiety (Barwise, 2018). (Barwise, 2018).

ii. "Digital media security" refers to the practice of using digital media files as cover media for the purpose of carrying out covert communications. Photographs, motion pictures, audio files, and HTML pages are just a few examples of the types of digital media assets that can be utilized as cover media.

iii. In the past, security malware would make advantage of these media assets in order to hide malware settings and configuration files, provide a URL from which the virus could download further components, and/or implant the malicious code directly within the file (Cabaj et al., 2018). (Cabaj et al., 2018). Advanced malware, such as the kind that is deployed by Advanced Persistent Threat (APT) groups employed by

nation-states, is able to exfiltrate stolen data by masking it in digital media thanks to a security technique (Barwise, 2018). (Barwise, 2018).

iv. The year 2011 was dubbed "The Year of the Hack" by the media as a result of the multiple data breaches that rocked corporate companies as well as government bodies. These breaches affected both private and public institutions. It is anticipated that the information stolen as a result of these data breaches totaled petabytes, with Operation Shady RAT being responsible for a significant portion of that amount (Zielinska, Mazurczyk, & Szczypiorski, 2014). (Zielinska, Mazurczyk, & Szczypiorski, 2014). The first large-scale malware operation that utilized digital media security was called "Operation Shady RAT," and it was the name of the operation. The attack started in 2006, but it wasn't discovered until 2011, even though it had been going on since 2006. It was a campaign of online espionage that had caused damage to more than 70 governmental and commercial sector entities across 14 countries. At one point, it was believed that China was involved in the advanced persistent threat known as Operation Shady RAT (Gross, 2011). (Gross, 2011). The phishing scam disguised as a Trojan horse is responsible for infecting the vast majority of users. A piece of malicious software known as Downbot, which attacked PCs, established a backdoor in the system. When files that appeared to be HTML pages or JPEG images were downloaded, it was clear that more security precautions were going to have to be taken. These JPEG and HTML photos contained commands that made it possible for an unauthenticated person to access files on the compromised computer from a distant location (Mazurczyk & Caviglione, 2015).

v. It is hypothesized that the computer worm known as Stuxnet was infected by the virus called Duqu, which was discovered in 2011. This virus took precautions in order to get access to industrial control systems, which it did by modifying its own security protocols (ICS). The information that was obtained was encrypted and appended to some seemingly innocuous digital photos right before it was sent over the Internet to a server that was used for command and control (C&C). It was impossible for Duqu to differentiate the altered images from the real ones because

they were supplied in a box along with authentic digital photographs (Mazurczyk & Caviglione, 2015). 6) The Gatak/Stegoloader malware improved upon the security methods that were utilized by Duqu and other malware. These methods were used to add encrypted data to an image file that was already in existence. This malicious software, which was found for the first time in 2015, did not affix its harmful code to the very end of an image file; rather, it contained it entirely within the picture file itself. As a result of this, it was able to avoid being discovered. Gatak/Stegoloader infected its victims by disguising itself as a PNG image that was inserted in unapproved software and then spreading the infection. The Gatak/Stegoloader unloading process begins as soon as a user downloads a piece of pirated software (Mosuela, 2016). This virus needed to be able to determine whether it was operating inside of a virtual machine, or whether it was being analyzed by Wireshark or Fiddler. This was necessary for the virus to be successful (Tibbals & Lowe, 2016).

vi. VPNFilter is a piece of malicious software that affects digital media security that was developed more recently. The previous year saw attacks on around 500,000 Internet of Things devices and routers. A cutting-edge technology of digital media protection was utilized in the construction of the command and control channel for this sophisticated piece of malicious software. After being activated on the compromised computer, the malicious software downloaded a number of images from the Photobucket website, which allows users to upload and share photographs. The IP addresses of the C&C servers were hidden in the photographs' EXIF metadata under the pretext of GPS latitude and longitude (Largent, 2018). The use of security on digital content is still the most common form of protection against malicious malware (Cabaj et al., 2018).

vii. The digital media security plan, on the other hand, suffers from not one but two significant faults that render it ineffectual. To begin, it is highly portable due to its diminutive size and low weight. There is a maximum number of hidden pieces of information that can be stored within a media file because doing so would

compromise the file's overall reliability. The length of time it will be effective for is the second consideration to take into account.

viii. It's possible that the files were saved on a server or in memory, making them accessible to forensics professionals who are looking into the matter (Zielinska et al., 2014). As a direct result of this, the process of developing and extending network security, which entails concealing data in a wide variety of network protocols, has increased significantly in recent years. Because of improvements in network security, it is now possible to send large volumes of information thanks to the steady flow of data that is maintained (Cabaj et al., 2018). If network data is not collected in real time, forensics professionals will have very little information to work with (Zielinska et al., 2014). Recently, malicious software has started utilizing network security techniques that have been there for decades. This is a relatively new development (Wendzel, Mazurczyk, Caviglione, & Meier, 2014).

ix. In August of 2011, researchers discovered two new forms of harmful software, both of which were considered to be firsts in the field of malware. W32 is a virus that can infect computers. Morto was able to take advantage of a security hole in the Remote Desktop Protocol (RDP) (RDP). After installation, it was successful in establishing a connection with its command and control server through the use of the Domain Name System (DNS) protocol (Drzymala, Szczypiorski, & Urbanski, 2016). Through the use of the DNS protocol, the Feederbot malware was able to interact with its command and control channel (Bureau & Dietrich, 2015). The Domain Name System (DNS) is an excellent safety measure for the internet. Because of the high number of packets it contains, it is possible to send enormous amounts of data. In addition, the performance of the network will suffer significantly if traffic filtering rules that are too stringent are implemented to this protocol. Because of this, it is far more challenging to prevent W32.Morto and Feederbot from spreading (Drzymala et al., 2016).

x. APT When it was found in 2013, Regin was largely regarded as being among the most sophisticated and creative pieces of malware ever developed. According to claims originating from Western intelligence agencies, Regin was a highly modular form of malware whose primary purpose was to gather information. These sophisticated payloads consisted of a wide variety of malware payloads that were able to be tailored to a particular target (Symantec Security Response, 2015). Several different Internet protocols were utilized in Regin's communications with its C&C server. These methods, which included the usage of ICMP packets and HTTP cookies, were utilized in order to acquire access to the system (Mazurczyk & Caviglione, 2015).

xi. More than 300 South Korean companies that had some kind of connection to the Pyeongchang Olympics were the intended victims of the fileless malware operation known as Operation Gold Dragon. Researchers in the field of information security uncovered Operation Gold Dragon around the end of December 2017, and discovered that it disguised dangerous code behind a script written in Microsoft PowerShell. Because of this PowerShell implant, the infected computer and the attacker's server were able to interact with one another across a backdoor channel that was encrypted. Through this route, messages were transmitted between the infected computer and the server (Sherstobitoff & Saavedra-Morales, 2018).

xii. Attacks from malware such as these are an excellent example of how NSM is put to use. The human senses can be tricked by digital media security, whereas traditional security only fools network nodes. This makes digital media security unique from network security (Mazurczyk et al., 2014). There is no significant difference between NSM and any other approach to safeguarding a network. Conventional network traffic is supplemented with covert data streams in order to shield users' information from being intercepted or modified in transit (Kaur, Wendzel, Eissa, Tonejc, & Meier, 2016). In order for NSM to be correctly disguised, a network carrier must fulfill both of the requirements listed above. Carriers have to have a good reputation

in order to prevent the flow of network traffic from appearing abnormal at first glance.

xiii. During the transmission of non-standard measurement data (NSM), there must also be no modifications identified in any network packets. This is also one of the requirements (Mazurczyk et al., 2014).

xiv. Utilizing security across networks and digital media allows attackers and authors of malicious software to infect the computers of their targets while going undetected for extended periods of time. This is possible thanks to the interconnected nature of the internet. The rise of malware that uses this strategy is especially troubling because commercial security solutions are woefully unprepared to identify or combat malware that uses this tactic (Cabaj et al., 2018). Malware developers have begun focusing on security as a defense mechanism against computer security measures that have improved in their own level of protection. These individuals have been successful employing even the most basic of security precautions. Recent attacks that combine these straightforward techniques with fileless or memory-resident implementations are significantly more challenging to identify and eliminate (Cabaj et al., 2018).

xv. Finding and removing potentially harmful information from the internet has become a task that is much more difficult to accomplish in this day and age of internet apps and cloud computing. P2P programs such as Skype and BitTorrent generate a large number of traffic channels, which can be leveraged to conceal network-based monitoring monitors from being discovered. Using NSM, it is possible to abuse the use of game consoles, cellphones, and devices connected to the internet of things (Wendzel et al., 2014). Both the potential and the stealth of covert communication have been increased thanks to the proliferation of increasingly complex non-standard ways of communication (Wendzel et al., 2014).

xvi. There are around 1200 programs in this category, the vast majority of which are available for free download. Finding software that provides security is not difficult (Barwise, 2018). As a direct result of the ease of access, IBM found that the number

of security malware attacks increased by a factor of six in September of 2017. To paraphrase what McMillen had to say: (2017), It is difficult, if not impossible, to develop effective defences against NSM since malware uses such a wide variety of security mechanisms. This makes it difficult for users to keep their data safe. Due to the pervasiveness of security features in malicious software, countermeasures that concentrate on a particular kind of security approach, such as network security, will be ineffective in resolving the issue (Cabaj et al., 2018).

xvii. Even though there are hundreds of ways to inject malicious data into network traffic, only a few of these techniques are frequently used. This is despite the fact that there are hundreds of ways to do so. 18) One of these is the procedure of inserting data into the headers of network protocols, which is a common practice. Altering the size of network packets in order to enable their use in the transmission of confidential information is another similar tactic (Wendzel et al., 2014). Both of these procedures are included within the scope of the subclass of covert channels known as storage channels (Dhamade & Panchal, 2015). The amount of time that passes between network packets, which is often referred to as inter packet gaps, as well as the sequence in which network packets are transmitted are commonly manipulated by NSM (Wendzel et al., 2014). This strategy is a subset of a wider group that is collectively referred to as the hidden channel timing channel. When these two distinct kinds of covert channels are combined, a new type of covert channel known as a hybrid covert channel can be produced (Dhamade & Panchal, 2015). The great variety of techniques that can be used to conceal data within network traffic makes NSM detection notoriously challenging (Wendzel et al., 2014).

xviii. Technologies such as code analysis and real-time traffic inspection have made network security procedures more complicated, which has made it more difficult to put them into practice. Additionally, their application has seen a rise in demand in recent years (Cabaj et al., 2018). When placed up against the great diversity of security approaches, security systems of every kind are put to the test. The protocols that are used to send information over the secret channels of a network can have just

as many variations as the networks themselves. Countermeasures can only identify a limited number of concealment strategies with any degree of accuracy (Wendzel et al., 2014). Traditional antivirus and blacklist solutions based on signatures are only able to provide protection against a subset of NSM malware threats due to the increasing sophistication of these attacks. [This phrase needs a reference] (Invernizzi et al., 2014).

xix. In spite of the fact that it has been put to use as a method of message encryption for millennia, modern researchers have only very lately rediscovered security and steganalysis. It is a widely held belief that prior to September 11, 2001, al Qaeda hid instructions for terrorist acts behind images that were uploaded to public websites. One year later, the arrest of members of the Shadowz Brotherhood, who had been selling child pornography via security, marked the successful conclusion of Operation Twins (Zielinska, Mazurczyk, & Szczypiorski, 2012).

xx. Following the publication of a paper by the National Institute of Standards and Technologies (NIST) in 2006, it was anticipated that malicious software would make advantage of security technology. According to this study, network security is more difficult to detect than digital picture security. As a result, a variety of countermeasures against network security should be developed by making use of the resources that are already accessible (National Science and Technology Council, 2006). Since the publishing of this study, the United States has upped the amount of work it puts into improving its cybersecurity in response to the breach that occurred in 2015 at the Office of Personnel Management (OPM). However, the security of government networks in 2018 is not much better than it was prior to the breach at the Office of Personnel Management in 2015. (Marks, 2018).

2.2.2 Computer Network Security Improvement

i. The practice of encrypting digital files extends back almost to the beginning of the development of computers themselves. In the 1970s, there were many different ways that digitized images might be used to encode information, and some of those

approaches are still in use today (Zielinska et al., 2012). During this time period, for instance, the LSB encoding method, which alters the image's least significant bit, was developed and is now utilized by a large number of people (Cabaj et al., 2018). The term "covert channel" wasn't coined to characterize the method of data theft utilized by mainframe computers until 1973. Prior to that, the phrase "covert channel" was used (Zielinska et al., 2012). During the 1990s, there was a rise in the use of computer networks, which necessitated the development of covert channels for the encryption of data used in network protocols (Wendzel et al., 2014).

ii. Security in the network is a topic that has only been around for a short while, but it is swiftly becoming one that is in high demand. The use of network security offers a solution to two of the most significant shortcomings of security applied to digital media. To begin, there is a cap on the total quantity of free space that can be utilized. There is a cap on the quantity of information that can be stored in media files before the quality suffers significantly. The following characteristic is that of permanence. Examining previously altered media files that have been saved on a server or in memory is something that experts in forensics may be able to do (Zielinska et al., 2014). Security on networks, which guarantees a steady stream of data, paves the way for the transfer of enormous amounts of data (Cabaj et al., 2018). Reconstructing network traffic that has not been caught in real time by professionals in the forensics field is impossible (Zielinska et al., 2014).

iii. It is possible for security communications to make use of the pervasive nature of modern communications networks. In general, for security communications to be successful, a carrier channel needs to have two features, and modern networks meet both of these requirements (Mazurczyk & Caviglione, 2016). Consider first whether or not there are airlines operating in your region as a place of departure. Because modern networks are widely interconnected and make use of a diverse set of network protocols, there are a great deal of carriers that are appropriate for security data transmissions. A high capability for concealment comes in second. It is possible to

conceal security communications quite well among the massive amounts of network traffic and data that are exchanged (Mazurczyk & Caviglione, 2016).

2.2.3 Network Security's characteristics

i. Network security makes use of three components of network communication in order to conceal information. To begin, failures in the communication route are normal, and as a result, updated protocol data units (PDUs) may be missed. The second reason is that the majority of network protocols include some type of redundancy, therefore it is quite unlikely that altering PDUs will cause disruptions to the flow of traffic. Thirdly, the manner in which a protocol is put into operation may typically be modified, at least to some degree. By taking advantage of this ambiguity, securitys can be concealed even when they are visible (Zielinska et al., 2012).

ii. It is feasible to classify several types of security technologies for networks according to the number of protocols that they make use of. One of the Open Systems Interconnection model protocols is modified by security used internally in the network. This technique of security is successful when the security payload is modified, the security payload is inserted, or securitys are injected into the PDU. Several different protocols are utilized in the process of inter-network security. By adjusting the time relations that exist between PDUs, this form of security can change the order of PDUs, cause them to be lost, or cause them to be delayed. Hybrid security combines the benefits of both traditional and modern (Zielinska et al., 2012).

iii. In the beginning phases of network security, TCP/IP stack protocols were utilized to conceal the network's security. Voice over Internet Protocol (VoIP), peer-to-peer services like Skype, social media networks, and cloud computing are just some of the extra capabilities that have been added to the Internet's specialized network topologies in recent years (Zielinska et al., 2012). These many different establishments and services all make use of a comprehensive arsenal of safety precautions (Zielinska et al., 2012).

2.2.4 Security Methods for the Online platform

i. There are three requirements that need to be satisfied before a security strategy can be classified as network-based (Mazurczyk, Wendzel, Zander, Houmansadr, & Szczypiorski, 2016). To get started, they make a change to one of the properties of the protocol. These changes take advantage of the inherent flaws in the communication protocol in order to address issues such as failures and delays, as well as to describe the type of information that is being transferred (for example, query results and file transfers). In addition to this, the message format that is going to be broadcast on a data transmission carrier might be altered with their help (for example, fragmentation and segmentation). As a result of this, the parties participating in security communication generate the adjustments in order to ensure that they are compatible with the typical operation of the protocol. It is consequently more challenging to recognize when anything has been modified. The third condition that a security strategy needs to fulfill in order for it to be successful is the following: (Mazurczyk et al., 2016).

ii. A classification system for the TCP/IP stack. The TCP/IP stack layer, which was the environment in which security communication was conducted, was the initial focus of research into network security. The several layers each offer unique options for data embedding (Mazurczyk & Caviglione, 2016).

iii. The layer that incorporates a wide array of distinct components and substances. Errors in transmission or delays in transmission can be introduced into the process of transmission by physical layer covert channels, which take advantage of network weaknesses for their own benefit. Given that these channels require access to network hardware in order to function, it follows that they cannot be utilized on a diverse collection of network devices. One of the most successful strategies for protecting the physical layer of a network is optical security, which involves changing the physical characteristics of a fiber optic link (Mazurczyk & Caviglione, 2016).

iv. The process that is used to transmit data. The data link layer of the TCP/IP stack is responsible for a number of important operations, including the organization of bits into frames and the monitoring of transmissions to prevent collisions. At this level, the padding between frames or the collisions between frames are utilized in order to conceal the security of network covert channels. It is important to keep in mind that covert channels can only be used in the immediate area because the data link layer functions on a hop-by-hop basis (Mazurczyk & Caviglione, 2016).

v. The networking layer of the stack. The TCP/IP network layer is in charge of addressing and routing network traffic. This is one of its responsibilities. Altering the speed at which packets are carried or hiding the security information in header fields that are otherwise unused are two of the most prevalent approaches to safeguarding this layer. Cyberattacks usually target this layer as their point of entry. On the network layer, covert communication can be carried out end-to-end thanks to the channels (Mazurczyk & Caviglione, 2016).

vi. a transport-related layer The TCP/IP stack's transport layer is responsible for managing how data is transmitted from one node to another over the network. At this level of the protocol stack, there are several header fields that can be exploited for the purpose of security encoding. Stream Control Transmission Protocol, also known as SCTP, is a popular choice for a target protocol at the transport layer due to the versatility of the protocol and its nearly ubiquitous application across operating systems (Mazurczyk et al., 2016).

vii. The layer of software that is installed on top of and operates in conjunction with the operating system. The application layer makes available a diverse assortment of covert communication channels. It supports an extensive variety of protocols and a comprehensive assortment of functions. The versatility that is provided by all of these different protocols paves the way for the creation of covert channels. One example of this is covert channels that take advantage of the lack of case sensitivity in the protocol fields (Mazurczyk et al., 2016).

viii. On the basis of an early classification, this location was selected to be the one where all security-related conversations were carried out. Despite the fact that security was built into every step of the network flow, we did not talk about it (Mazurczyk & Caviglione, 2016). This taxonomy places a significant emphasis on the TCP/IP stack; yet, despite this, it is not able to accurately represent more complex network services or applications. According to Mazurczyk et al. (2016), it was necessary to develop taxonomies that could be applied in a wide variety of settings.

ix. classification scheme that takes into account patterns Dr. Steffen Wendzel and his colleagues conducted research on 109 distinct approaches to establishing hidden channels in computer networks between the years 1987 and 2013. Wendzel, Zander, Fechner, and Herdin were the researchers who carried out this investigation (2015). After going through each of these methods, it was found that the data had eleven different patterns altogether. It was discovered that only four patterns were responsible for more than 70 percent of all of the strategies that were investigated. The research conducted by Wendzel and colleagues lacked the solid underpinnings in technical approaches that were present in previous studies. As a consequence of this, they drew motivation from the fundamental characteristics shared by these various approaches. [*] While helping with future study, the pattern-based taxonomy discovered commonalities among a wide variety of security approaches. This made the process of developing countermeasures much simpler (Wendzel et al., 2015).

x. Covert timing channels and covert storage channels are two distinct classifications that can be applied to each of the 11 patterns that were designed by Wendzel and his colleagues. The categories are denoted by the white boxes, while the gray boxes display the individual designs that can be found within each of those categories (Wendzel et al., 2015). Covert storage channels fall under this category, but Wendzel et al. did not investigate network covert channel options that alter payloads in their research (Wendzel et al., 2015).

xi. Carriers are able to have covert channels in patterns for covert storage channels established after being adjusted. It is possible, for example, to change the bits of a header element or PDU that are not currently being used. In order to construct a covert channel, it is necessary to adhere to specific covert time channel patterns. One example of this would be to rearrange the order or timing of PDUs (Mazurczyk, Wendzel, & Cabaj, 2018). Wendzel's work, which was published under the title Information Hiding in Communication Networks in 2016, expanded on the previous research by making use of hybrid methods. In a single implementation, these procedures use at least two different patterns each and every time. The two primary types of covert timing channels are known as protocol-aware methods and protocol-agnostic methods respectively. In both of these fields, having a comprehensive understanding of the carrier protocol is required. As a direct consequence of these differentiations being made, the taxonomy has been updated to include three additional patterns. Pattern 4 was also changed to accommodate the insertion of two more patterns, which enabled the taxonomy to expand even further (Mazurczyk et al., 2016).

xii. The pattern-based taxonomy that Wendzel developed in 2018 saw significant improvements. In the past, the taxonomy did not contain Storage Channels that Modified the Payload, but now it does. The entire number of patterns, which were then classified into user-data aware and user-data agnostic strategies. (Mazurczyk et al., 2018).

xiii. We have included the most recent description of each pattern because we want to make sure that you have access to the most latest information possible for each pattern. Patterns that can be classified as covert timing are denoted with the letters PT, whereas patterns that can be classified as covert storage are denoted with the letters PS. Example security representations are provided for each pattern.

xiv. Arrangements of time that are not seen. In this kind of covert channel, the encoding of confidential information is accomplished by manipulating the time of protocol

message transmissions. Because of intrinsic timing flaws, message loss, and jitter in the network, the security capacity for timing channels is significantly lower than the bandwidth for concealed storage channels. Finding and getting rid of time channels is a more difficult task than finding and getting rid of storage channels (Mazurczyk et al, 2016).

xv. Storage configurations that are not visible to the naked eye. In order to protect the confidentiality of the information being transmitted, this kind of covert channel modifies certain protocol fields. When it comes to the amount of bandwidth that is available for security, storage channels have a significant advantage over covert timing channels. It is typically more difficult to locate and remove time channels than it is to locate and remove storage channels (Mazurczyk et al, 2016). Both the original and updated versions of Wendzel's taxonomies differentiate between two distinct sorts of non-payload storage patterns: those that change the structure of the data and those that keep the structure of the data intact (Mazurczyk et al., 2016).

2.2.5 Against Network Security

i. Tools that perform code analysis and real-time traffic inspection have contributed to the evolution of network security solutions, which has resulted in the solutions' increased use (Cabaj et al., 2018). The sheer number of different security measures that can be implemented renders even the most sophisticated security protocols worthless. The transmission of confidential network channels is made possible by the protocols and flow configurations used in networks. In comparison, only a small number of the many different techniques to conceal one's identity can be uncovered by countermeasures with any degree of precision (Wendzel et al., 2014). Anti-virus and blacklisting software, despite their long-standing reputation as the major protection against malware, are only somewhat effective against NSM (Invernizzi et al., 2014).

ii. Protecting security network connections may require the adoption of countermeasures that fall into either the passive or active categories. When employing passive

countermeasures such as these, it is a lot simpler to uncover covert communication channels. Some examples of covert routes include the following: They can also be utilized to locate the originator or receiver of covert communications and bring them to justice.

iii. Active countermeasures are able to interrupt or otherwise change confidential communication in a variety of ways (Wendzel et al., 2015). Finding hidden channels can be accomplished with either passive or aggressive countermeasures, depending on the situation. Depending on the applications and protocols that are utilized for clandestine communication, as well as the security standards and regulations that are in place for the organizations that are impacted, there are a variety of methods in which contraceptive measures can be put into action (Mazurczyk et al., 2016).

iv. Detection. The uncovering of the hidden route is the first stage in the process of addressing network security concerns (Mazurczyk et al., 2016). Because the malicious code or scripts need to be identified, there is no way that NSM can use security that is unnoticeable to the target operating system or application (Wiseman, 2017). Methods of detection that are utilized frequently include signature-based detection, anomaly detection, and traffic analysis. In order to correctly identify an infection, initially a virus signature needs to be located and catalogued. Techniques of encryption and obfuscation that are not overly complicated can easily sidestep this kind of detection. Anomaly detection examines the typical flow of traffic on a network in order to identify previously unknown varieties of malicious software. On the other hand, this strategy frequently suffers from the problem of producing false positives. The length of an encrypted network connection, the amount of power it uses, and how long it will continue to use it can all be ascertained by traffic analysis (Zhong, Fu, Yu, Brooks, & Venayagamoorthy, 2017).

v. Elimination. When it comes to getting rid of hidden channels in a network that has countermeasures activated, there are two different approaches that can be used. The first thing that needs to be done is to ensure that no one can utilize any hidden

channels by securing the network and/or the host machine. Second, in order to avoid congestion, traffic needs to be managed (Mazurczyk et al., 2016).

vi. Although isolating a host does not remove a covert channel, doing so can be an efficient means of preventing command and control channels from functioning and preventing data exfiltration from taking place. As a consequence of this, this countermeasure is insufficient due to the fact that it is difficult to identify hosts that are infected (Mazurczyk et al., 2016). In firewalls, one common countermeasure involves blocking ports or protocols that are susceptible to attack. However, this countermeasure, especially if it is implemented in an overly restrictive manner, can have a severe influence on the performance of the network (Mazurczyk et al., 2016). In the case of the DNS protocol, for example, it is well knowledge that the successful operation of the protocol cannot be guaranteed through the implementation of stringent filtering methods (Drzymala et al., 2016).

vii. Because the DNS protocol is used by 91 percent of malicious software, it is not enough to just secure the network in order to prevent security on the network (Cisco, 2016).

viii. As part of the process of normalizing traffic, protocol communications are repackaged into their respective standard forms. When it comes to analyzing traffic, normalizers can either be stateless or stateful, depending on whether or not they remember and take into consideration previous packets (Mazurczyk et al., 2016). Congestion on the network is caused by the practice of traffic normalization, which, while an effective countermeasure against hidden channels, also has the disadvantage of being an impractical practice (Mazurczyk et al., 2016).

ix. In the context of video games, defenses against the assaults of an adversary. Researchers have come up with a diverse selection of strategies to circumvent the utilization of secure channels in computer networks. These countermeasures are, for the most part, only accessible in the form of proofs of concept and are unable to be utilized in the context of real-world networks. Hendra Gunadi and Sebastian Zander

of Murdoch University utilized an open-source network intrusion detection tool in 2017 to successfully find hidden network channels (NIDS). They started by conducting research on the three open-source NIDS apps that are the most well-known (Zander & Gunadi, 2017).

x. Snort. The network intrusion detection system (NIDS) Snort, which was developed by Cisco in 1998, is frequently considered to be the most effective in the business. This is accomplished by sniffing the network and then applying a predetermined set of rules to each individual packet. It is able to determine, using this method, whether or not a packet contains code that is malicious. Depending on the rules that are implemented, Snort has the capability of operating as either a signature-based or an anomaly-based detection system (Cooper, 2018). Gamers frequently select Snort as their activity of choice because of the user-friendliness of the game and the adaptability of its rules. The simplicity of Snort, on the other hand, makes it difficult for it to function well in the complicated network environments of today (Bricata, 2018).

xi. Suricata. When Suricata was first introduced in 2009, its primary purpose was to address the problems with network speed and complexity that Snort was experiencing at the time (Bricata, 2018). The Suricata system is a monitoring tool for application layers that is also capable of keeping an eye on lower-level protocols. Due to the fact that Suricata is compatible with the rules that Snort uses and that it receives support from the community (Cooper, 2018). Even though Suricata's rules are still easy to write, identifying network hidden channels might be a challenging task (Gunadi & Zander, 2017).

xii. Bro. The pace of the study has finally began to pick up for the first time since 1995, when it first received financing from the National Science Foundation, which is a full decade after it was initially initiated (NSF). As of 2018, (Bricata, 2018). Snort and Suricata are both examples of network intrusion detection systems (NIDS), but Bro possesses a deeper knowledge of network protocols (Cooper, 2018). Bro depends on

plugins that keep track of network traffic to limit network activity rather than using a set of user-defined rules to do so. They came to the conclusion that Bro was the most effective NIDS on which to build their covert channel capabilities. (Gunadi & Zander, 2017).

xiii. BroCCaDe was developed by Gunadi and Zander as an addition to Bro. It is a framework for locating hidden channels in network protocols, and it works very well with Bro. The covert channels Inter-packet timing, rate/throughput, payload field size modulation, and data corruption were tested with the help of a UDP network flow during the BroCCaDe development process. Approximately 98 percent of the time, BroCCaDe was successful in discovering the hidden channels (Gunadi & Zander, 2018).

2.3 OBJECTIVES IN RESPECT TO SECURITY

When documenting the needs for our security, what kind of structure should we use? That is a query that has not been resolved as of right now. Writing security criteria is no longer an artistic endeavor but rather an engineering one now that the internet has made it so widespread (which many can be trained to apply). Despite the fact that security requirements engineering has been increasingly popular in recent years, no one approach has yet managed to corner the market on this topic. Using the following method, security requirements engineering may be carried out in an efficient and unburdensome manner. This strategy will be utilized in the course project that you are responsible for.

2.3.1 Preliminaries

To begin addressing issues of system security, the first thing that must be done is to determine what the system in question actually is. When we speak of "recall security," we are talking to the state of having systems that only do the tasks for which they were designed. It is essential that we comprehend the function of the system before we get started with this. When contemplating system security, it is necessary to take into account the system's specification.

This specification derives its foundation from the functional criteria that must be met by the system. It is essential to keep in mind that functional requirements may be tested to see whether or not a system satisfies them, and this is a point that can be tested. When it comes to matters of safety and protection, we have the same goal in mind. The first step in resolving any concerns you may have regarding your level of security is to do a threat analysis. Determine the dangers that need to be addressed, along with their objectives and capabilities. Unreliable principles need to be taken into consideration in the same way that security cannot be dealt with in a vacuum.

2.3.2 Assets

When a system has assets that can be affected by attacks, there is a need for security. Included among assets are:

i. Such as printed money.

ii. Something that can't be touched, like money in your bank account.

Information is the primary asset of most computer systems. Additionally, any peripherals (such smart cards), backup disks, and so on could prove to be an asset. Because the system cannot directly safeguard or govern people, they are not typically considered assets in computer systems.

Having value for stakeholders is an asset's most important quality. Damage to an asset directly impacts the asset's value. Damage to an asset's reputation, for example, can have an indirect impact on its direct worth. Banks benefit from both the direct and indirect use of printed money. Because it has no money to lend, the bank suffers both financially and reputationally from a robbery (if the robbery is made public). Stakeholders and assets go hand in hand. Unless a stakeholder has a stake in the object, it's not an asset. To be a stakeholder, an entity has to value something in the system. A general "attacker" is not a stakeholder, even though he or she may have a strong interest in the target. The system may be attacked by legitimate stakeholders, such as unhappy employees.

2.3.3 Analyses That Do Damage

The protection of the assets contained within a system is one of the primary focuses of system security. When an action has the effect of decreasing the value of an asset, that action is referred to as being detrimental. Even while the thief may later gain from the worth of that cash, the bank is the one that loses out on the value of that cash because of the robber. In addition, the account balances of the bank are of little value to the bank as well as to its customers, who will most likely start looking for a new bank once the balances are removed from their accounts.

When a bank has fewer customers, the bank's revenue decreases. Concerns relating to security can be broken down into three categories: availability, integrity, and confidentiality. The loss of information poses a significant risk to the confidentiality of data stored in computer systems. It is possible to compromise the system's integrity by manipulating either the information, the resources, or the outputs. The availability of a product or service might be negatively impacted by a shortage of either input or output.

Disclosure, change, and deprivation are all behaviors that are not exclusive to digital systems; they can also occur in physical ones. When the components of a system are put through damage assessment, it is possible for potential issues to be identified and resolved. By coming up with as many potential ends to the following sentence as possible, we can make great headway in the assessment of the damage.

It is possible to cause damage to the object by doing certain actions on, to, or with it. It is possible to suffer a loss of revenue in a variety of ways, including "stealing money" and "erasing account balances." The majority of the time, the focus of damage analysis is on locating a group of triples that conform to a particular form (action, asset, harm). You can begin with the asset, and then brainstorm possible behaviors that could put its security at risk. By applying this method, you will be able to generate triples of this kind. It is highly recommended that you do so.

As a direct consequence of this, triples are produced. Before going on to the next phase, it is important to first consider any activities that would put the system's availability at risk, followed by any actions that might put the system's integrity at risk. Within the restrictions of the system that is in place for certain assets, it is possible that it will not be able to cause any of these CIA concerns any harm.

2.3.4 Goals Through Respect To Security

After doing a comprehensive risk analysis of the system, the only thing that is left to do is draft a list of security objectives. Consider the following instance of a security objective as an example: "Action on, to, or with asset" is what the system is meant to prevent or detect if it is functioning properly. Regarding the program, one may say either "the system shall prevent account balance erasure" or "the system shall prevent money theft." Both of these statements are true.

Because every goal must be connected to the maintainance of confidentiality or integrity, a security aim is a category of security property. It is important to keep in mind that the security goals of the system do not determine how the system protects itself. Both "the system shall use encryption to prevent the reading of messages" and "the system shall use authentication to verify user identities" are inadequate security goals.

The system should encrypt all communications to prevent anyone from reading them. When the software is in its final stages of development, it is very necessary for these implementation details to remain hidden from view. The objective that "the system shall be able to resist attacks" is inefficient since it does not specify which components of the system require protection and hence cannot be effectively implemented. In the "real world," it is impossible to accomplish all of the security goals. When designing a vault, one needs to keep in mind that there is always the risk of money being taken. However, the capabilities of any system can be overcome by an adversary that is sufficiently determined, resourceful, and skillful (as so many heist movies attest).

The system's risks should be considered while evaluating the system's security goals; if the risks warrant it, the security goals should be relaxed. For example, the objective of "preventing any robbery from a vault" could be rephrased as "resisting penetration for 30 minutes." This would make the aim more manageable. The ratings of locks used in the real world are determined using such criteria.) Another legitimate objective could be "finding evidence of theft from a vault." (In the real world, this objective is typically accomplished by the installation of security cameras).

2.3.5 Requirements For Care And Security

The development of systems is directed by security goals; nevertheless, these goals aren't detailed enough to be referred to as requirements because they lack sufficient precision. The systems' development is guided by security goals. A good security requirement would clarify what actions should be taken in a specific scenario, as opposed to a security goal, which would often say what actions should never be taken in that circumstance. This is the case because a requirement outlines what must take place in a given situation.To put it another way, a security requirement is anything that prevents the fulfillment of another need. This definition applies to both physical and information security.

In order for the restriction to be beneficial, it must be able to contribute to the achievement of a certain security goal. The requirements for the security of systems, rather than the purposes for which they were designed, place a higher emphasis on particular operations. It is possible to evaluate not only the significance of a practical necessity but also the necessary level of security. In addition, in order for financial institutions to fulfill their functional and safety requirements, they might need to give consumers the ability to "cash checks" and also "prevent the loss of money through incorrect checks."

This is an example of a security goal that should be followed. The convergence of these two lines of inquiry resulted in the formulation of rules governing the cashing of checks written on behalf of individuals:

i. If the teller works for the bank, the check has to be drawn on the bank that the teller works for.

ii. If you want to cash a check, the bank requires that you already be a customer there and that you put the check money into your account first.

The functional requirement is bound by security considerations in both of these examples, and both of these examples contribute to the achievement of the overall security purpose. (However, they do not provide an ironclad guarantee of achieving this objective. In actual fact, banks are unable to completely eliminate the possibility that bad checks would result in revenue loss).

When developing security requirements, one must ensure that each and every functional requirement F and each and every security aim G are taken into consideration. You can get to G by utilizing the constraints you've placed on F. In order for a real system to function as it should, there is an exceptionally large number of security criteria that need to be satisfied. It should not come as a surprise that actual systems struggle with security issues because it is so simple for developers to ignore some of these requirements.

In an ideal world, security criteria would not define how a system is created but rather how it reacts to a variety of different scenarios. On the other hand, because requirements need to be able to be evaluated, they might provide more explicit information on what the goals actually are. It is feasible to conceive of a scenario in which the system demands that the number I be transmitted from Bob to Alice, but at the same time, it stipulates that only Alice should be aware of its existence.

Within the context of this scenario, a security condition could be formulated along the lines of "the contents of m can never be accessible by anybody other than Alice." This is an example of a more esoteric kind of security requirement, like the one shown below. Despite the fact that this need cannot be demonstrated, it must nonetheless be satisfied. A more stringent safety criterion would be met if at least one of the following requirements were met:

i. Because it was transmitted via a safe channel, the private message labeled m can only be seen by Alice and Bob.
ii. Message m's decryption key is only known to Alice and Bob at this point.

Both of these preconditions are amenable to investigation and analysis. On the other hand, no one has made any excessive commitments with regard to the implementation particulars. The second criterion does not include any specifics like the encryption algorithm or the size of the key, for instance.

2.3.6 Iteration

It is not unusual for the process that came before it to be non-linear. [Case in point:] The establishment of new security requirements and objectives, new functional requirements, new assets, and so on will lead to the establishment of new security objectives. In the later stages of design and implementation, it's possible that new objectives and prerequisites may need to be outlined. In order to maintain the confidentiality of anything, you need establish a security requirement that mandates the utilization of an alternative authentication technique. In order to successfully execute this method, it may be necessary to include new system functionality, such as a prompt for entering a password. This additional functionality, in addition to introducing a new asset in the form of a password, necessitates the introduction of new security standards (the password checking routine should not display cleartext passwords, etc.). If you give it some thought, you might come to the conclusion that developing security criteria that accomplish an important security objective in addition to serving a practical function is an extremely difficult undertaking. After that, you will be forced to make a decision between diluting the purpose of your operation or modifying the prerequisites for its execution.

2.3.7 Requirements Vs. Goals

The distinctions between security goals and security needs are summarized in the table 2.1 below.

Table 2.1: Requirements vs. goals.

Requirements	Goals
Are limited in the scope	Are extensive in the scope
Apply to certain functional needs	Use throughout the system
System restrictions	Express desires
Are verifiable	Are untestable
may start providing design/implementation specifics	Nothing about design or implementation

2.4 SUMMERY

This investigation's primary focus was on researching security forms of malicious software Network Security Malware (NSM). How much of malicious software makes use of security in networks? How exactly does NSM cover up potentially damaging information when it encrypts network traffic? Are the countermeasures provided by the NSM proving to be effective at this time?

Some experts believe that instances of NSM in the wild are extremely uncommon, whereas others disagree (Barwise, 2018). When investigating how network security is being utilized in malware, the author discovered that many scholarly articles discuss security approaches that have been developed by researchers. This was one of the findings that the author came across. The names Regin and Linux appeared multiple times in numerous peer-reviewed works on the issue that purported to address real-world examples of NSM. Forkitor and Feederbot both. According to the author, the most important factor contributing to the proliferation of network security in malicious software is the absence of a standardized taxonomy. If the academic community and the security community adopt Wendzel's pattern-based taxonomy, it will be possible to make a more accurate identification of NSM and to take more effective preventative actions.

3. METHODOLOGY

3.1 INTRODUCTION

Wireless Sensor Networks (WSN) are used in different important applications, where they use a big number of sensor nodes deployed in large areas. The sensor nodes are used to collect, process, transmit and receive data. The transmitted data is in the form of signals that are transmitted between the sensor nodes, making them vulnerable to security risks and consuming high energy and longtime [17].

Providing protection for WSN data requires a high-security system that consumes low power and low time to fit the constraints of these networks which considered a big challenge. In this thesis, a hybrid security technique has been designed using two algorithms, the asymmetric key algorithm used in generating keys, and the symmetric key algorithm used in data ciphering. The proposed technique is provided high security and it is consuming low energy and low time.

3.2 SECURITY OF WSN

Network security is the method used to provide the maximum possible protection of information in networks because of internal or external risks that threaten the security of that the information stored in those networks can only be accessed by individuals who have access to information [4].

The continued development of information technology has led to increasing the use of networks (especially WSN). WSN is used in many applications to transmit important information such as medical, environmental, and military information. It may sometimes be necessary to set up the network in enemy territory in order to collect the required military information, which would expose it to various attacks.

All these reasons have made security information one of the basic requirements for WSN. The presence of restrictions on WSN has led to restrictions on the methods and algorithms used to protect this type of network. The task of protecting WSN is therefore very difficult [4].

There are some important requirements that must be met in every security system of WSN. These requirements are including:

i. Information Confidentiality: It means not allowing unauthorized persons access to the secret information. The type and confidentiality of the information vary according to the application who uses that information and the policy applied by the application itself. Examples of this type of information that requires high confidentiality: personal medical information of individuals, military and intelligence information of countries and military sites of the country [4].

ii. Information Integrity: Sometimes the main requirement is to maintain the integrity of information from fraud and change after it becomes publicized. A particular entity may announce important information concerning it, which requires the safety of that information from manipulation. In such cases, the safety of the information should be maintained. For example the deletion of digital orders from the number, which causes significant financial losses [4].

iii. Information availability: One of the most important requirements of the basic information industry is delivering information to authorized persons in a timely manner. Since it is useless to maintain the confidentiality and integrity of information if unauthorized persons are unable to obtain it. Hence, the importance of the third requirement, which is considered a fundamental requirement and a complement to the previous demands, is clear. Providing confidential information to authorized persons through safe and fast channels is an important requirement. Some attackers use different means to prevent authorized users from accessing information. Some of these means include deleting or destroying information or destroying devices that store that information before the authorized persons arrive [18].

iv. Information Reliability: It means that the recipient of the information clarifies that the information comes from the reliable source authorized to send. The purpose of authentication used to ensure that the information comes from the source required to ensure its integrity and non-vulnerability [18].

3.3 ENCRYPTION IN WSN

Encryption is the method in which text or any type of data is converted from readable data to encrypted data that can only be decrypted by someone who has the decryption key. Encryption is one of the most important methods to provide data security, especially to protect end-to-end data sent over networks [2].

The purpose of the encryption is to ensure that the privacy of the data is not allowed to tamper with or view it, because it is either confidential or private and very important. Anyone cannot be understood the content of this information or messages only those who have their secret key (key encryption). Encryption keys are used in encryption and decryption operations. These keys are calculated using complex mathematical formulas (algorithms). On the other hand, decryption is the process of re-converting the encrypted data to its original form, using the secret key [2].

The data encryption algorithms used in WSN depending on the key type of encryption are usually divided into two main types: asymmetric key algorithms and symmetric key algorithms [8].

3.4 ASYMMETRIC KEY ALGORITHM

It is also called (Public Key Algorithm), it uses two different keys, (public key, and private key) which are linked mathematically. The public key is used for encryption by the sender and the private key is used for decryption by the receiver. The sender transmits its public key to the receiver after encrypting the message, but only the recipient's private key, which remains secret, is capable of decrypting it [2].

The advantage of this type of ciphering that it is more secure than symmetric ciphering because it uses a key for deciphering which is different from a cipher key. The disadvantage of asymmetric ciphering is consuming more energy and memory storage because it consumes more time and energy in computation operations of generating the keys. Asymmetric algorithms have many algorithms such as ECC, RSA, and ElGamal. In this work, the ECC algorithm is implemented because it is considered the best in terms of confidentiality [2].

The asymmetric algorithms can be classified into three systems: algorithms based on Integer Factorization System (IFS) like RSA, algorithms based on Discrete Logarithm System (DLS) like Diffie-Hellman, and algorithms based on Curve Discrete Logarithm System (CDLS) like ECC [19].

Elliptic Curve Cryptography (ECC) is the more secure public key cryptography algorithm because it uses fully exponential computations to solve the problems. It is used to generate smaller, faster, and more efficient cryptographic keys. ECC generates the keys through the properties of an elliptic curve equation instead of the conventional generation method as a result of very large initial numbers. Because ECC helps create equivalent security with less power to use the computer and use battery resources, it's becoming widely used for mobile applications [20][21].

An elliptic curve equation is:

$$y^2 = x^3 + ax + b \tag{2.1}$$

Where:

x, y, a, and b are numbered in a finite field.

Both the public key and private key must be generated. The sender will be encrypting the message with the public key of the recipient and the receiver will be decrypting it with his private key [20]. Equation (2.2) used to generate the public key.

$$Q = d * P \tag{2.2}$$

Where:

d: The random number that selected within the range of (1 to n-1), it is represented private key.

n: the normal numbers. P: the point on the curve. Q: the public key.

Then a number (d) within the range of (n) must be selected [20]. Figure (2.1) shows an elliptic curve example.

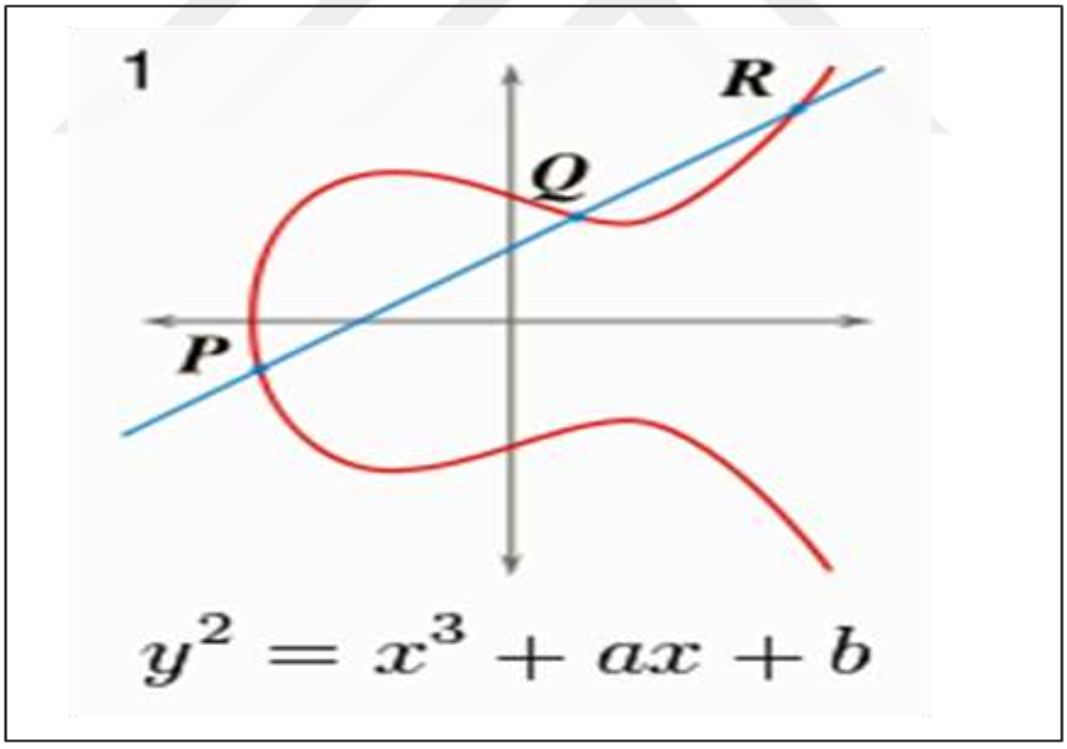

Figure 3.1: An elliptic curve example.

3.5 SYMMETRIC KEY ALGORITHM

It is also called (secret key algorithm). It uses a single key. Sometimes referred to as a (shared secret algorithm) because the encrypting system must share it with any object. It intends to be able to decrypt the encrypted data. The most widely used symmetric key ciphering algorithm is the Advanced Encryption Standard (AES) [2].

Generally, symmetric key encryption is much faster than asymmetric encryption, but the key used to encrypt the data must be exchanged by the sender with the recipient before performing decryption on the ciphertext by the recipient [2].

The advantages of this type of ciphering are the high speed of ciphering, consuming low energy and low memory storage. The disadvantage is the problem of key distribution [2]. Encryption of data using symmetric key algorithms requires one secret key for each data interchange. The greater the number of parties exchanging data, the greater the number of cryptographic keys used. Increasing the number of secret encryption keys consumes large memory to store these keys and the possibility of hacking these keys becomes more dangerous. To solve this problem, scientists resorted to using public key cryptographic algorithms. Symmetric algorithms have many algorithms such as AES, Blowfish, and DES. In this work, the AES and Blowfish algorithms are implemented because they are considered the best in terms of confidentiality and energy consumption [22].

3.6 AES ALGORITHM

The Advanced Encryption Standard (AES) is one of the most symmetric block cipher algorithm that used commonly in ciphering. When encrypting data by AES algorithm it is very difficult for hackers to get the original data. There are three key sizes allowed for the AES algorithm: 128, 192, and 256 bit; but the block size of the message allowed is fixed which is 128 bit [23]. AES based on a principle called the substitution-permutation. The AES algorithm consists of a number of rounds depending on the key length. Thus, AES with a key length 128 bit consists of 10 rounds; 192-bit key length consists of 12 rounds, and 256-bit key length consists of 14 rounds. Every round consists of the following operations:

Add round key, Sub bytes, Shift rows, and Mix columns [12]. Algorithm (2.1) demonstrates the AES algorithm [24].

Algorithm (2.1): AES Encryption Algorithm

Input*:* X: array (4 rows of bytes) of the input message

Output: Y: array (4 rows of bytes) of the cipher

1 *Begin*

2 *State = X*

3 *Add round key (State, Key0)*

4 *For r = 1 to (N_r - 1) à (Where N_r: the number of rounds = 10 or 12 or 14 depending on the length of key)*

5 *Sub bytes (State, S-box)*

6 *Shift rows (State)*

7 *Mix columns (State)*

8 *Add round key (State, Key$_r$)*

9 *End for*

10 *Sub bytes (State, S-box)*

11 *Shift rows (State)*

12 *Add round key (State, Key$_{Nr}$)*

13 *Y = State*

14 *End*

3.7 BLOWFISH ALGORITHM

Blowfish is a symmetric-key block cipher which considered one of the most common Feistel network cryptosystems. It is introduced as a new encryption standard algorithm. Blowfish consists of a 16 round, which uses large key-dependent Substitution boxes (S-boxes). Blowfish block size is fixed which 64 bits, but the key size may differ from 32 bits up to 448 bits. It provides a good encryption rate in software and there is no effective cryptanalysis of it has been found to date [25].

Algorithm (2.2) demonstrates the Blowfish encryption algorithm [26]:

Algorithm (2.2): Blowfish Algorithm Input: M: Message

Output: *C: Ciphertext*

i. Begin
ii. M is divided into two parts ML (Message Left), and MR (Message Right)
iii. For i=1 to 15
iv. ML= ML XOR MR
v. MR= F(ML) XOR MR
vi. Swap ML and MR
vii. End For
viii. Swap *ML and MR*
ix. *ML= ML XOR P17*
x. *MR= MR XOR P18*
xi. *C= combine ML, MR*
xii. End

3.8 CIPHER MODES

The block cipher mode is used in order to encrypt a plaintext with size more than one block.

The purpose of block cipher modes is encryption and authentication [27].

3.9 CHAINING BLOCK CIPHER MODE

To encrypt data using Chaining Block Cipher (CBC) mode the first block of plaintext must be firstly xor with the initial vector, then the resulted block is ciphered by the key to producing the cipher block. The next block is xor with the previous block and the resulted one is ciphered by the key and hence. The initial vector is generated randomly, and it is used for encryption and decryption. This mode is used for ciphering and authentication [27].

3.10 COUNTER MODE

To encrypt a plaintext using the counter (CTR) mode a nonce must be chosen and a counter must be used. The counter must be incremented for each block. Usually, the nonce is chosen randomly and it is used in encryption and decryption [27].

3.11 CCM MODE

The CCM mode consists of CBC and CTR modes. It is used with block cipher size 128- bit only. In order to encrypt a message with CCM mode, the CBC mode must be used firstly for authentication and the CTR mode is used then for encryption. This mode is provided with authentication and strong security [27].

3.12 ENERGY OF WSN

WSN consists of a big number of sensor nodes. A small battery inside the sensor supplying the power is limited. The power in a sensor node is consumed in three types of activities: sensing, processing, and radio operations such as receiving and transmitting. The radio operations of these three activities take up the majority of the power [28].

The requirements of security in WSNs consist of three major aspects: confidentiality, integrity, and authentication. Sometimes availability is added to the previous requirements. In order to provide these requirements of security, ciphering is typically applied to the data and as a result, additional energy cost is presented. The energy cost includes the cost of computation and the cost of communication.

The cost of computation (which means the energy consumed by implementing the security system) is affected by the ciphering algorithm. Therefore, the limited energy of a sensor node is a challenging constraint in WSNs [29].

For this reason, low energy consumption and a highly efficient algorithm must be used. The previous works have been focused on studying ciphering algorithms which have a direct impact on the computational energy cost in a WSN, by analyzing existing ciphering algorithms or proposing new ciphers with efficient energy [29].

Since asymmetric algorithms consume high power, while symmetric algorithms have a low-security level, the use of a hybrid system that combines the advantages of both types has been suggested [30].

The Power Scope was used to measure the consumed energy by the proposed technique in the present study. The Power Scope is a tool which performs statistical sampling of system activities. It computes the consumed energy by determining the fractions of energy consumed during a certain period or in individual procedures [31].

3.13 NETWORK SIMULATOR

The environment which used to simulate the computer network behavior is called a network simulator. It is used to providing a perfect understanding of the behavior of the system because the communication networks have become very complicated for traditional analytical methods. The network of the computer in simulators is modeled with devices, links, applications, etc. and the performance of the network is recorded.

Simulators come with support for the most commonly used techniques and networks nowadays, for example, Wireless LANs, Wireless Sensor Networks, mobile ad hoc networks, Internet of Things. There are many types of simulator programs such as NS1, NS2, NS3, OMNET [32].

NS2 is used to simulate network scenarios and analyze its outcomes based on numerous network factors such as arrived packets successfully, losing packets, delay, error, intrusion, etc. It usually runs on the Linux operating system. It provides substantial support for simulation of routing, multicast techniques and IP techniques, such as UDP, TCP, and SRM over wired and wireless networks. It has many benefits that make it a suitable tool, such as supporting many protocols and the ability to graphically detailing network traffic [32].

3.14 RANDOMNESS TEST

Random sequences and random numbers are a necessary part of ciphering. Many ciphering techniques rely on random values. The ciphertext of the higher random has more security. Randomness is measured by statistical tests. Therefore, the security evaluation of ciphering algorithms depends largely on statistical random tests [33][34].

Numerous useful measures of randomness have been used to measure the randomness of a binary sequence. The randomness measures consist of events based on statistical tests, transforms, complexity or a mixture of these. The most commonly used test introduced by Marsaglia is called the Diehard Test [35].

The Diehard tests are a collection of statistical tests used to measure the quality generator of random numbers. They have been developed by George Marsaglia over several years and first published in 1995 [36].

The tests of Diehard are:

i. Birthday spacings.

ii. Overlapping permutations.

iii. Ranks of matrices.

iv. Monkey tests.

v. Count the ones.

vi. Parking lot test.

vii. Minimum distance test.

viii. Random spheres test.

ix. The squeeze test.

x. Overlapping sums test.

xi. Runs test.

xii. The craps test.

The output of the Diehard is groups of p-values which belong to the interval [0,1). The interval in the method is dividing into three areas: (safe, doubt and failure). The limits of the three areas are demonstrated in Table (2.1).

Table 3.1: Limits of safe, doubt and fail areas of the diehard test.

Safe Area	(0.25 > p-value <= 0.75)
Doubt Area	(0.1 > p-value <= 0.25) and (0.75 > p-value <= 0.9)
Fail Area	(0 > p-value <= 0.1) and (0.9 > p-value <= 1)

The p-values can be lying in one of the three regions (failure, doubt, and safe). Whenever the failure region contains a large number of points that means the randomness is low. On the other hand, if the points are closer to the safe region, the randomness will be higher [37].

3.15 THE PROPOSED WORK

Wireless Sensor Network uses the AES algorithm to implement its security technique. The proposed technique MECC uses multiple ECCs to generate multiple secret keys of AES for improving the security of WSN. The using of the Blowfish algorithm instead of AES produces another technique MECC- Blowfish. The MECC-Blowfish technique improves WSN performance, lifetime and security. The proposed MECC-Blowfish technique consists of MECC technique and the Blowfish algorithm. Figure (3.1) shows a block diagram of the proposed hybrid security technique.

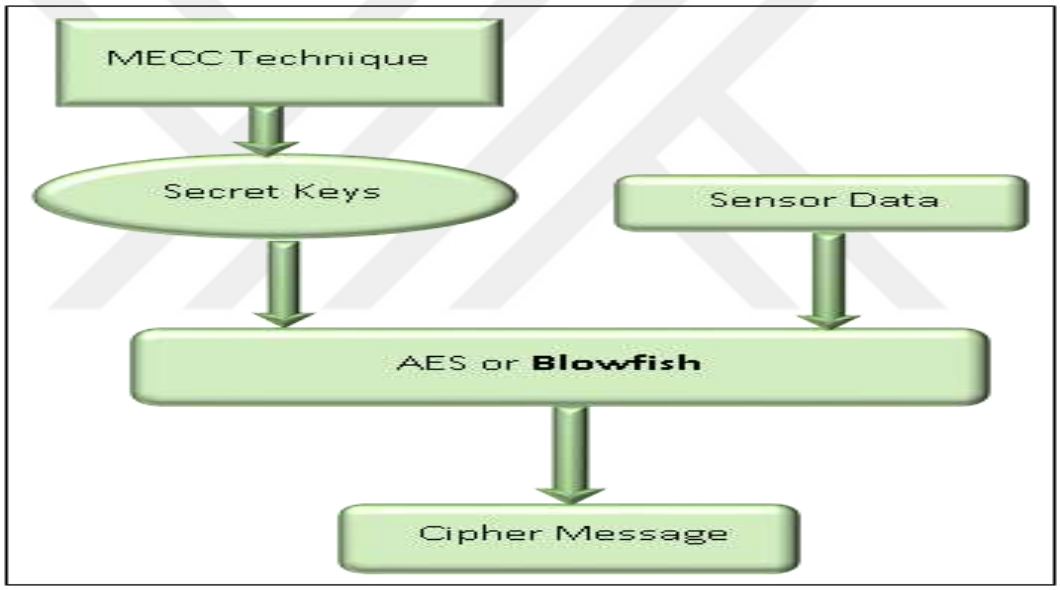

Figure 3.2: Block diagram of the proposed hybrid encryption technique*.*

3.16 MECC TECHNIQUE

The proposed MECC technique can generate N keys in three models using multiple ECC public keys. In every model, a different number of initial sequences and different ECC public keys are used. The generated keys by the MECC technique are used as secret keys for the symmetric algorithm of the WSN hybrid security technique. The three models of the proposed MECC technique are.

3.17 TWO SETS KEY GENERATOR

In the first model, two initial random key seeds r_0 and s_0 are used to generate two sets of keys: $r_1, r_2 \ldots r_n; s_1, s_2 \ldots s_n$, where the sets are generated according to the Equations (3.1) and (3.2).

Then, the xor function $\boldsymbol{f}$ equation (3.3) is used to obtain the final keys Key_i, which are used in the encryption and decryption process.

$$r_i = ECCen\,(TCk1,\ r_{i-1}) \qquad 0 < i \leq n \tag{3.1}$$

$$s_i = ECCen\,(TCk2,\ s_{i-1}) \qquad 0 < i \leq n \tag{3.2}$$

Where:

$\boldsymbol{ECCen}$ is elliptic curve ciphering (ECC algorithm).

$\boldsymbol{TCk1}$ and $\boldsymbol{TCk2}$: are trust center public keys which are different keys.

The trust center must satisfy the values of r0 and s0 in the sender and receiver sides.

$$Key_i = f\,(k_i, r_{n-i+1}) \qquad 1 \leq i \leq n \tag{3.3}$$

The diagram of the proposed MECC technique with two sets key generator model is demonstrated in Figure (3.2).

Algorithm (3.1) shows the proposed MECC technique with two sets key generator model.

Algorithm (3.1): MECC Algorithm (Two Sets) Input: *s0, r0; key sets*

PK1, PK2; generated public keys

Output*: K1; proposed generated keys*

i. Begin

ii. *For i= 1 to n (Where n: is number of the secret keys)*

iii. *si = ECC (si-1, PK1)*

iv. *ri= ECC (ri-1, PK2)*

v. *End for*

vi. *For i= 1 to n*

vii. *K1i= f (si, rn-i+1)*

viii. *End for*

ix. End

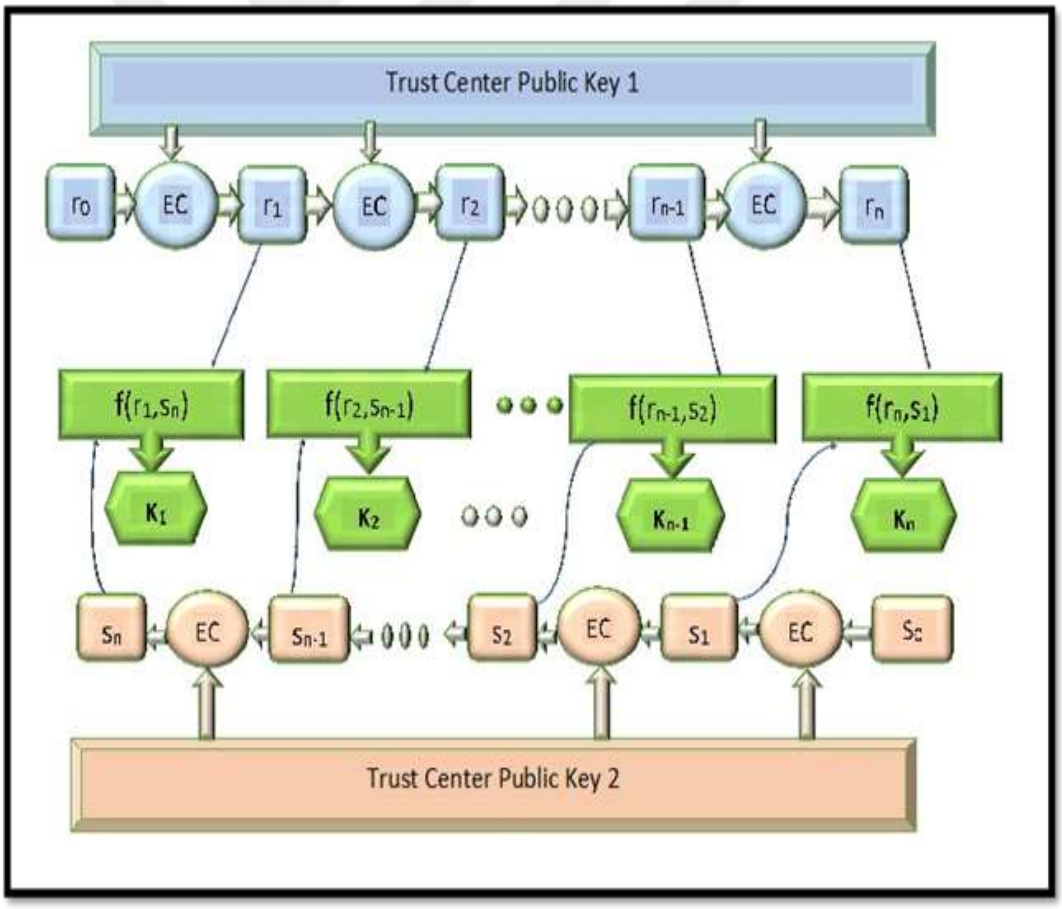

Figure 3.3: The diagram of the proposed MECC technique with two sets key generator model.

3.18 THREE SETS KEY GENERATOR

For more randomness that increases the security, three initial random key sets r_0, s_0, and m_0 are used to generate three sets of keys: $r_1, r_2 \ldots r_n$; $s_1, s_2 \ldots s_n$; $m_1, m_2 \ldots m_n$. The first and second sets are generated according to Equations (3.1) and (3.2), respectively.

The third set was generated according to Equation (3.4).

Then, the xor function $\boldsymbol{f}$ is used to obtain the final keys Key_i as in the Equation (3.5).

$$m_i = ECCen\,(TCk3, m_i - 1) \qquad 0 < i \leq n \tag{3.4}$$

Where:

$\boldsymbol{TCk3}$ is trust center public key.

$$Key_i = f\,(s_i, r_n - i + 1, m_i) \qquad 1 \leq i \leq n \tag{3.5}$$

The diagram of the proposed MECC technique with three sets key generator model is demonstrated in Figure (3.3).

Algorithm (3.2) shows the proposed MECC technique with three keys generator model.

Algorithm (3.2): MECC Algorithm (Three Sets) Input*: s0, r0, m0; key sets*

PK1, PK2, PK3; generated public keys

Output*: K2; proposed generated keys*

i. Begin

ii. *For i= 1 to n (Where n: is number of the secret keys)*

iii. s_i *= ECC (*s_i*-1, PK1)*

iv. r_i*= ECC (*r_i*-1, PK2)*

v. m_i*= ECC (*m_i*-1, PK3)*

vi. *End for*

vii. *For I = 1 to n*

viii. *K2i= f (si, rn-i+1, mi)*

ix. *End for*

x. End

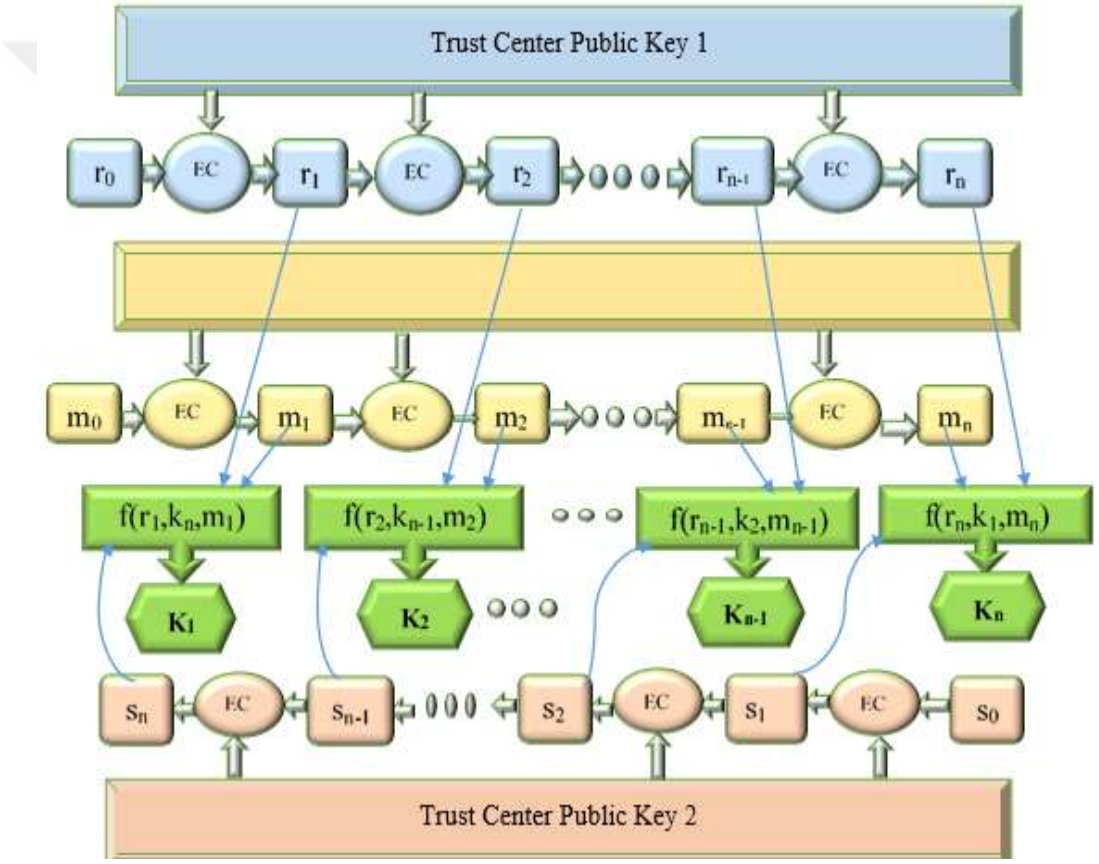

Figure 3.4: The diagram of the proposed MECC technique with three sets key generator model.

3.19 FOUR SETS KEY GENERATOR

For more security, four initial random key seeds r_0, s_0, m_0, and g_0 are used to generate four sets of keys: $r_1, r_2 \ldots r_n$; $s_1, s_2 \ldots s_n$; $m_1, m_2 \ldots m_n$; $g_1, g_2 \ldots g_n$. The first, the second, and the third sets are generated according to Equations (3.1), (3.2) and (3.4), respectively, and the fourth set was

generated according to the Equation (3.6).

Then, the xor function $\boldsymbol{f}$ is used to obtain the final keys Key_i as in the Equation (3.7).

$$g_i = ECCen\,(TCk4, g_{i-1}) \qquad 0 < i \leq n \tag{3.6}$$

Where:

$\boldsymbol{TCk4}$ is trust center public key.

$$Keyi = (s_i, r_{n-i+1}, m_i, g_{n-i+1}) \qquad 1 \leq i \leq \tag{3.7}$$

The diagram of the proposed MECC technique with four sets key generator model is demonstrated in Figure (3.4).

Algorithm (3.3) shows the proposed MECC technique with four sets key generator model.

Algorithm (3.3): MECC Algorithm (Four Sets) Input: *s0, r0, m0, g0; key sets*

PK1, PK2, PK3, PK4; generated public keys

Output*: K3; proposed generated keys*

i. Begin

ii. *For i= 1 to n (Where n: is number of the secret keys)*

iii. *si = ECC (si-1, PK1)*

iv. *ri= ECC (ri-1, PK2)*

v. *mi= ECC (mi-1, PK3)*

vi. *gi= ECC (gi-1, PK4)*

vii. *End for*

viii. *For i=1 to n*

ix. *K3i= f (si, rn-i+1, mi, gn-i+1)*

x. *End for*

xi. End

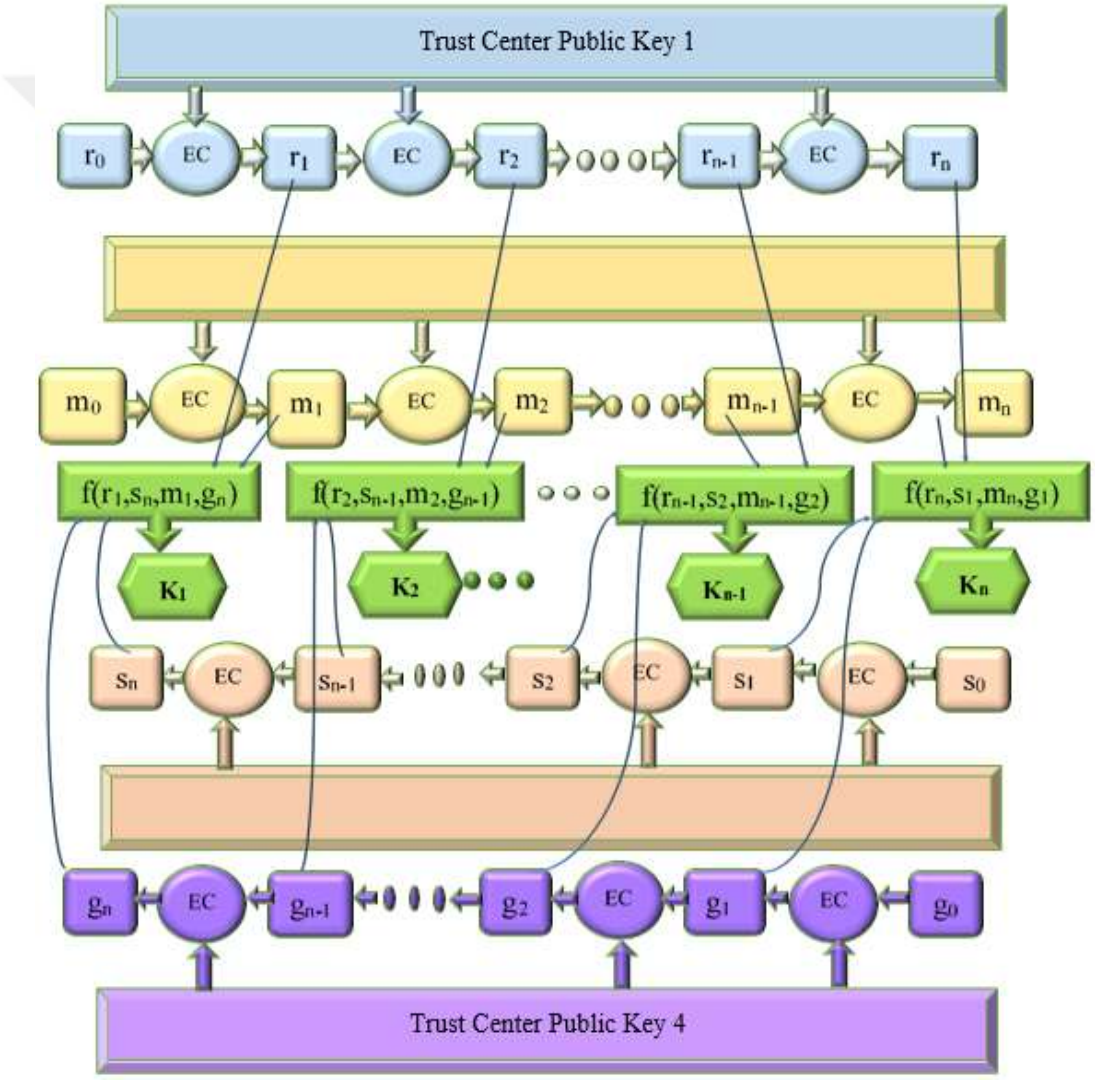

Figure 3.5: The diagram of the proposed MECC technique with Four sets key generator model.

3.20 RANDOMNESS ANALYSIS OF MECC TECHNIQUE

One of the measures used to compute the strength of the encryption system is the randomness of the ciphertext. In order to evaluate the randomness of the secret keys generated by the proposed MECC technique, four experiments are executed. The MECC technique is executed four times to generate 10 secret keys each time. The secret keys are generated by four different models and then saved in four separated files. The Diehard test is used in all experiments to test the randomness of the four files of keys in order to evaluate the confidentiality of the MECC technique.

All the files that used in the experiments are binary files with size 10 MB. The first file contains secret keys generated by the MKP technique. The second file contains secret keys generated by the MECC technique using two different ECCs' public key. The third file contains secret keys generated by the MECC technique using three different ECCs' public key. The fourth file contains secret keys generated by the MECC technique using four different ECCs' public key.

3.21 MECC-AES TECHNIQUE

The MECC-AES technique is suggested to evaluate the strength of the security system that is available when using keys generated by the MECC technique. The MECC-AES technique consists of a MECC technique that generates secret keys and the AES algorithm which uses these secret keys in the encryption and decryption operations.

3.22 MECC-AES ENCRYPTION

In order to encrypt the plaintext by the MECC-AES technique, the plaintext must be divided into n blocks, as demonstrated in Equation (3.8). Then these blocks are distributed in groups. The number of these groups is equal to the number of secret keys. All blocks belonging to the same group i are encrypted by the secret key Keyi. The encryption operation using AES algorithm is demonstrated in Equation (3.9).

$$M = \sum_{i=1}^{n} B_i \quad (3.8)$$

Where:

M: Message.

B: Block of the message.

$$C_i = AES_{Keyi}(B_i) \quad (3.9)$$

Where:

C: Ciphertext.

AES: AES algorithm.

Key: Secret key which is generated by the MECC technique.

Figure (3.5) shows a general view of the proposed MECC-AES encryption technique.

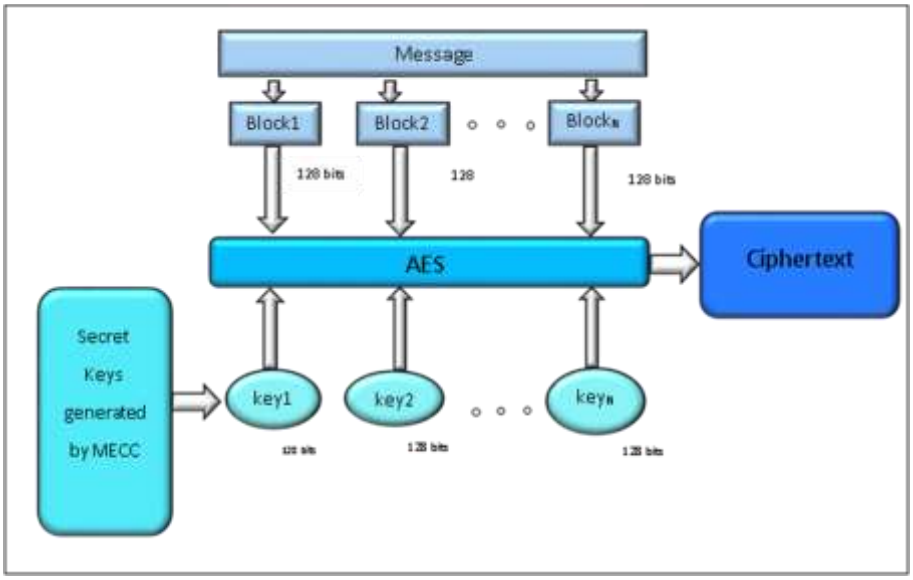

Figure 3.6: The roposed MECC-AES encryption technique.

Algorithm (3.4) demonstrates the encryption algorithm of the proposed MECC-AES technique.

Algorithm (3.4): MECC-AES Encryption Algorithm Input: *Message; secret message*

Key1 or Key2 or Key3; secret keys (where: Key1: is generated by the algorithm (3.1) Key2: is generated by the algorithm (3.2)

Key3: is generated by the algorithm (3.3) (only one key used depending on the selected algorithm that used for generation))

Output*: C1 or C2 or C3; the ciphertext (depending on the selected secret key)*

i. Begin

ii. Block-size = 128

iii. No-of-Blocks= Message / block-size

iv. No-of-Blocks-in-Group= No-of-Blocks / n (Where n: is number of the secret keys)

v. i=1

vi. For j= 1 to No-of-Blocks

vii. C1j= AES (Bj, Key1i)

viii. C2j= AES (Bj, Key2i)

ix. C3j= AES (Bj, Key3i)

x. If (j mod (No-of-Blocks-in-Group) = 0) then

xi. i=i+1

xii. End If

xiii. End For

xiv. End

3.23 MECC-AES DECRYPTION

In order to decrypt the ciphertext which encrypted by the MECC-AES technique to retrieve the original plaintext, the secret keys generated by the MECC technique are used. Each block of the plaintext B_i can be retrieved by decrypted it by the AES algorithm using the secret key Key_i used for encrypting that block. The MECC- AES decryption is demonstrated in Equation (3.10).

$$B_i = AES_{Keyi}\ (C_i) \tag{3.10}$$

3.24 MECC-BLOWFISH TECHNIQUE

To increase the confidentiality of the WSN security system, the Blowfish algorithm has been used to build a proposed hybrid security technique. The proposed technique is called MECC-Blowfish technique. It includes the MECC technique that is used for generating secret keys and the Blowfish algorithm which used for encryption and decryption. The Blowfish algorithm is used because it has more security compared with the AES algorithm.

Figure (3.6) shows a general view of the proposed MECC-Blowfish hybrid security technique for encryption.

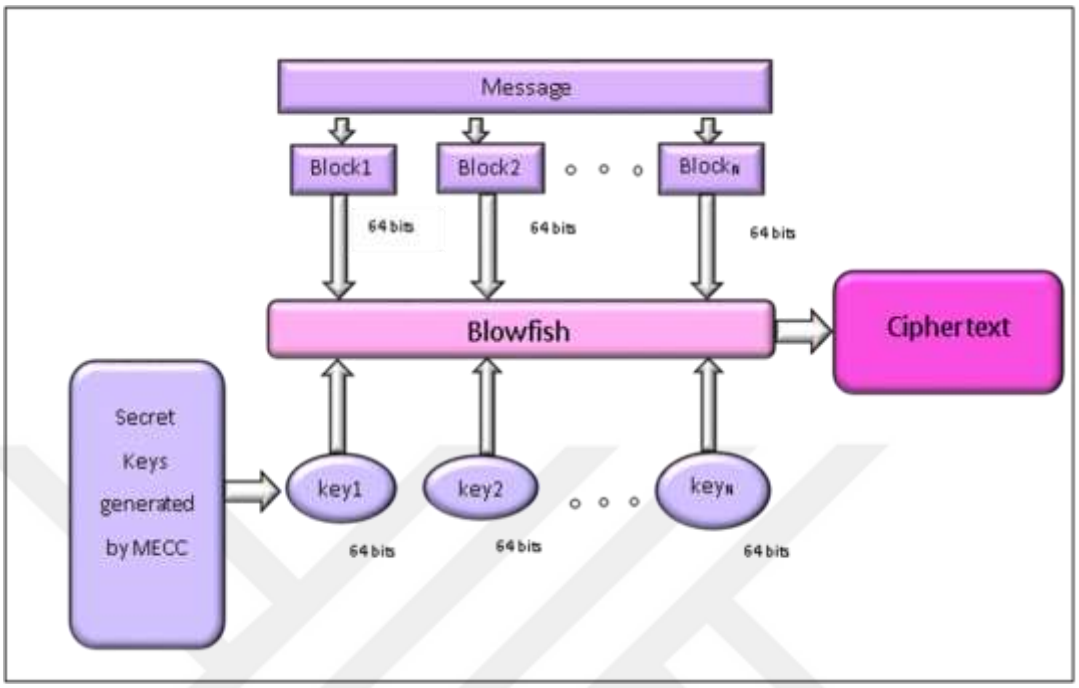

Figure 3.7: The MECC-lowfish technique**.**

3.25 MECC-BLOWFISH ENCRYPTION

In order to encrypt the secret message by the MECC-Blowfish technique, the message must be divided into n blocks, as demonstrated in Equation (3.8). Then these blocks are distributed in groups. The number of groups is equal to the number of secret keys. All blocks $\boldsymbol{Bi}$ belonging to the same group i are encrypted by the secret key Key_i. The encryption operation of the Blowfish algorithm is demonstrated in Equation (3.10).

$$C_i = Blow_{Keyi}(B_i) \tag{3.10}$$

Where:

$Blow$**:** Blowfish algorithm.

Algorithm (3.5) shows the proposed MECC-Blowfish technique encryption algorithm.

Algorithm (3.5): MECC-Blowfish Encryption

Algorithm Input***:*** *Message; secret message*

Key1 or Key2 or Key3; secret keys (where:

Key1: is generated by an algorithm (3.1)
Key2: is generated by an algorithm (3.2)

Key3: is generated by an algorithm (3.3) (only one key used depending on the selected algorithm that used for generation))

Output*: C1 or C2 or C3; the ciphertext (depending on the selected secret key)*

i. *Begin*
ii. *Block-size = 64*
iii. *No-of-Blocks= Message / block-size*
iv. *No-of-Blocks-in-Group= No-of-Blocks / n (Where n: is number of the secret keys)*
v. *i=1*
vi. *For j= 1 to No-of-Blocks*
vii. *C1j= Blow (Bj, Key1i)*
viii. *C2j= Blow (Bj, Key2i)*
ix. *C3j= Blow (Bj, Key3i)*
x. *If (j mod (No-of-Blocks-in-Group) = 0) then*
xi. *i= i+1*
xii. *End If*
xiii. *End For*
xiv. *End*

3.26 MECC-Blowfish Decryption

The secret key Key_i that used for encrypting any block of the message Bi must use to decrypt the same block of the ciphertext $\boldsymbol{Ci}$. Equation (3.12) demonstrated the decryption operation of the ciphertext encrypted by the

proposed MECC-Blowfish technique.

$$B_i = Blow_{Keyi}(C_i) \tag{3.12}$$

3.27 RANDOMNESS ANALYSIS OF MECC-AES AND MECC-BLOWFISH TECHNIQUES

Five experiments are executed to evaluate the randomness of the proposed MECC-Blowfish technique for comparing it with the MECC-AES technique. The Diehard test was used in the experiments to test the randomness of the five data files. All the files that used in the experiments are binary files with size 11 MB. The first file represented the plaintext file. The second file contains a ciphertext encrypted by the standard AES algorithm.

The third file contains a ciphertext encrypted by the MECC-AES technique with 10 secret keys generated using two different ECCs' public key. The fifth file contains a ciphertext encrypted by the Blowfish algorithm. The fifth file contains a ciphertext encrypted by the MECC-Blowfish technique with 10 secret keys generated using two different ECCs' public key.

2.28 REDUCING THE CONSUMING ENERGY OF THE SECURITY TECHNIQUE

One of the WSN restrictions is limited power. In this thesis, a MECC-Blowfish technique has been proposed that is characterized by the strong cryptographic system and it reduces the amount of energy consumed in the encryption and decryption operations. The MECC-Blowfish technique increases the lifetime of the WSN because it uses two security algorithms to consume low energy compared to the others. In order to measure the energy consumption

by the proposed hybrid security technique, NS2 simulator has been used. A code to simulate a WSN with fifteen nodes has been implemented. The NS2 environment simulated building a small WSN with fifteen sensor nodes distributed in a star topology around a coordinator node. Figure (3.7) shows node placement by NS2 simulator of the proposed work.

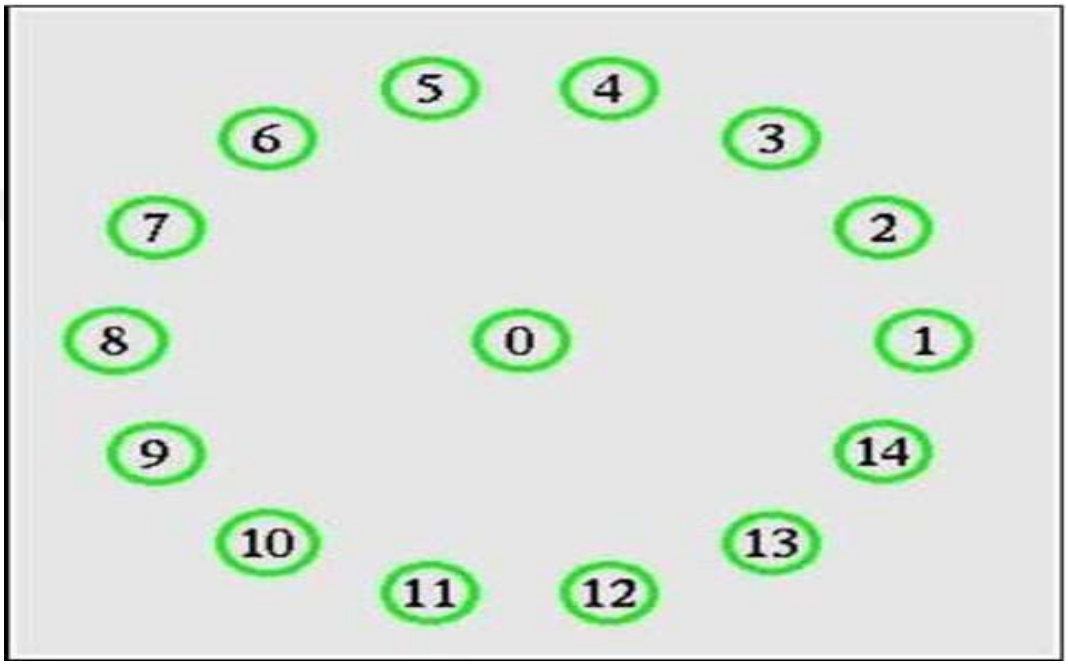

Figure 3.8: Sensor nodes placement by NS2 imulator.

The NS2 environment simulated transferring thirty groups of packets with different data rates. The operation of transferring data packets repeated thirty times for each data rate. The average of thirty transferring operations is calculated for each data rate. The traffic and node parameters are summarized in Table (3.1).

The energy is calculated two times for each technique, one for the transmitted packets and the other for the successfully received packets. Then a comparison is done between the energy consumed by the transmitted packets when ciphering them by MECC-AES technique and the energy consumed after ciphered them by the MECC- Blowfish technique.Then the difference between the energy consumption by the two techniques for each one of the two cases also is calculated. The results of the differences represent the saving energy from using the Blowfish algorithm instead of the AES algorithm by the proposed technique.

Table 3.2: Traffic parameters of simulated WSN using the NS2 simulator.

Parameter	Value
Topology Type	Star Topology
Traffic Type	CBR
Number of Nodes	15
Number of Flows	8
Number of Coordinators	1
Node Movement	None
Node Position	Manual (Along a circle of radius POS)
Traffic Direction	Node -> Coordinator
Packet Size	70 Bytes
Initial Energy	12999.99997 joules
The Radius between the Coordinator and Sensor Nodes	10 meters
The Distance between Nodes	5.347 meters
Area size	10 * 10 meters

3.29 REDUCING THE IMPLEMENTATION TIME OF THE PROPOSED SECURITY TECHNIQUE

In order to improve the system performance of WSN, the MECC-Blowfish technique has been proposed. This technique consists of MECC technique and the Blowfish algorithm. It reduces the implementation time of the cryptographic system. The MECC-Blowfish technique consumes less time when compared with the previous hybrid ECC-AES security technique. In order to measure the time consuming by the proposed hybrid security technique in ciphering data, NS2 simulator had been used. NS2 simulation code has been implemented to simulate a

WSN with fifteen nodes. The NS2 environment simulated transferring thirty groups of packets with different data rates. The operation of transferring data packets repeated thirty times for each data rate and the average of them has been calculated to compare between the time consumed by the transmitted packets when ciphering them by MECC-AES technique and the time consuming when ciphering them by the MECC-Blowfish technique. Then the difference between the times consuming by the two techniques for each one of the two cases also is calculated. The results of differences represent the saving implementation time of using the Blowfish algorithm instead of the AES algorithm by the proposed technique.

4. RESULT AND DISCUSSION

4.1 INTRODUCTION

This chapter recommended a way to check how well and safely similar algorithms work. This let us compare the selected ECC, VC, AES, as well as RC4 schemes, in addition to the new Coordinate Matrix Encryption system, to a certain standards. In this chapter, the findings of the testing framework from the earlier chapters are shown, first one at a time for each algorithm and then as a whole. presents the results for such CME algorithm, then the outcomes for AES, then the outcomes for ECC, therefore the results for VC, then the results for RC4, and then the conclusion and an overview of the results.

4.2 COORDINATE MATRIX ENCRYPTION SCHEME

This section talks about what we found when we looked at the Coordinate Matrix Encryption scheme (CME). It also talks about how the technique is put together when it works of binary strings or byte arrays. describes the findings of the tests formed to determine how well the optimization techniques work. Last but not least, the consequences of something like security assessment for algorithms are given.

4.2.1 SPECIFICS OF THE CME'S BINARY STRING DEPLOYMENT

Java and Php were utilized in order to successfully finish the execution of the CME scheme. In order to accomplish the necessary high-level software implementation, the plan made use of arrays that were two-dimensional (henceforth referred to as "key matrices"). The overall architecture of this implementation has been discussed in the past, and the comprehensive architecture is now available.The first step in the process of using it consisted of the generation of all area taken strings of the desired length. After being enumerated, each string was assigned a one-of-a-kind specific case that fell under both the Coordinate Entry category and the String category. These were a list of all the matrix locations for the that string, a list of each of the matrix locations for that chain, the value of the binary string, and a Boolean that said whether or not the string was empty.

A generator that generates numbers using a method known as pseudorandom sampling assigned several possible positions to each bit string in the key matrix (PRNG). In this particular instance, the algorithm checked the legitimacy of two ostensibly random numbers (x and y) that were employed as coordinates in the matrix and were produced by the PRNG. When a matrix slot became available, the appropriate bit string was written into that slot whenever it was possible to do so. In the event that this was not the case, the PRNG was used to select a new location at random. After the matrix was created, the empty cells were each put into one of the predetermined categories.

After multiplying the number of coordinates assigned to each blank entry by the number of bits assigned to each bit string entry, the final product was a total number of blank entries that was equal to the number of bit strings that could possibly be used. Before the key string was calculated, the required number of empty coordinates was added to each and every one of the blank entries. The one and only binary string that was available contained the first x-coordinate of each bit string.

This binary string was present. The key string provides the closest thing to a PRNG bit key, with a distance of 2:, where n is the number of all possible bit strings. This is due to the fact that the coordinates were generated randomly as part of the key matrix setup, and only the pair of [x,y] coordinates has been required to be unique. If the plaintext wasn't already a binary string, it had to be changed to binary before the CME algorithm could be used to encrypt it. The first thing that was done to encrypt the information was to use an exclusive-OR operation on binary data to combine the binary representations of the plaintext and the key string. An exclusive-OR operation was employed to combine bits from the key string and the plaintext one-to-one.

If the plaintext length was longer than expected, the key string wrapped back around to the beginning. The final string was found by decrypting the whole encrypted string one n-bit chunk at a time. The next part of the ciphertext, which could be either an encrypted part of the message or a blank padding organize, was chosen by a pseudo-random number generator (PRNG) that flipped a coin.

If the result of the coin flip meant that a padding subject had to be added, a second PRNG was used to pick, at random, from a list of blank entries, a coordinate couple location that had no coordinates or subjects at all. If the flip of a coin showed that the next bit string of the message needed to be encrypted, the PRNG would pick a random offset from the coordinates table and use that as the starting point for the encryption.

The algorithm then added the representation of the two [x, y] coordinate values that had been chosen at random to the string that held the ciphertext. This message is encrypted so that each character has the same number of padding coordinates.

An entry lookup is utilized in the decryption process of the CME scheme. The algorithm examines each coordinate in the ciphertext to determine whether or not it corresponds to the value found in the key matrix.

If there is no value in the coordinates section of the ciphertext, that part of the ciphertext is ignored. The coordinates and the plaintext both receive a bit string value that is appended to them.

After each coordinate has been verified twice, Using exclusive-OR, the key string is added to the string that was found. After a lookup is finished, the key string is used to get back to the plaintext that was originally stored. Because encryption depends on lookups, the CME integration could theoretically take longer to decrypt than to encrypt.

4.2.2 SPECIFICS OF THE BYTE ARRAYS-BASED CME IMPLEMENTATION

Since the CME framework is a high-level application that operates with binary strings, its overall efficiency will be much lower than that of all the embedded Java characteristics for encryption systems like AES, which work well with arrays of bytes. This is because the CME framework was already described as a shows a high level that works with binary strings. By making a version of CME with a fixed 8-bit scheme size, it was possible to get a good idea of how well the different schemes work.

This version of CME works with arrays of bytes instead of binary strings. This was done to save time. The CME scheme can then be incorporated at a lower level, and its performance can be made to be similar to that of the native Java functions for AES encryption and ECC key generation. The plan for the CME Byte Array works in a way that is similar to what was said in the last section.

A changed version of the Coordinate Entry class stores each bit string in the same way that its byte value is stored in a matrix that has all of the possible binary strings. Since each UTF-8 character is converted to a byte value that falls within the range [-128, 127], there is no need to pad the data before encryption. After that, the ciphertext is a string of integers in the range [0, 255], with each pair of integers corresponding to the x and y coordinates of an entry in the key matrix. You can find these coordinates in the step before this one. In contrast to the high-level implementation, the exclusive-OR operation on the plaintext is done in place, byte by byte, during the coordinate encryption process. This is done to move things along faster.

As the array goes through its iterations, the encryption algorithm gets either the two integer coordinates for such target byte value or the coordinates for a randomly chosen empty padding location. The data is then encrypted using this information. This list of numbers is used to send the ciphertext.

During the decryption process, each pair of x and y values in the integer array that makes up the ciphertext is checked to see if they both point to the same place. This step turns the encryption process around. Any bytes that are already at that position are exclusive-ORed with the current integer value of the key string. The bytes that result from this operation are added to the array of plaintext bytes. After the coordinates of the plaintext byte array have been checked, the array is converted back into the UTF-8 characters that are encoded for each byte value and then shown.

4.2.3 EFFICIENCY

We used a wide variety of research methods, as described in the research plan that was shown in Research work, to figure out how well the CME program worked. It was found out how long it takes to make a key, how long it takes to both encrypt and decrypt data, and the amount of memory the implementation needs to run. When we worked with bytes, we had to encrypt and decrypt each bit of data a thousand times. When we worked with bit strings, we only had to do this 500 times. We went through each of these iterations and wrote down how long it took on average to run and how much memory it used. We then took the average of these numbers for each of the different dataset sizes. In Appendix C, you can find the data from the tests that were used. The byte implementation of CME could be tested by taking the UTF-8 encoded plaintext of Hamlet and Pride and Prejudice. the time needed for the byte implementation of CME increased proportionally with the size of the data, which is what you would expect from a stream cipher that practices data bytes one at a time. Table 4.1 gives an overview of how long it takes on average to encrypt and decrypt data sets of different sizes.

Table 4.1: Mean duration for byte CME encryption and decryption.

Data Size	Average Encryption (ms)	Average Decryption (ms)
304	0.031	0.012
928	0.059	0.023
3024	0.136	0.065
4408	0.173	0.05
8144	0.262	0.093

All of the operations took longer to finish because they needed to work together on a technical level using bit strings. This was because string variables were used, and during the implementation process, different permutations and substitutions were made to them, which made the total time needed longer.

Table 4.2 shows the average amount of time it takes, which was found by doing 500 repetitions.

Table 4.2: Normalized 4-bit string CME encrypt/decrypt times.

Data Size (bits)	Average Encryption (ms)	Average Decryption (ms)
16	0.02	0.026
32	0.066	0.036
64	0.104	0.06
128	0.214	0.074
256	0.396	0.136
512	1.13	0.328

We were able to test how much memory the byte implementation of CME used by measuring how much memory the Java Virtual Machine environment used when the scheme and key matrix were initialized, encryption, and decryption were done. This was done by checking how much memory was being used in the Java Virtual Machine setting as a whole. After setting up and making keys a hundred times for testing purposes, the amount of memory needed, measured in megabytes, as well as the amount of time needed, measured in milliseconds, were written down. After that, the average of these numbers was found by doing some math. Table 4.3 shows what was found when the configuration of the scheme was looked into.

Table 4.3: The arithmetic mean of byte CME's configuration time and memory.

Memory used (MB):	1.217
Time taken (ms):	80.13

For every one of the different lengths of data string, the amount of memory needed for encrypting and decrypting with a single key in the byte model of CME was measured. This was done over a thousand times with a single key. In Table 4.4, the results of these experiments are listed by their averages. In a similar way, the time it takes to decrypt a file grows directly with the size of the file.

Table 4.4: The arithmetic mean of byte CME encryption and decryption memory.

Data Size	Average Encryption (MB)	Average Decryption (MB)
304	1.244	1.244
928	1.245	1.246
3024	1.251	1.251
4408	1.255	1.255
8144	1.263	1.264

The same recollection testing was done on the bit-string deployment of CME. Over 100 iterations, the average time it took for JVM to start up and how much memory it used were recorded. Table 4.5 shows how things turned out.

Table 4.5 : Cost of setup and average amount of memory used by CMEs using 4 bit strings.

Memory used (MB):	0.448
Time taken (ms):	21.44

The bit string implementation's four-bit arrangement size and 16*16 key matrix made it easier to make and store the key. This is in contrast to the byte implementation, which used an eight-bit scheme and a 256*256 key matrix. For the byte implementation, the byte implementation was used. For each of a pseudo-random bit string's four data sizes, the amount of memory needed to encrypt and decrypt a 4-bit string has been figured out.

To do this, a single key was used to encrypt and decrypt more than 500 different versions. After getting these numbers, the means for each of the datasets were calculated. In the table at 4.6, you can see the results.

Table 4.6: The median amount of RAM needed for 4-bit string CME encryption and decryption.

Data Size (bits)	Average Encryption (MB)	Average Decryption (MB)
16	0.475	0.475
32	0.476	0.478
64	0.476	0.476
128	0.478	0.478
256	0.480	0.480
512	0.484	0.484

4.2.4 SECURITY

The CME scheme is secure and efficient in warding off brute-force attacks for deployments that contain at least a predetermined minimum number of bits. This is because the matrix appears to be quite large, and each position contains an extremely diverse set of values. It is necessary for the attacker to try approximately half of all possible keys for a brute-force attack to be successful. this is the number of matrices that would need to be tried by an adversary if they attempted to brute force a 4-bit CME scheme over and over again. The end result is significantly greater than the largest value that can be represented by a single floating-point integer, which means that it cannot be calculated using a standard calculator. Because of this, the calculation that needed to be done in MatLab had to make use of the variable point integer toolbox.

There are 257 different keys that could be used for a matrix if the scheme used is byte-oriented and 8 bits long. Because of this, it is now possible to use an implementation strategy that is more flexible.

It is speculated that breaking an 8-bit scheme would require a significant number of attempts using brute force.If a bad guy had access to all of a scheme's matrix keys, it's likely that they would find more than one key matrix that led to a plaintext that they could figure out. Because the exclusive-OR action uses the first generated x position for each bit-string/byte, an attacker would have to try all of the possible key strings for every key matrix, which would be a lot of work. This is due to the exclusive-OR operation uses the first x position that is made. So, if someone tried to break in, they would have to figure out which of the keys was the real one.By putting each bit string in more than one place in the matrix and doing an exclusive-OR operation well before coordinate encryption, the algorithm spreads out any statistical properties of the plaintext.

The matrix is set up so that half of the cells only have empty space in them, while the other half have bit strings. During the procedure of encrypting plaintext, a Pseudo-Random Number Generator (PRNG) is used in a way similar to flipping a coin to decide whether the algorithm will put an empty space or any of the spaces for the next bit string to encrypt. This means that the likelihood of any given bit string is about the same as the likelihood of any given blank padding location.

This technique is used to count the number of positions in a certain sequence. the CME scheme is immune to frequency analysis because of the exclusive-OR operation that is performed before encryption and the numerous places that are assigned to every bit string in the scheme. The list of empty entries, which is made in the same way as the ranking of occupied coordinates, makes it even harder to understand the statistical data that are based on the data.. Frequency analysis software was developed specifically for this scheme; it took the input ciphertext and key matrix and determined how frequently each blank and full coordinate appeared in the ciphertext.

This software also took the input key matrix into consideration. The software that was used to perform the frequency analysis then printed out a series of numbers that indicated what percentage of single occurrences had no coordinates, what percentage had full coordinates, what percentage had both, what percentage had neither, and so on.

The program utilized the CME byte implementation to produce 1000 ciphertexts from a single key by using plaintext strings of varying lengths. These ciphertexts were generated from the single key. Next, we tabulated the results of the average frequency analysis for each of these ciphertexts, and then we mapped those results. the probability that one-time coordinates would either be empty or full was the same as that of secondary occurrences.

When we compared using full coordinates to using blank coordinates, we found that there was only a ten percent difference between the third and fourth instances. The English language string with 8816 bits was the longest one used for testing. The information on the average frequency of occurrences for that string is presented in Table 4.7, along with the total number of times that frequency was sought from a ciphertext over the course of a thousand separate tests. Since the frequencies are a statistical representation of encrypted data, they do not give much away about the plaintext and are fairly consistent overall.

This is so because the frequencies themselves represent the secret information. In addition, the distribution became more even and flattened out as the number of occurrences of a given frequency increased. Because of this, it's highly likely that we need to carry out even more experiments, which would result in distributions that are even more uniform.

Table 4.7: Deciphering ciphertext from such an 8816-bit string using frequency analysis.

Frequency	Blank	Full	Times Occurred
1	50.047%	49.953%	1000
2	48.589%	51.411%	1000
3	46.491%	53.508%	308
4	40%	60%	5

Since With the addition of empty neoprene coordinates, the scheme is safeguarded from known and selected plaintext attacks, as the plaintext attacker will not be able to determine which ciphertext coordinates are empty padding and which are places for plaintext strings.

The reason for this is that it is impossible to determine which ciphertext coordinates are just for filler and which are intended to hold actual plaintext strings. For added security against the statistical method, an exclusive-OR operation is performed on each bit chain and byte. A foe with infinite computational resources would still need to make educated guesses to determine whether or not each coordinate was null. When a 4048-bit string of a single repeated character was used for a chosen plaintext attack, the CME byte scheme produced a distribution that was statistically indistinguishable from the null distribution.

After encrypting a thousand different texts with the same key, only 46 of them contained coordinates that appeared more than twice. The typical outcomes of analyzing the selected plaintext attack are summarized in Table 4.8. A potential key string or matrix can be learned very little from a selected plaintext attack due to the small number of repeated coordinates and the occurrence of blank coordinates within the repetitions. This occurs because some of the repeated coordinates are blank. Since the key string is generated at random, it would be impossible to learn how often any particular character appears within it.

Table 4.8: Examining the ciphertext's frequency using the plaintext's 4048 bits.

Frequency	Blank	Full	Times Occurred
1	50.105%	49.895%	1000
2	44.127%	55.873%	1000
3	31.915%	68.085%	46

In Table 4.9, you can see the results of encrypting a 4408-bit English plaintext string using a single string of repeated characters. It is possible to contrast these results with those obtained by deciphering the text. Slight variations in distributions can be attributed to the different key strings and key matrices, but on the whole, the distributions are very similar.

Table 4.9: Analysing the frequency of 4408-bit ciphertext.

Frequency	Blank	Full	Times Occurred
1	50.015%	49.985%	1000
2	49.109%	50.891%	1000
3	53.704%	46.296%	54

The avalanche effect generated by the algorithm was the system's final defense. The CME scheme is great for creating an avalanche effect because it generates new ciphertext almost each time the same plaintext is scrambled by the same key, with less than 0.5% of bytes remaining constant from the ciphertext that came before it.

When using bytes, the avalanche effect was tracked by counting how often the same byte appeared and how often it occurred in the same position. These metrics were measured over the course of a thousand encryption and decryption cycles on data that varied by a single bit, as well as over the course of a thousand cycles on the same piece of data, for the purpose of making a comparison. In Table 4.10, we can see the median results of these tests for both the modified and the unmodified datasets.

Table 4.10: Effect of an avalanche in byte CME.

Data Size	Unchanged from Previous		1-bit Altered from Previous	
	Same Bytes	Same Position	Same Bytes	Same Position
304	44.653%	0.441%	44.839%	0.414%
928	84.064%	0.419%	84.026%	0.388%
3024	99.722%	0.390%	99.713%	0.422%
4408	99.973%	0.400%	99.984%	0.404%
8144	100%	0.396%	100%	0.395%

4.3 ENCRYPTION STANDARD FOR THE FUTURE

The next section will focus on the AES low-level software solution and the testing results in terms of efficiency and security. in addition to providing a detailed evaluation of the technique employed in practice. and then analyzes the AES's functionality in light of the test results. The paper concludes with a brief summary of the security offered by AES, drawing on both the algorithm's theoretical foundations and the empirical data derived from the analysis of tests.

4.3.1 SPECIFICS OF THE ROLLOUT

The javax.crypto as well as java.security bundles in Java SE 7 contain the APIs required to implement AES encryption. The Initialization Vector and the random 128-bit key were both produced by the system's integrated Secure Random function. Using the AES Encryption parameters, CBC mode, and PKCS5 Padding, the Java Cipher function encrypted the plaintext string used by the algorithm.

The function took as input a byte array containing the ciphertext and returned the same array. The message was decrypted using the same Cipher parameters that had been used to encrypt it, and the resulting plaintext byte array was decoded to recover the original UTF-8 string. research indicates that ECB mode is the least comfortable mode of operation for AES implementations because it can cause repeated blocks of ciphertext. Therefore, CBC was chosen as the encryption mode because it offers superior security against statistical analysis techniques.

Although ECB mode is the safest option for AES implementations, it was not selected (Thakur & Kumar, 2011). By combining each ciphertext block with the following plaintext block in sequence, CBC mode eliminates the possibility of repeated blocks. To prevent the same blocks from appearing multiple times, this is done. Complete scripts for both AES ciphering and decryption.

4.3.2 EFFICIENCY

Time taken to initialize the scheme, generate the key, encrypt the plaintext, and decrypt the ciphertext all factored into an overall evaluation of AES's efficiency. Also monitored was the JVM's memory consumption at various points in time. From there, we have been able to zero in on the center. This was done with every possible data set size. the time required to encrypt and decrypt five different data sizes was measured while encrypting as well as decrypting a total of the one thousand times. The length of time needed to complete each cycle was recorded. The data compilation is summarized in Table 4.11.

Table 4.11: A statistical analysis of 128-bit advanced encryption standard (AES) encryption and decryption.

Data Size	Average Encryption (ms)	Average Decryption (ms)
304	0.199	0.17
928	0.142	0.182
3024	0.173	0.179
4408	0.185	0.196
8144	0.148	0.25

The time and memory required to set up the implementation have been measured. The framework setup procedure was repeated a hundred times, and data on the amount of RAM needed and the time it took to complete were recorded after each run. The data is summarized in Table 4.12 for your convenience.

Table 4.12: An average of the time and space needed for 128-bit AES setup.

Memory used (MB):	2.364
Time taken (ms):	409

4.4 QUANTUM CRYPTOGRAPHY

Encrypts data using the unbreakable properties of quantum theory and then transmits the encrypted data over a network. The purpose of cryptography is to guarantee that no one other than the recipient of a message that has been encrypted is aware of the decryption key. Quantum cryptography, on the other hand, employs a security model that is founded on physics rather than mathematics, in contrast to more traditional cryptographic systems.

With quantum cryptography, there is no way for the sender or the recipient of a message to have their private information compromised in any way. It is impossible to view or copy the information that is encrypted within a quantum particle without revealing its presence to either the receiver or the sender of the message. Quantum cryptography must maintain its level of safety even in the presence of quantum computers for it to be useful. Information can be encrypted using quantum cryptography by directing photons, which are individual particles of light, down a fiber optic cable. In a binary system, these photons serve the same function as "bits." The application of quantum mechanics is required to ensure the integrity of the system. The following are some examples of these safety features:

i. Particles can be in multiple locations or states simultaneously.

ii. It is impossible to observe a quantum property without altering or disturbing it.

iii. We are unable to duplicate particles in their entirety.

When attempting to measure the quantum state of a system, it is always possible to have an adverse impact on the system that is being investigated due to the characteristics described above. The photon is the particle of choice for quantum cryptography because it possesses all of the necessary qualities, including the following: Because of their role as information carriers in fiber optic cables, their type of behaviour has been extensively researched. Quantum key distribution (QKD) is an established implementation of quantum cryptography that ensures a secure method of key exchange. QKD is also known as quantum key distribution.

4.5 METHODOLOGICALLY, HOW WILL QUANTUM CRYPTOGRAPHY FUNCTION

In 1984, a working model for quantum cryptography was developed, and the technology ultimately derived from that model. A hypothetical conversation between two individuals, Alice and Bob, who are interested in exchanging messages in an encrypted fashion serves as the foundation for the model. Alice kicks off the conversation by delivering Bob a key to his door. The essential component is a flow of photons in only one direction.

Each photon can either be a 0 or a 1 when functioning as a unit of information. Not only are these photons moving in a straight line, but they are also vibrating in a pattern that is very specific.because of this, Alice makes sure that the photons she uses to send the message go through a polarizer first. A polarizer is a specific kind of filter that allows only particular kinds of vibrating photons to pass through, while permitting other kinds of photons to do so in a modified form. There were four possible orientations for the polarization: vertical (1), horizontal (0), 45 degrees to the right (1), or horizontal to the left (1). (1). (0 bit). In her schemes, either one of the two polarizations that were used in the transmission could serve as a substitute for a single bit.when they have passed through the polarizer, the photons are sent along the optical fiber to Bob, who is the receiver.

For the purpose of determining the polarization state of each individual photon, a beam splitter is utilized. When Bob receives the photon key, any one is chosen at random as the polarization of photons to use because he does not know which polarization of photons to use. Alice will now reveal to Bob which polarizer she used to send each photon after the two of them have compared the various methods. After that, Bob makes sure that he is using the appropriate polarizer by checking it out. The remainder of the sequence is what is used as the key after the photons that were read with the incorrect splitter have been thrown out.consider the possibility that Eve is listening in on this conversation from a nearby location.

Eavesdropping is attempted by Eve, who employs the same methods as Bob and uses the same equipment. Bob, in contrast to Eve, is able to confirm the type of polarizer that was applied to each photon by discussing the matter with Alice. As it happens, Eve arrives at the wrong conclusion regarding the last key. Bob and Alice would also be aware of the situation if Eve was eavesdropping on them. If Eve were to observe the flow of photons, a shift would occur in the positions of the photons that Bob and Alice anticipated seeing, as shown in Figure 4.1. This shift would be visible to both Bob and Alice.

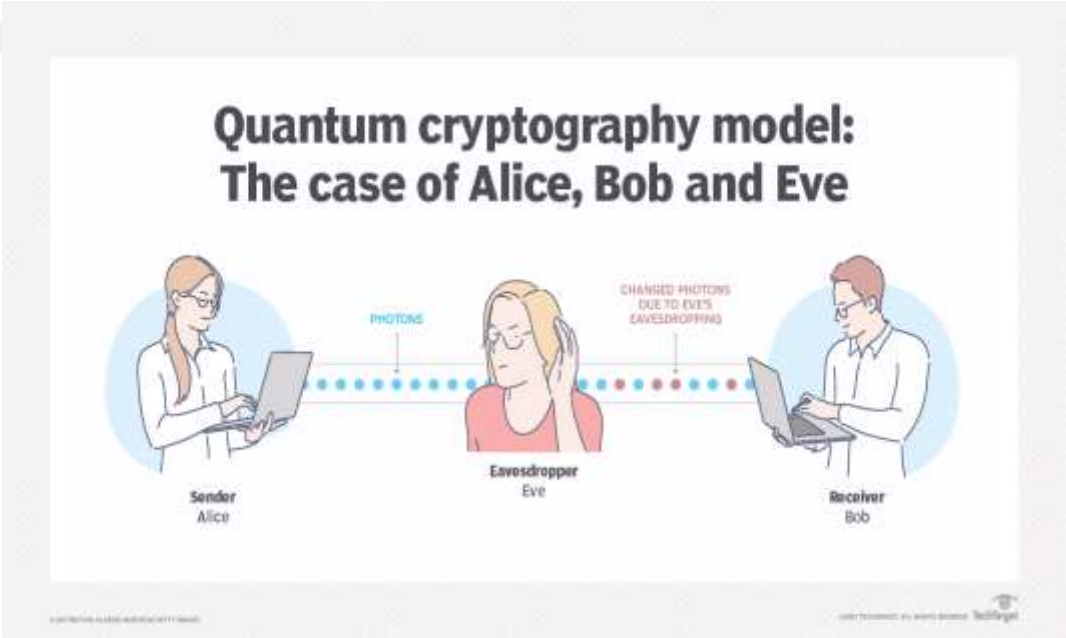

Figure 4.1: Quantum cryptography mpdel.

4.6 APPLICATIONS AND BENEFITS OF QUANTUM CRYPTOGRAPHY

When compared to classical cryptography, quantum cryptography affords users enhanced confidentiality in their exchanges. There is little cause for concern that a bad attacker could decode the information without the key after the involved parties have exchanged keys. Both the sender as well as the receiver can be warned if the key is tampered with while it is being constructed. While further research is needed, there have been functional implementations of the following cryptographic technique:

i. To use the BB84 quantum cryptography protocol, researchers at Cambridge University and engineers at Toshiba Corp. developed a high-bit-rate QKD system.

ii. After years of development by researchers at Boston University, Harvard University, and IBM Research, the Defense Advanced Research Projects Agency Quantum Network operated from 2002 until 2007.

iii. The first U.S.-based quantum network was unveiled by Quantum Xchange, and it included a fiber optic cable length of one thousand kilometers (km).

iv. Commercial QKD systems were also developed by companies like ID Quantique, Toshiba, Quintessence Labs, and MagiQ Technologies Inc.

Other than QKD, some of the most well-known protocols as well as quantum algorithms in quantum cryptography are as specified. Quantum cryptography may take many forms, including quantum ciphers based on position or devices, and even quantum coin flipping.

4.6.1 BENEFITS OF QUANTUM CRYPTOGRAPHY

The following are some of the advantages of quantum cryptography:

i. Allows for risk-free conversation: Quantum cryptography is a more advanced and safe method of encryption that uses the laws of physics rather than mathematically challenging numbers.

ii. Finds listening devices: The quantum state shifts and the intended outcomes for the users is altered if a service provider attempts to decode the information.

iii. provides a variety of safeguards: Multiple quantum cryptographic protocols are currently in use. Some, such as QKD, for instance, can supplement traditional encryption techniques.

4.6.2 CONSTRAINTS ON QUANTUM CRYPTOGRAPHY

Some potential drawbacks and restrictions of quantum cryptography include:

i. Error rates and polarization changes: Polarization of photons may shift during transmission.

ii. Exceptionally in the case of Terra Quantum, which will be discussed further down, the maximum distance of quantum cryptography had also generally been between 400 and 500 kilometers.

iii. It's costly because quantum cryptography typically needs its own infrastructure, like fiber-optic lines as well as repeaters.

iv. It is impossible to send keys through a quantum channel to more than one destination.

4.7 QUANTUM CRYPTOGRAPHY AND ITS DIFFERENCES FROM CLASSICAL CRYPTOGRAPHY

Information is mathematically encoded in traditional cryptography in such a way that it can be decoded by only one person who possesses the correct decryption key. This ensures that only one person can read the information. Symmetric and asymmetric key distribution are the two primary methods utilized in conventional cryptography for the purpose of key distribution.

A public key is used to encrypt messages, while a private key is used to decode them when utilizing asymmetric cryptography. Symmetric key algorithms, on the other hand, only require a single key to function. Because it is impossible for traditional computers to factor the large numbers used in public and private keys in a timely manner, traditional cryptography makes use of these numbers. Figure 4.2 shows an example of traditional cryptography.

The mathematical foundations of classical cryptography are not used in quantum cryptography because quantum cryptography is based on the principles of quantum mechanics rather than mathematics. In addition, while traditional cryptography is based on mathematical computation, quantum cryptography is significantly more difficult to decipher.

This is due to the fact that merely observing the involved photons changes the expected outcome, which alerts both the sender and the receiver to the presence of an eavesdropper. Classical cryptography uses a key and a public key. Because it is necessary to use fiber optic cables and strategically placed repeaters to amplify the signal, quantum cryptography is also commonly associated with a distance or range. This is because of the necessity of these two things.

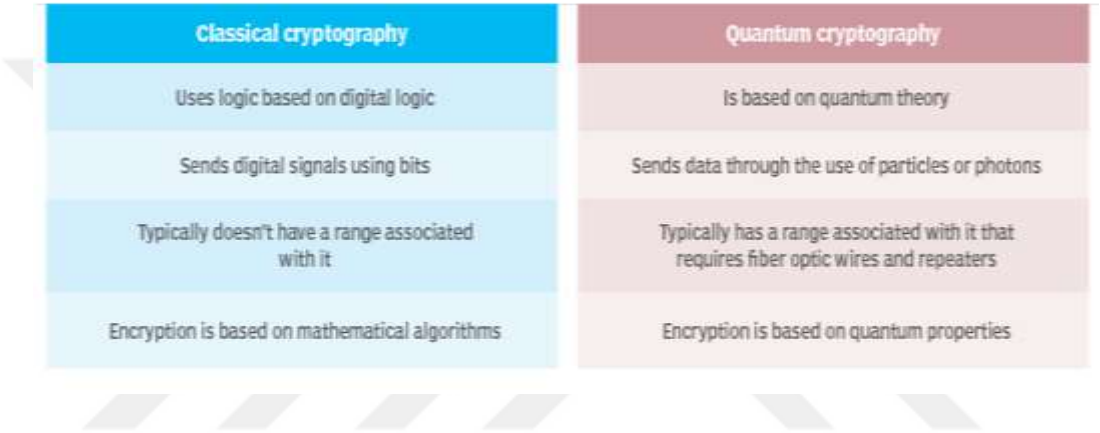

Figure 4.2: Quantum cryptography and classical cryptography.

4.8 QUANTUM CRYPTOGRAPHY: THE WAY FORWARD

Due to the fact that quantum computers are still in their early stages of development, there is still a significant distance to travel before the general public will be able to use quantum communications. Even though it has some drawbacks, such as the incapability to send keys to even more than one location at the same time, quantum cryptography is continuing to grow in popularity.for example, there has recently been an expansion in the variety of modern devices.

A Swiss company specializing in quantum technology called Terra Quantum just recently made an important announcement regarding an increase in the availability of quantum cryptography. Quantum cryptography could only be utilized locally, within a radius of between 400 and 500 kilometers, until very recently. The development of Terra Quantum has made it possible to send the keys to quantum cryptography over a distance of more than 40,000 kilometers.

The innovation developed by Terra Quantum makes it possible to distribute quantum keys within already-established, widely-utilized optical fiber lines in telecommunications networks. This removes the necessity of constructing a new optical line that contains a large number of repeaters.

4.9 AS AN ILLUSTRATION OF QUANTUM ENCRYPTION, CONSIDER THE FOLLOWING

Let's suppose that two individuals, Alice and Bob, want to have a one-on-one conversation with each other but want to keep it private. Alice sends Bob a stream of polarized photons via a fiber optic cable in the game QKD. Bob receives the photons. There is no point in securing this cable because the quantum states of the photons can never be predicted with any degree of accuracy. In order to discover the secret, Eve's eavesdropping would require her to decipher each individual photon. Therefore, it is necessary for her to give Bob that photon. When Eve reads the photon, as shown in Figure 4.3, she introduces information that is incorrect regarding the quantum state of the photon.

When something like this occurs, Alice and Bob are aware that the key has been compromised and that it can be safely discarded. Before Alice can let Bob in on the secret, she needs to make sure he has a fresh key that hasn't been compromised.

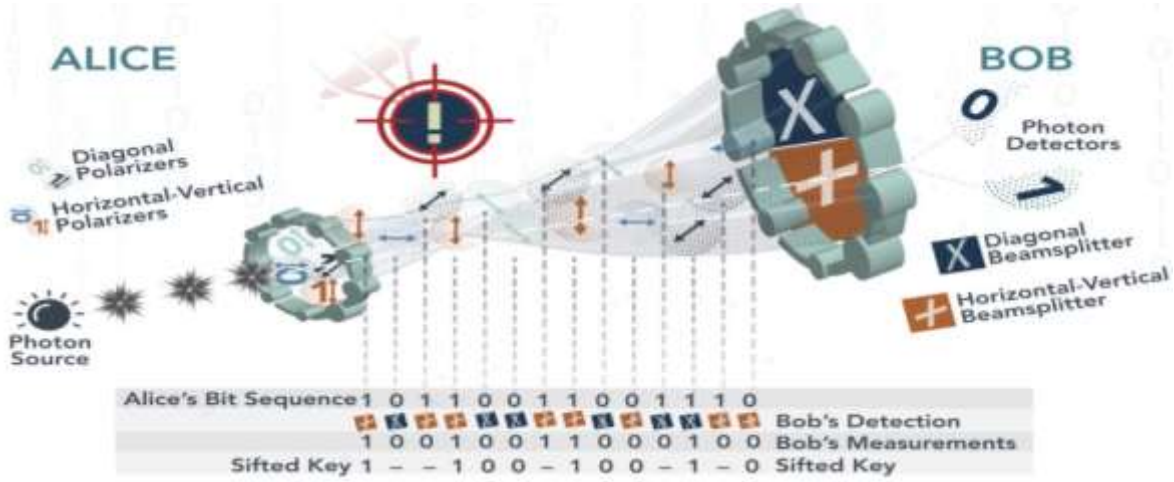

Figure 4.3: Examples of quantum cryptography.

4.10 DISCUSSIONS AND HYPOTHESES PROPOSED STUDY

Here we present the hypotheses and research questions that were generated by the test results.

4.10.1 TO MOST OF WHAT EXTENT DO GRAPHIC-BASED SYSTEMS IMPROVE SECURITY OVER MORE TRADITIONAL CRYPTOGRAPHIC ALGORITHMS?

When compared to implementations of the same size, alternative key structures that are used in graphic-based systems can provide a greater level of security. The Coordinate Matrix Encryption (CME) scheme that was proposed did not present any security concerns. The theoretical resistance of the CME scheme to brute-force attacks is significantly higher in comparison to the 128-bit AES implementation; specifically, it is 157,899 orders of magnitude higher in terms of the average number of key attempts that are required. After a variation of one bit in the plaintext, less than one percent of the bytes in the ciphertext are still in the same position.

This demonstrates that the avalanche effect of the CME scheme is faster than that of the implemented AES. In contrast, using AES ensured that between 25 and 49% of the plaintext's ciphertext bytes remained in the same position in the ciphertext even after a single bit shift in the plaintext. The frequency distributions over the ciphertext that were produced by both algorithms conveyed very little information about the plaintext that was being encrypted (Tables 4.13 to 4.14). Both algorithms are improved by the use of a pseudo-random key, which helps to disperse the statistical properties of the plaintext and adds to the confusion that is caused in the ciphertext as a result.

After applying AES 128 encryption to a string containing 8814 plaintext bits, the resulting ciphertext contained the same byte repeated 16 times. When the same string was encrypted using an 8-bit CME scheme, Table 4.15 reveals that the frequency of 3 was the one that occurred most frequently.

Table 4.13: A regular expression of 128-bit AES-encrypted ciphertext from an 8814-bit string.

Frequency	Average # of Bytes	Times Occurred
1	18.626	1000
2	37.606	1000
3	50.462	1000
4	50.01	1000
5	39.938	1000
6	26.581	1000
7	15.194	1000
8	7.554	1000
9	3.403	978
10	1.780	749
11	1.271	388
12	1.068	146
13	1.019	54
14	1	18
15	1	4
16	1	1

Table 4.14: Frequencies of cipher authentication codes for messages were looked for in the ciphertext of a string with 8814 bits (CMEs).

Frequency	Blank	Full	Times Occurred
1	50.047%	49.953%	1000
2	48.589%	51.411%	1000
3	46.491%	53.509%	308
4	40%	60%	5

With an 8-bit CME being 157,899 times more expensive to break than a 128-bit RC4 using a brute force attack, CME once more offered a measurably higher level of security than the stream cipher RC4. This is as a result of the larger key size of an 8-bit CME. Table 4.16 shows that the avalanche effect of the CME cipher was significantly larger than that of the RC4 cipher, resulting in only very small differences between the ciphertexts. Because RC4's frequency distribution and that of AES were so similar, the 8-bit CME's frequency distribution was flatter and contained fewer high frequency occurrences.

When using CME, only three occurrences in an 8814-bit string's ciphertext were found; however, when using RC4, this number rose to sixteen (Table 4.14). These comparisons show that the graphic-based alternate cipher provides a greater level of security than the RC4 stream cipher or the Feistel-based AES. In contrast to a standard binary key string, which only allows for two possible values for a given bit, a key with this alternative structure and using bit string code words as an alphabet produces a significantly larger set of possible keys.

4.10.2 IDENTICALLY WHAT CHALLENGES COME UP WHEN PUTTING IN PLACE GRAPHICAL USER INTERFACES?

However, the implementation of these alternative graphical-based systems involves computational overheads in the creation of the key structure. This is because the security

provided by these systems is dependent on complex key structures. When encrypting and decrypting data with traditional key structures, such as bit keys, there is not always the need for this additional computational overhead.

The CME method that has been proposed makes use of a large two-dimensional key matrix, and the majority of the computational overhead that the algorithm generates is concentrated there. The computational complexity of the algorithm is relatively high due to the fact that the design of the key plays such an important part in ensuring the algorithm's integrity. The key matrix that was produced by the 4-bit scheme is roughly equivalent to a 256-bit pseudorandom binary key as a result of the fact that each full coordinate in the matrix contains a 4-bit codeword.

As a direct consequence of this, the key matrix has a prerequisite level of memory that must be satisfied. This finding is consistent with ciphers that are based on large-girth special graphs, such as the Cayley graphs that were used by Ustimenko (2007). These ciphers can experience significant computational overheads due to the size of the graph, and this finding supports that hypothesis. In contrast to binary keys, for which the key size can be easily determined, such as in AES, the key structure of the CME design is unconventional, which requires the algorithm to come up with creative ways to store it. Since the key size can be easily determined for binary keys, this is not a problem for AES.

4.11 THE PROPOSAL PROJECT

The proposal project used for Cryptography Ciphers more than 30 Methods to get the best way to encrypt and decrypt any Statement and send it to the other person how use the proposal project and this site can use in any place and any OS, Figure 4.4, 4.5, 4.6.

Figure 4.4: Proposal project – part 1.

Figure 4.5: Proposal project – part 2.

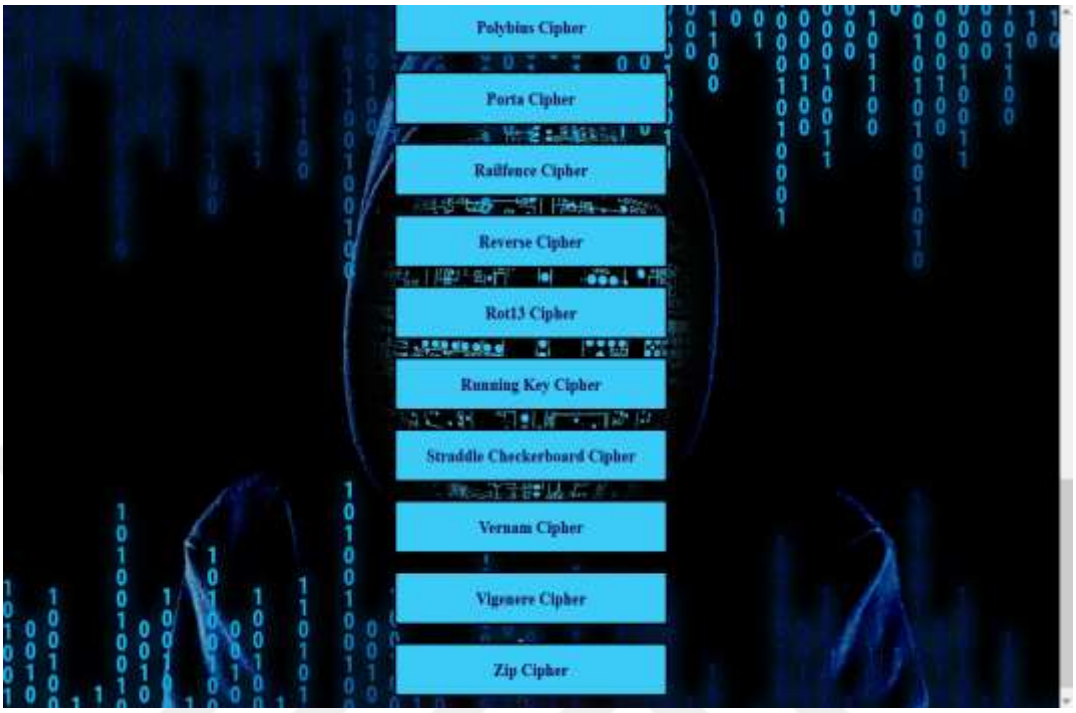

Figure 4.6: Proposal project – part 3.

4.12 ADFGVX CIPHER

In cryptography the ADFGXV cipher can define as the following and shown in Figure 4.7 below.

i. The ADFGVX cipher was a field cipher used by the German Army during World wari. It is an extension of an earlier cipher called the ADFGX cipher Invented by Colonel Fritz Nebel and introduced in March 1918.

ii. This cipher was a fractionating transposition cipher which combined a modified Polybius square with a single columnar transposition.

iii. The cipher is named after the six possible letters used in the ciphertext: A, D, F, G, V and X.

iv. The 'key' for a ADFGVX cipher is a 'key square' and a key word.

v. The key square is a 6 by 6 square containing all the letters and the numbers 0 - 9. The key word can be any word.

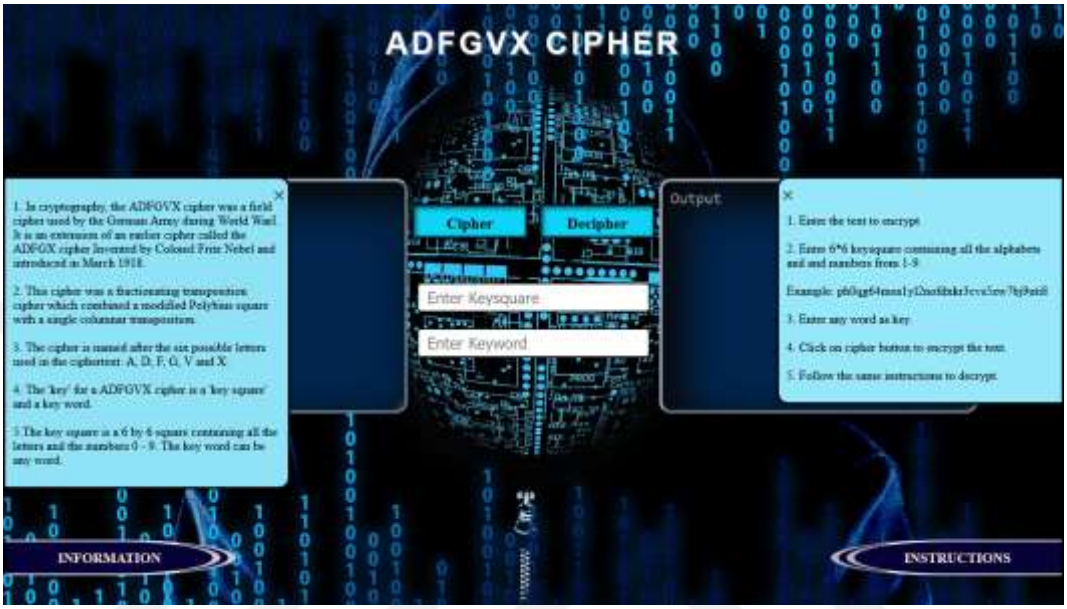

Figure 4.7: ADFGVX cipher.

4.13 ADFGX CIPHER

i. In cryptography the ADFGV cipher can define as the following and shown in Figure 4.8 below.

ii. The ADFGX cipher was a field cipher used by the German Army during World War I. It is closely related to ADFGVX cipher.

iii. ADFGX is a fractionating transposition cipher which combined a modified Polybius square with a single columnar transposition.

iv. The cipher is named after the five possible letters used in the ciphertext: A, D, F, G and X.

v. The 'key' for a ADFGVX cipher is a 'key square' and a key word.

vi. The key square is a 5 by 5 square containing all the letters except 'j'. The key word can be any word.

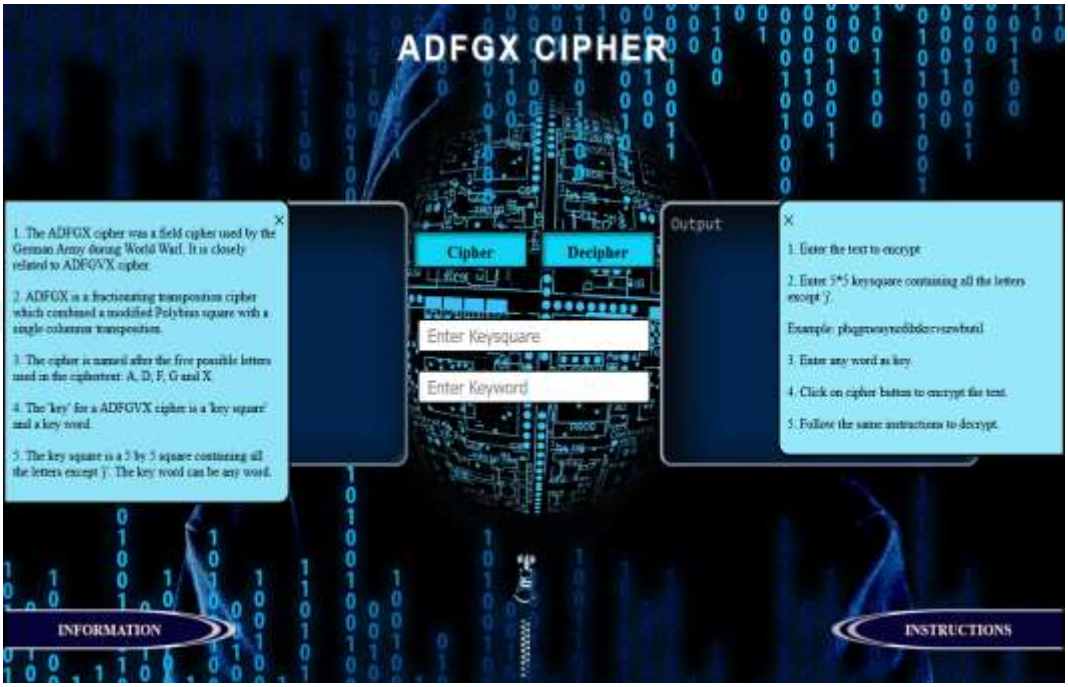

Figure 4.8: ADFGX cipher.

4.14 ASCII CIPHER

In cryptography the ASCII cipher can define as the following and shown in Figure 4.9 below.

i. The ASCII cipher is a substitution cipher.

ii. ASCII is a computer code that is like binary. Instead of using 1's and 0's like binary it uses the numbers from 1-256.

iii. In the encrypted text a is replaced with 97, b with 98, and so on. and in similar manner A is replaced with 65, B with 66 and so on.

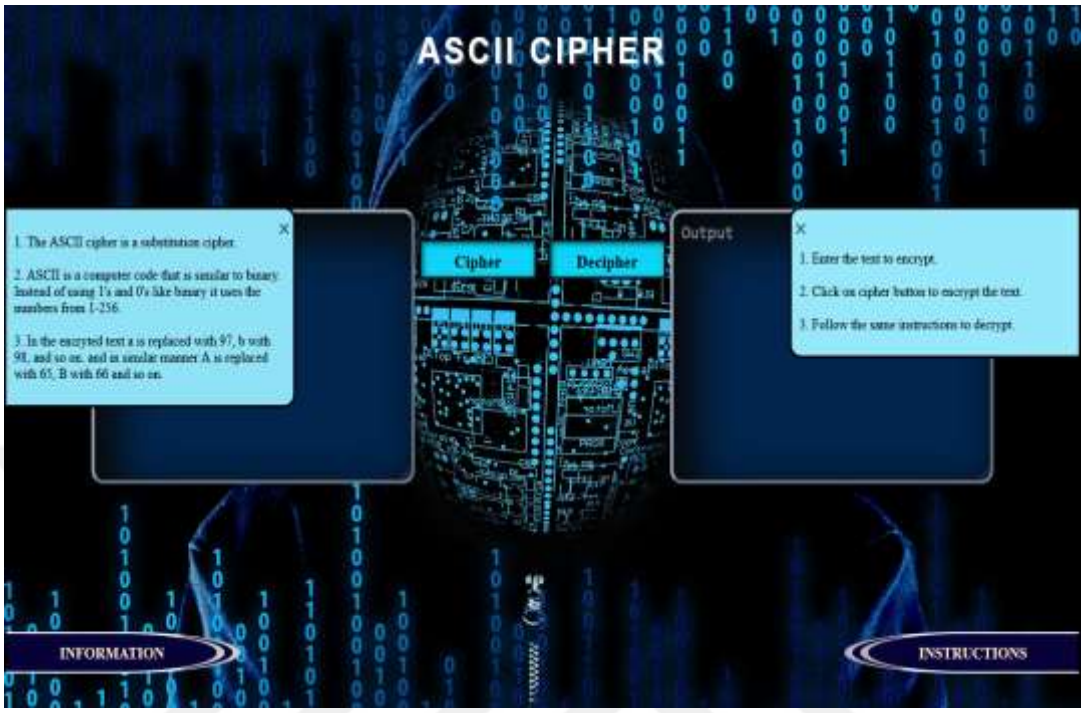

Figure 4.9: ASCII cipher.

4.15 ATBASH CIPHER

In cryptography the ATBASH cipher can define as the following and shown in Figure 4.10 below.

i. The Atbash cipher is a substitution cipher.

ii. It has a fixed key where the letters of the alphabet are reversed. I.e. all 'A's are replaced with 'Z's, all 'B's are replaced with 'Y's, and so on.

iii. The Atbash cipher offers almost no security, and can be broken very easily because of fixed key.

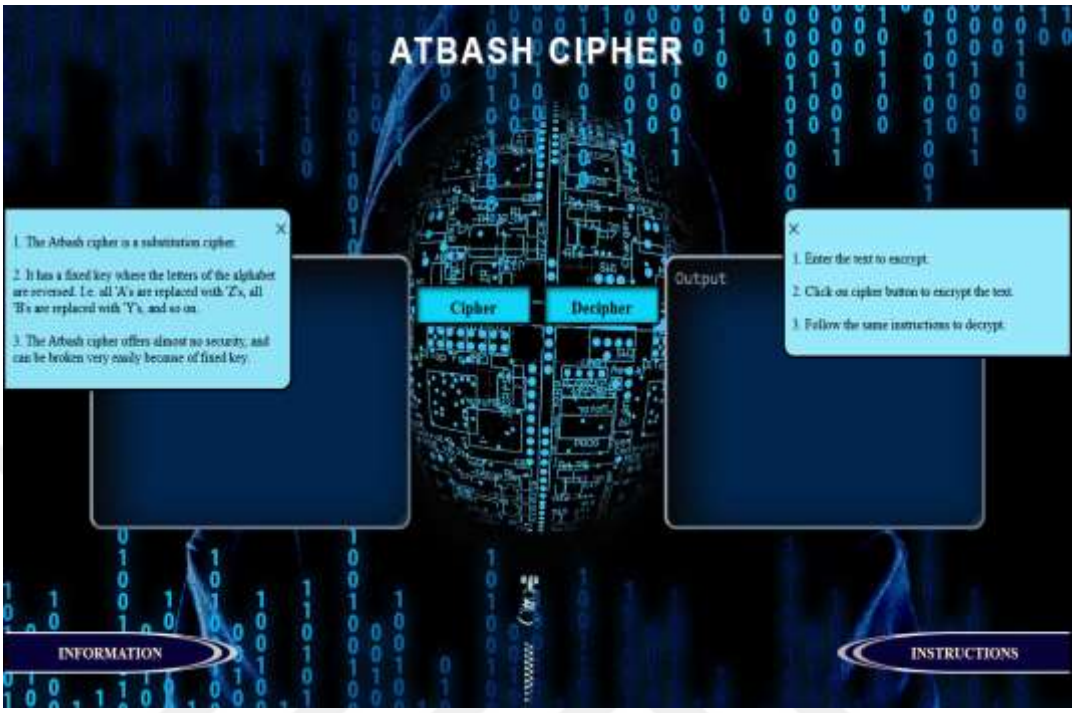

Figure 4.10: ATBASH cipher.

4.16 AUTOKEY CIPHER

In cryptography the Autokey cipher can define as the following and shown in Figure 4.11 below.

i. The Autokey Cipher is a polyalphabetic substitution cipher.

ii. It is closely related to the Vigenere cipher but uses a different method of generating the key.

iii. It was invented by Blaise de Vigenère in 1586 and is in general more secure than the Vigenere cipher.

iv. The 'key' for the Autokey cipher is a key word.

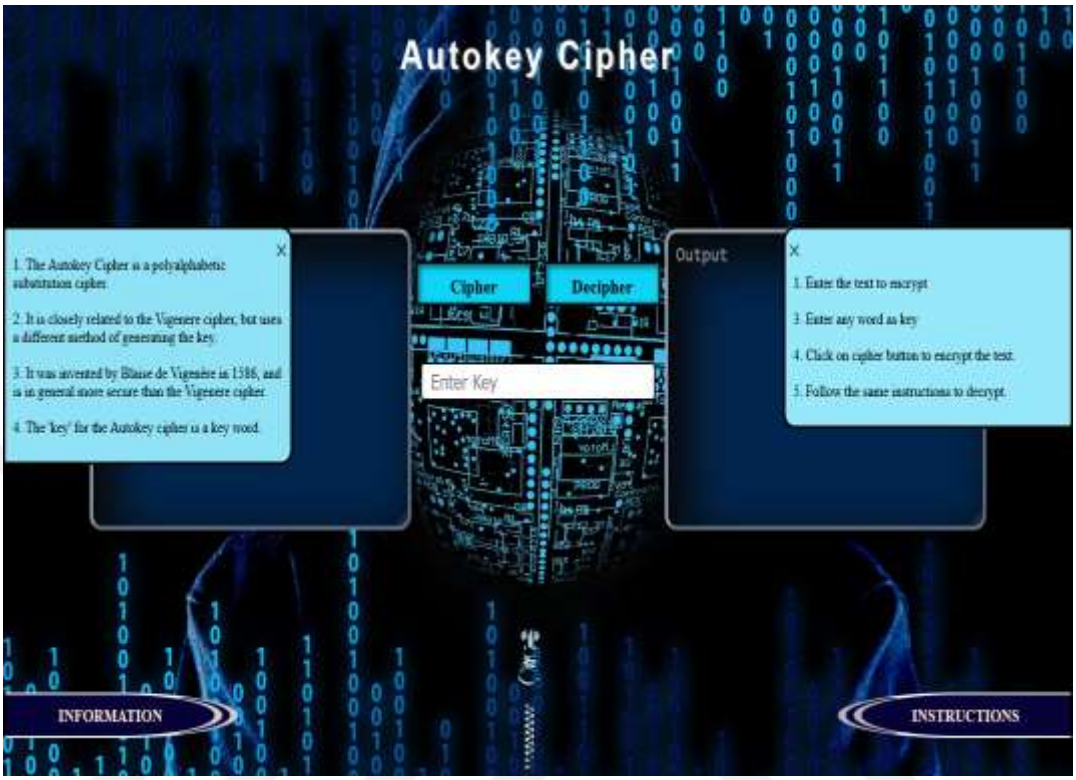

Figure 4.11: Autokey cipher.

4.17 BACONIAN CIPHER

In cryptography the BACONIAN cipher can define as the following and shown in Figure 4.12 below.

i. The Baconian cipher is a substitution cipher.

ii. The Baconian cipher is named after its inventor, Sir Francis Bacon.

iii. In this cipher each letter is replaced by a sequence of 5 characters.

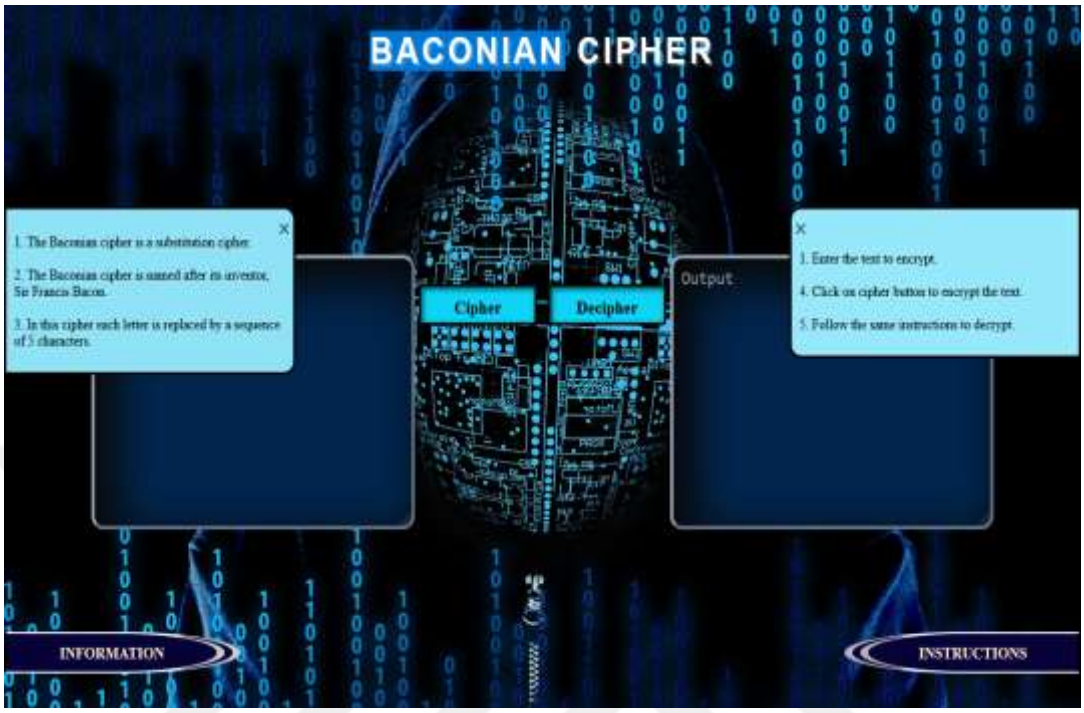

Figure 4.12: BACONIAN cipher.

4.18 BASE64 CIPHER

In cryptography the Base64 cipher can define as the following and shown in Figure 4.13 below.

i. Base64 is not exactly a cipher since there is no key. It's a way to mask a plaintext

ii. The encrypted text resulting from this cipher will often end in an equal sign (not always). There can be 0, 1 or 2 equals signs on the end, the exact number depends on the length of the input.

iii. This encrypted text has many uppercase and lowercase characters.

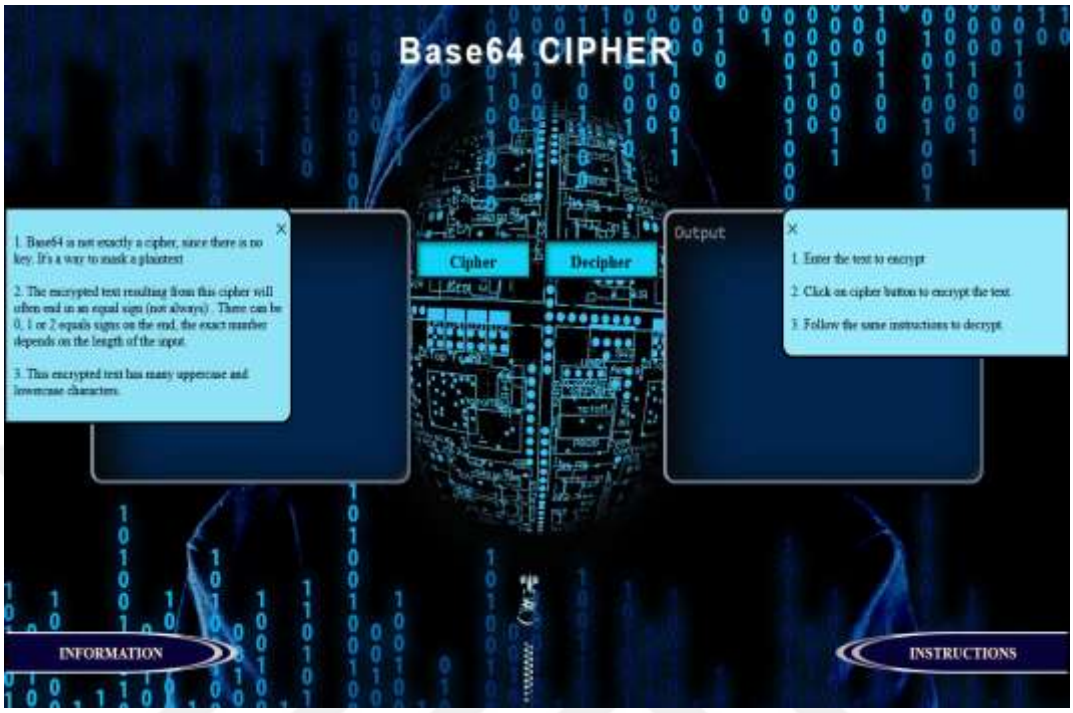

Figure 4.13: Base64 cipher.

4.19 BEAUFORT CIPHER

In cryptography the Beaufort cipher can define as the following and shown in Figure 4.14 below.

i. Beaufort is a polyalphabetic substitution cipher.

ii. The Beaufort cipher was created by Sir Francis Beaufort.

iii. It is similar to the Vigenère cipher, except that it enciphers characters in a slightly different manner.

iv. The 'key' for a beaufort cipher can be any word.

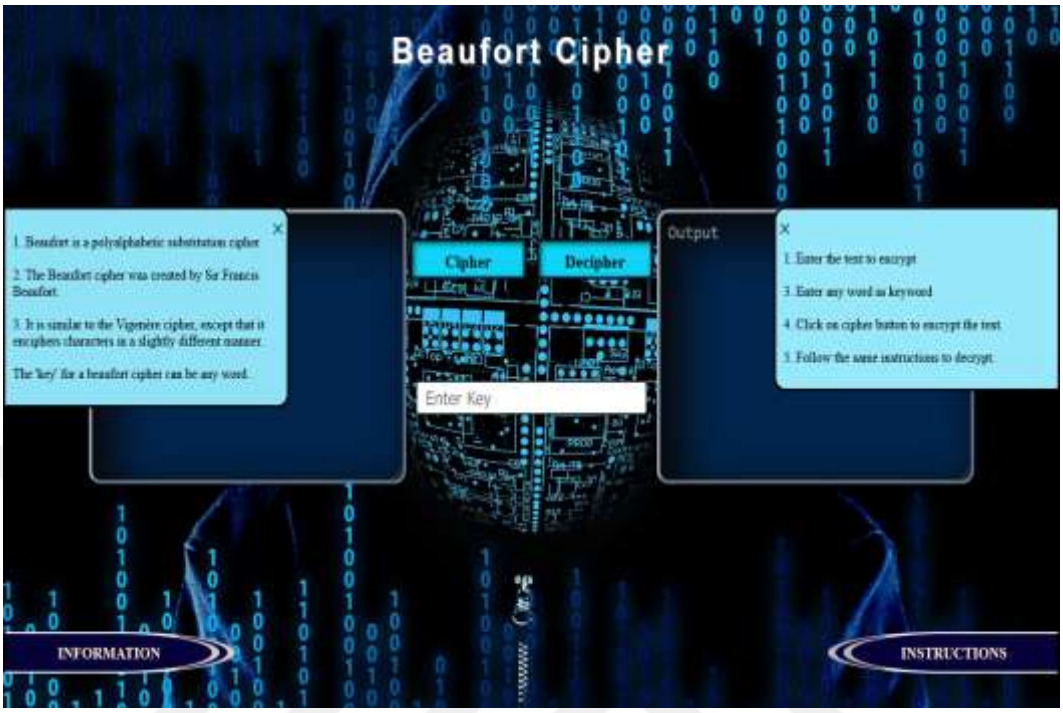

Figure 4.14: Beaufort cipher.

4.20 BIFID CIPHER

In cryptography the BIFID cipher can define as the following and shown in Figure 4.15 below.

i. Bifid is a cipher which combines the Polybius square with transposition and uses fractionation to achieve diffusion.

ii. This cipher was a fractionating transposition cipher which combined a modified Polybius square with a single columnar transposition.

iii. Keys for the Bifid cipher consist of a 25 letter 'key square and a period.

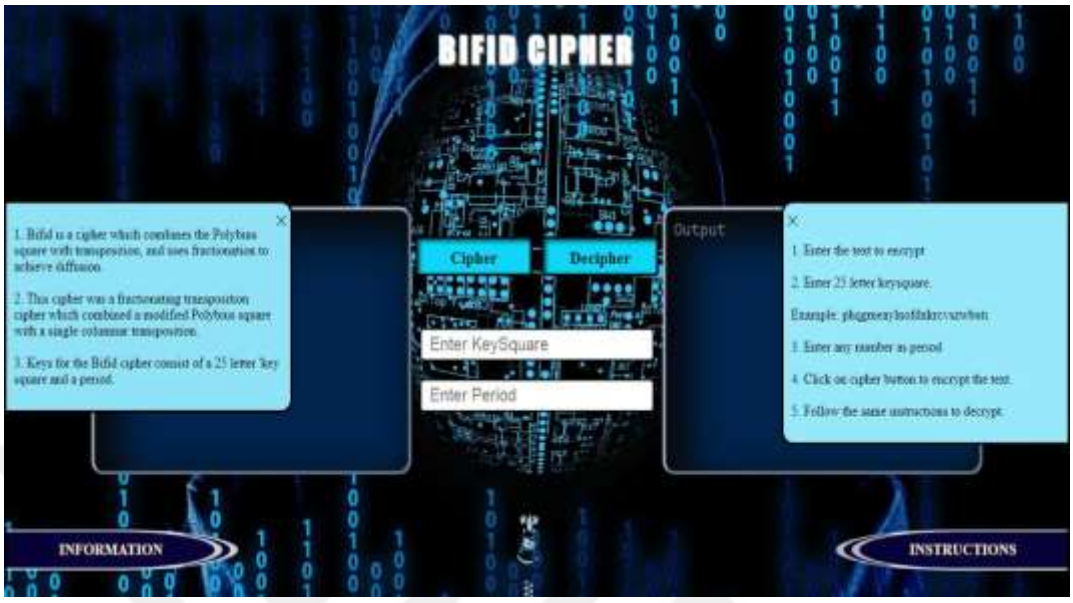

Figure 4.15: BIFID cipher.

4.21 BINARY CIPHER

In cryptography the BINARY cipher can define as the following and shown in Figure 4.16 below.

i. The Binary cipher is a substitution cipher.

ii. The Binary code is a code that the computers recognize using only 1's and 0's.

iii. It is a very complicated code because they are different for lowercase and capital. For example 'A' is replaced with 01000001 where as 'a' is replaced with 01100001.

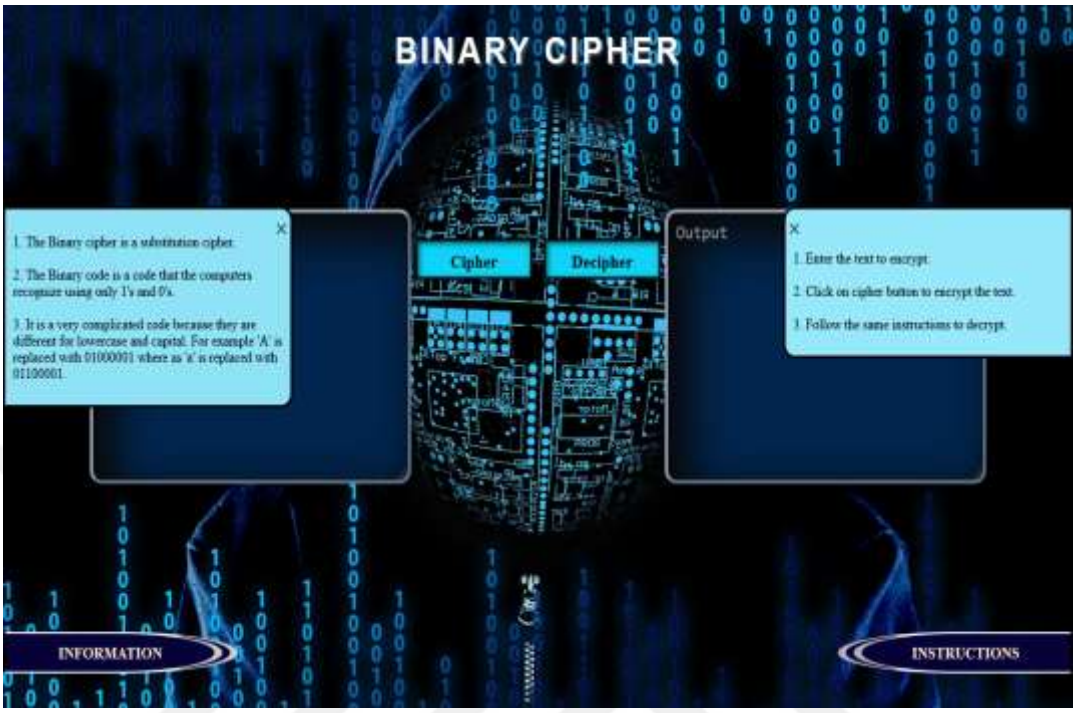

Figure 4.16: BINARY cipher.

4.22 CAESAR CIPHER

In cryptography the CAESAR cipher can define as the following and shown in Figure 4.17 below.

i. The Caesar cipher is a type of substitution cipher.

ii. It is one of the earliest known and simplest ciphers.

iii. In this cipher each letter in the plaintext is 'shifted' a certain number of places down the alphabet.

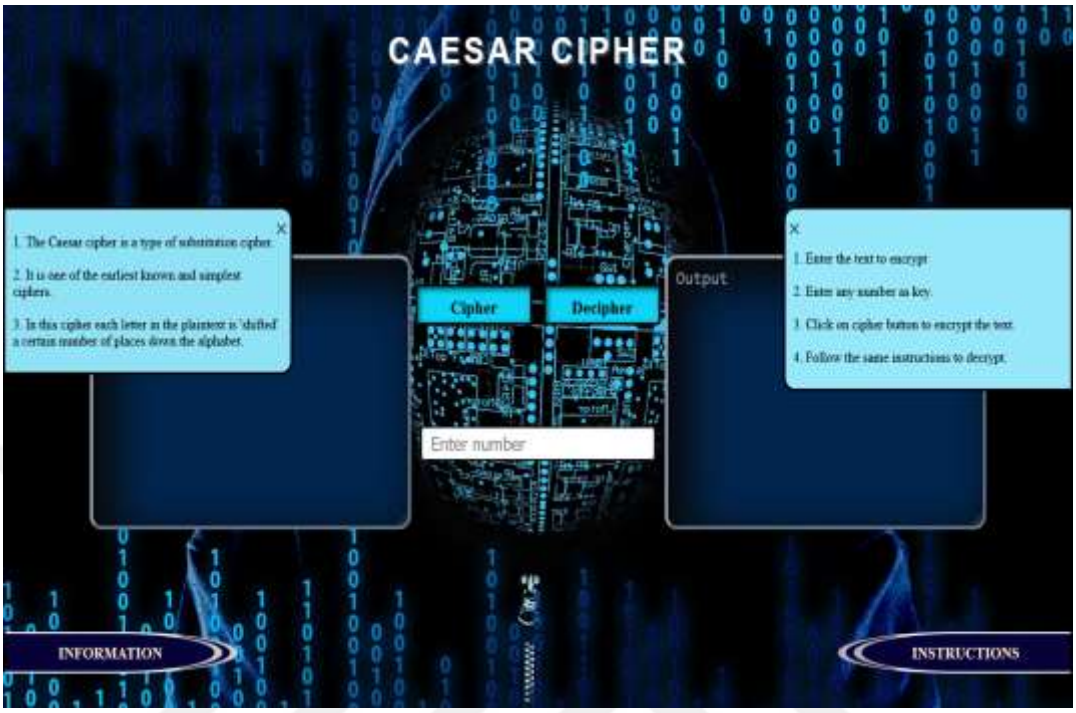

Figure 4.17: CAESAR cipher.

4.23 COLUMNAR TRANSPOSITION CIPHER

In cryptography the Columnar Transposition cipher can define as the following and shown in Figure 4.18 below.

i. The columnar transposition cipher is a fairly simple, easy to implement cipher.

ii. It is a transposition cipher.

iii. It follows a simple rule for mixing up the characters in the plaintext to form the ciphertext.

iv. The key for the columnar transposition cipher is a keyword. The row length that is used is the same as the length of the keyword used to encrypt a piece of text.

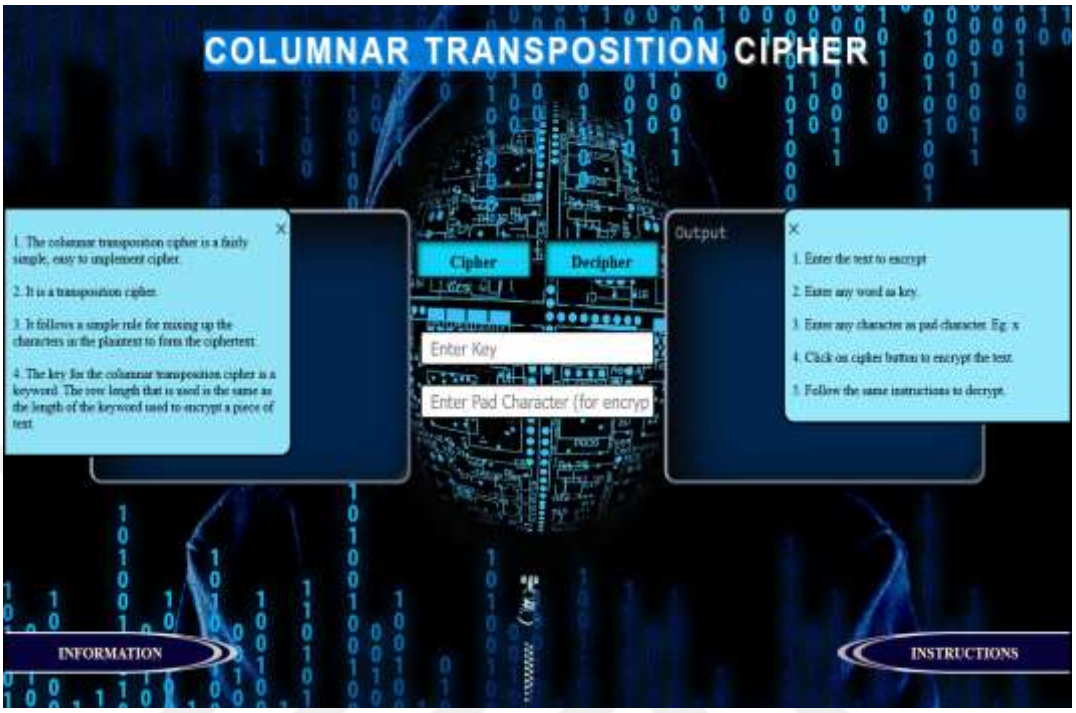

Figure 4.18: Columnar transposition cipher.

4.24 ENIGMA CIPHER

In cryptography the Enigma cipher can define as the following and shown in Figure 4.19 below.

i. The Enigma cipher was a field cipher used by the Germans during World War II.

ii. The Enigma is one of the better-known historical encryption machines, and it refers to a range of similar cipher machine

iii. The 'key' for the enigma is unique compared to other ciphers.

iv. It includes a rotor mechanism that scrambles the 26 letters of the alphabet.

v. There are three letter inputs for the rotor.

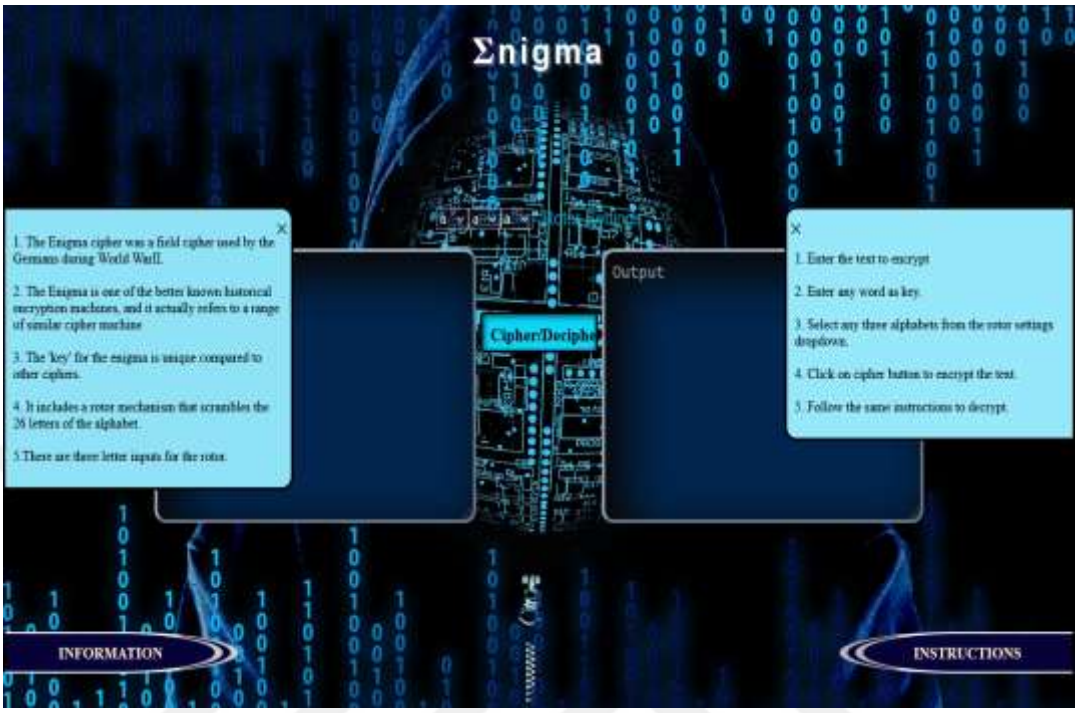

Figure 4.19: Enigma cipher.

4.25 FOUR-SQUARE CIPHER

In cryptography the Four-square cipher can define as the following and shown in Figure 4.20 below.

i. The Four-square cipher encrypts pairs of letters (like playfair), which makes it significantly stronger than substitution ciphers etc. since frequency analysis becomes much more difficult.

ii. Felix Delastelle (1840 - 1902) invented the four-square cipher, first published in a book in 1902.

iii. The four-square cipher uses four 5 by 5 matrices arranged in a square. Each of the 5 by 5 matrices contains 25 letters, usually the letter 'j' is merged with 'i'

iv. The ciphertext squares can be generated using a keyword (dropping duplicate letters), then fill the remaining spaces with the remaining letters of the alphabet in order. Alternatively, the ciphertext squares can be generated completely randomly.

v. The four-square algorithm allows for two separate keys, one for each of the two ciphertext matrices.

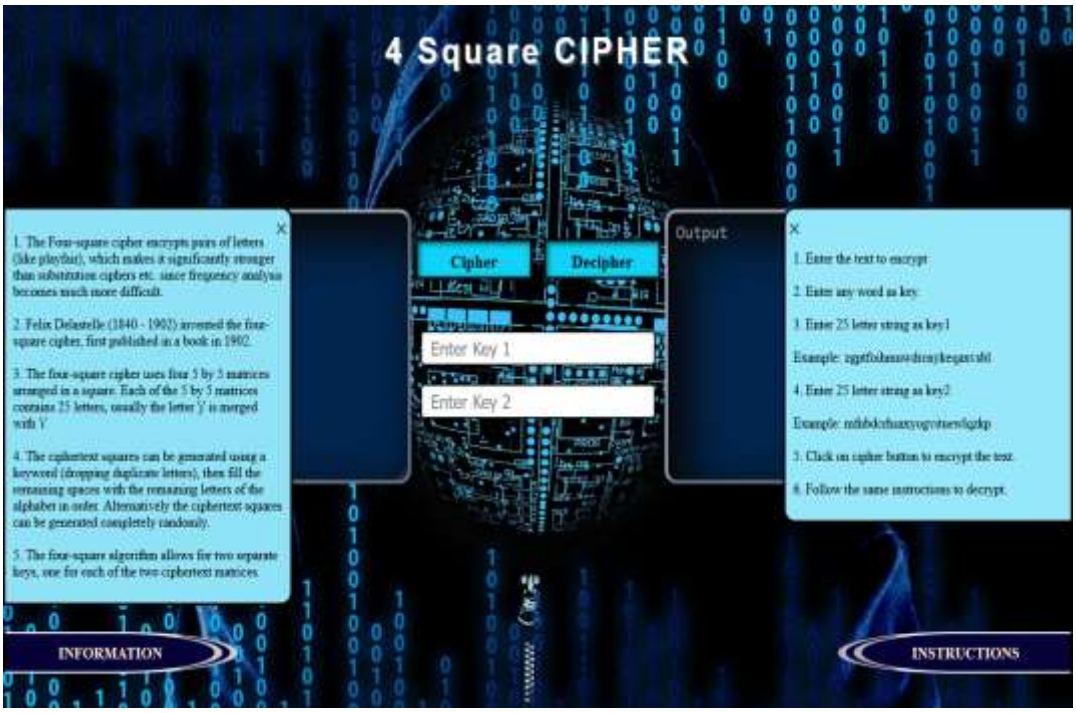

Figure 4.20: Four-square cipher**.**

4.26 HILL CIPHER

In cryptography the HILL cipher can define as the following and shown in Figure 4.21 below.

i. It is a Polygraphic substitution cipher based on linear algebra.

ii. It uses matrices and matrix multiplication to mix up the plaintext.

iii. It requires a branch of mathematics known as number theory for analysis of this algorithm.

iv. The key for a hill cipher is a matrix.

v. The 'key' should be input as 4 numbers. These numbers will form the key (top row, bottom row).

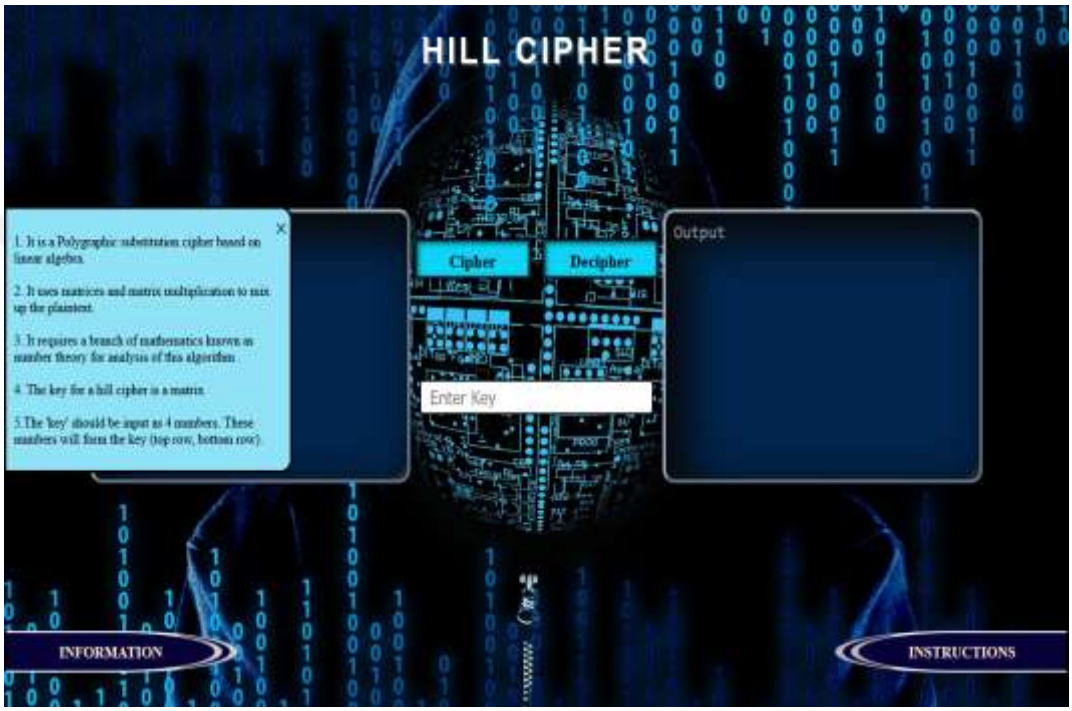

Figure 4.21: HILL cipher.

4.27 MORSE CODE CIPHER

In cryptography the Morse Code cipher can define as the following and shown in Figure 4.22 below.

i. The morse code cipher could be considered a substitution cipher.

ii. Morse code was invented by Samuel Morse and was used in the early 1800's to message people in a telegram. It is a series of beeps that are short and long.

iii. For example, _ is called a dash and it would be a long beep, and. would be dot and it would be a short beep.

iv. In this cipher each letter and number in the plain text are replaced by a unique series of dots and dashes representing the letters and numbers.

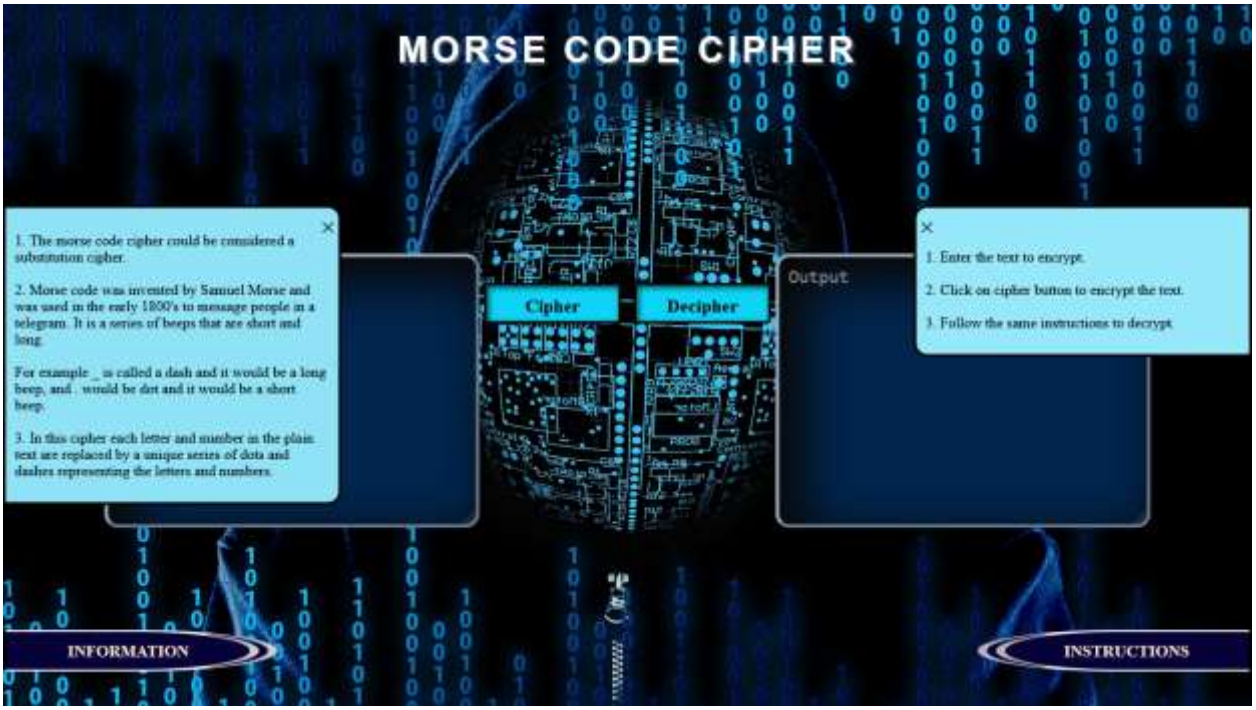

Figure 4.22: Morse code cipher.

4.28 MULTIPLICATIVE CIPHER

In cryptography the Multiplicative cipher can define as the following and shown in Figure 4.23 below.

i. The basic formula to be used to generate a multiplicative cipher is as follows −

ii. (Alphabet Number * key) mod (total number of alphabets)

iii. The number fetched through output is mapped in a specific table and the corresponding letter is taken as the encrypted letter.

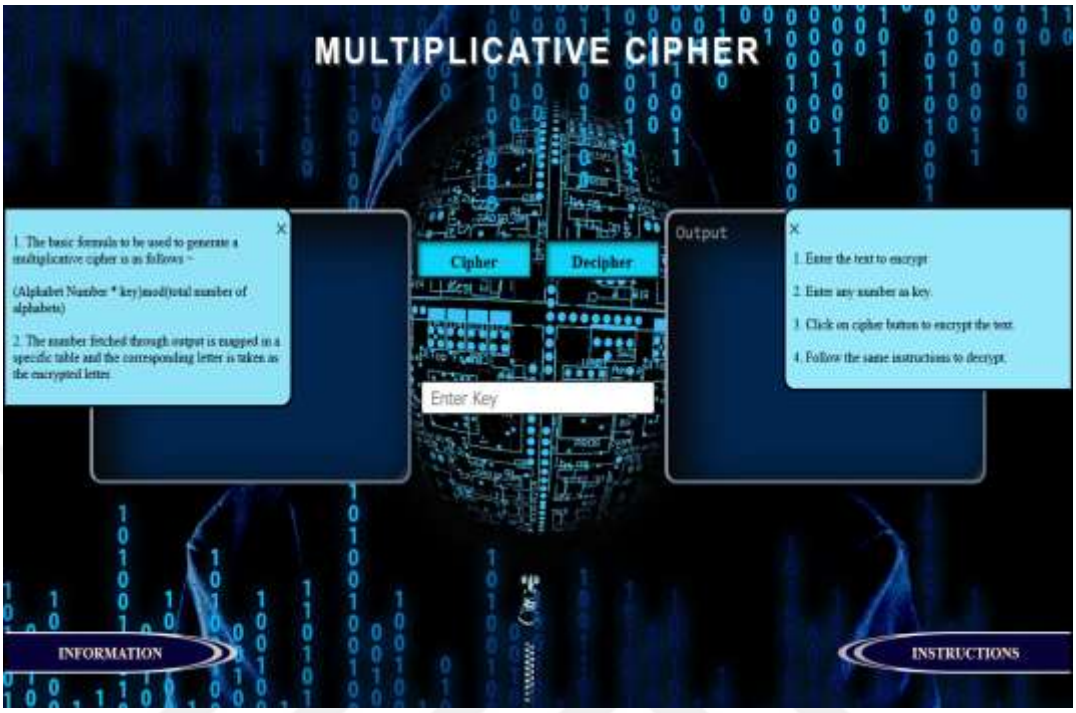

Figure 4.23: Multiplicative cipher.

4.29 PHONE CIPHER

In cryptography the Phone cipher can define as the following and shown in Figure 4.24 below.

i. Phone cipher is a unique cipher.

ii. It works in the following manner cipher text for an alphabet is the number the letter is on and then the number on which the previous number is on.

For example ,

A is on number 1 which is on 2 so the cipher text for A would be would be 2 1

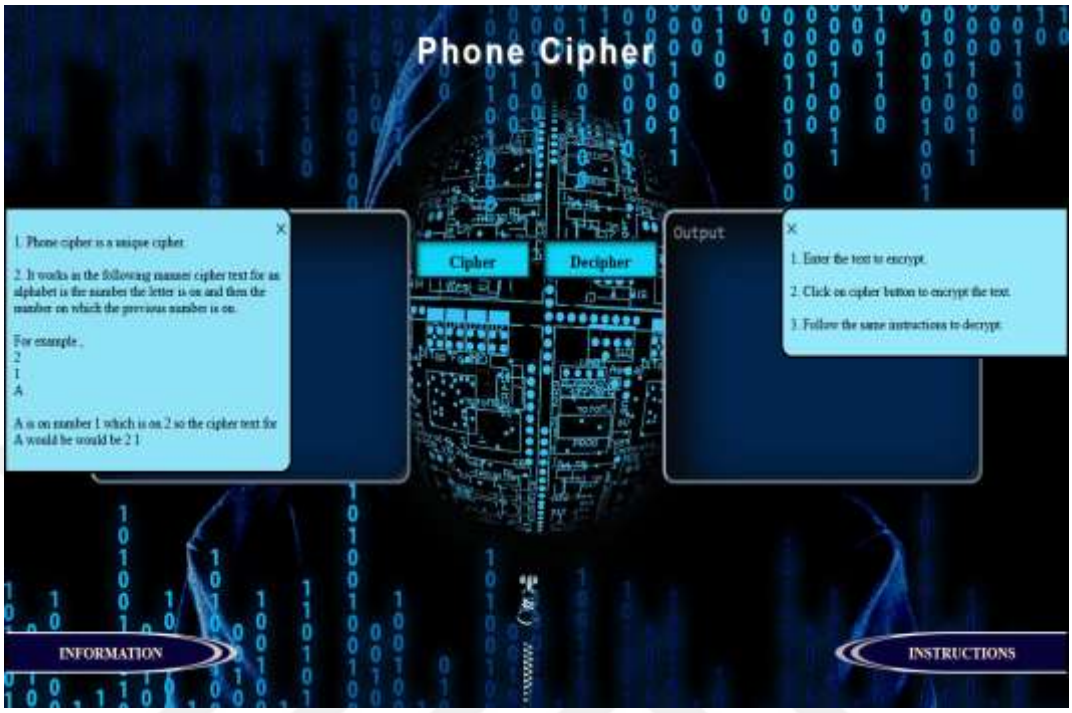

Figure 4.24: Phone cipher.

4.30 PLAYFAIR CIPHER

In cryptography the Playfair cipher can define as the following and shown in Figure 4.25 below.

i. The Playfair cipher was the first practical digraph substitution cipher.

ii. This was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher.

iii. It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.

iv. The technique encrypts pairs of letters (digraphs), instead of single letters as in the simple substitution cipher.

v. The Playfair is significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it.

vi. Any sequence of 25 letters can be used as a key.

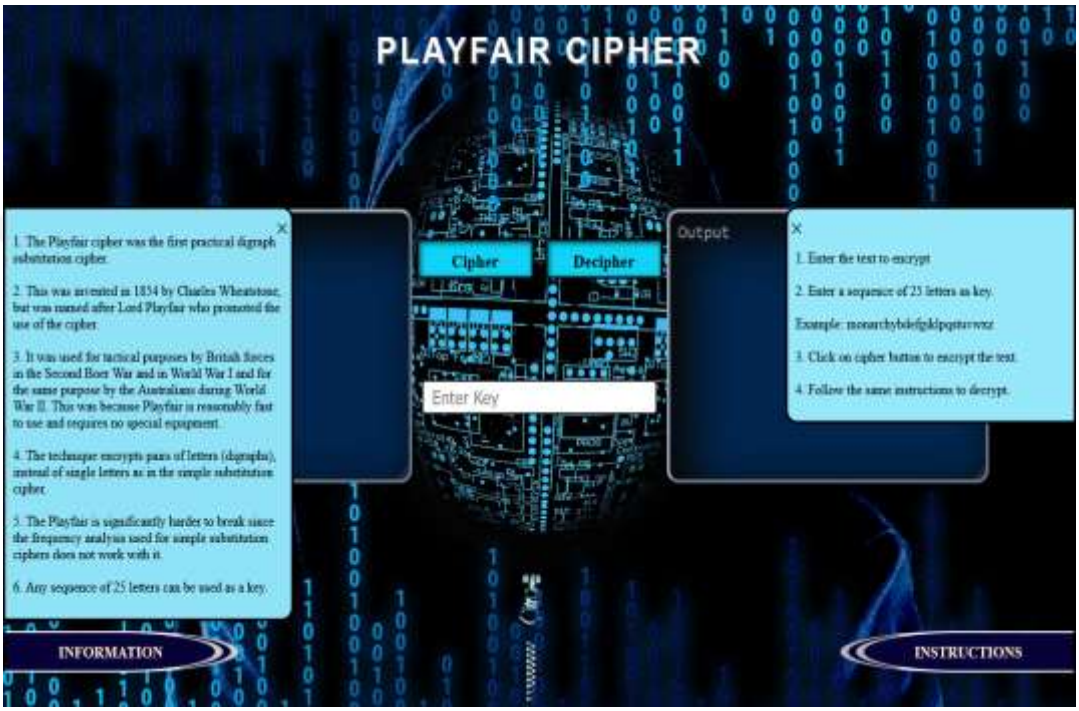

Figure 4.25: Playfair cipher.

4.31 POLYALPHABETIC CIPHER

In cryptography the Polyalphabetic cipher can define as the following and shown in Figure 4.26 below.

i. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets

ii. In this cipher the alphabet for the plain text alphabet may be different at different places during the encryption process

iii. Any word can be used as a key.

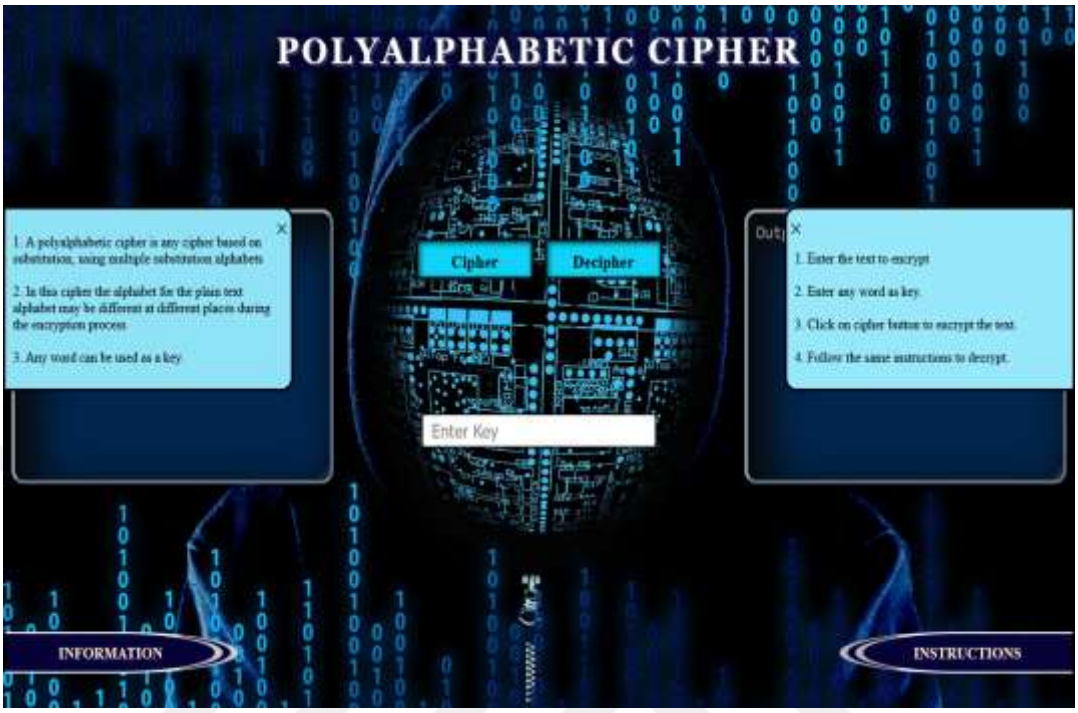

Figure 4.26: Polyalphabetic cipher.

4.32 POLYBIUS CIPHER

In cryptography the Polybius cipher can define as the following and shown in Figure 4.27 below.

i. The Polybius Square is essentially identical to the simple substitution cipher, except that each plaintext character is enciphered as 2 ciphertext characters

ii. Each character in the plaintext is replaced with 2 characters in the cipher alphabet

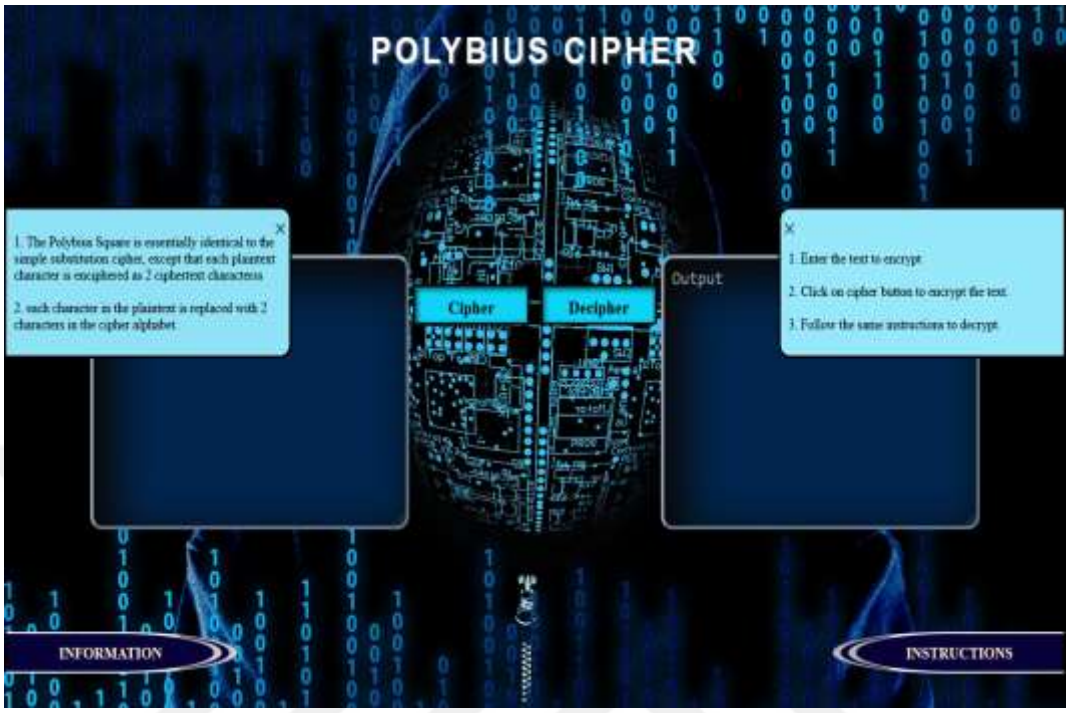

Figure 4.27: Polybius cipher.

4.33 PORTA CIPHER

In cryptography the Porta cipher can define as the following and shown in Figure 4.28 below.

i. The Porta Cipher is a polyalphabetic substitution cipher

ii. This was invented by Giovanni Battista della Porta.

iii. It is similar to the Vigenere cipher, which is a polyalphabetic cipher with 26 alphabets, the Porta is basically the same except it only uses 13 alphabets.

iv. The 13 cipher alphabets it uses are reciprocal, so enciphering is the same as deciphering.

v. The 'key' for a porta cipher can be any key word

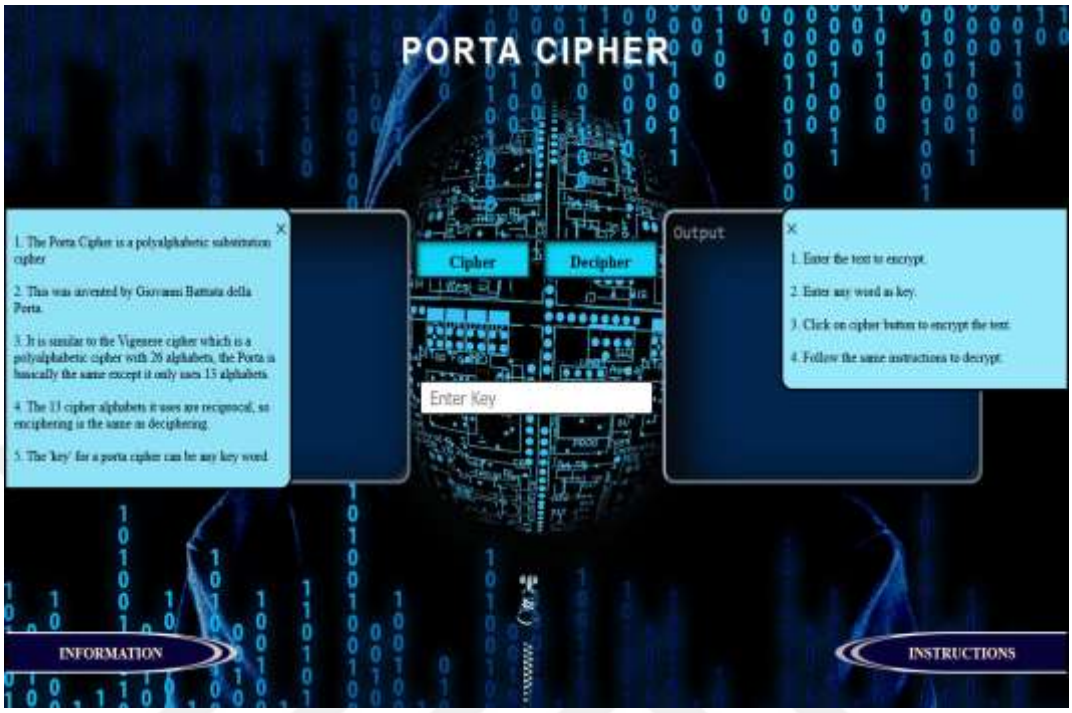

Figure 4.28: Porta cipher.

4.34 RAIL-FENCE CIPHER

In cryptography the Rail-fence cipher can define as the following and shown in Figure 4.29 below.

i. The Rail-fence cipher is a very simple, easy to crack cipher

ii. It is a transposition cipher that follows a simple rule for mixing up the characters in the plaintext to form the ciphertext.

iii. The key for the Rail-fence cipher is the number of rails.

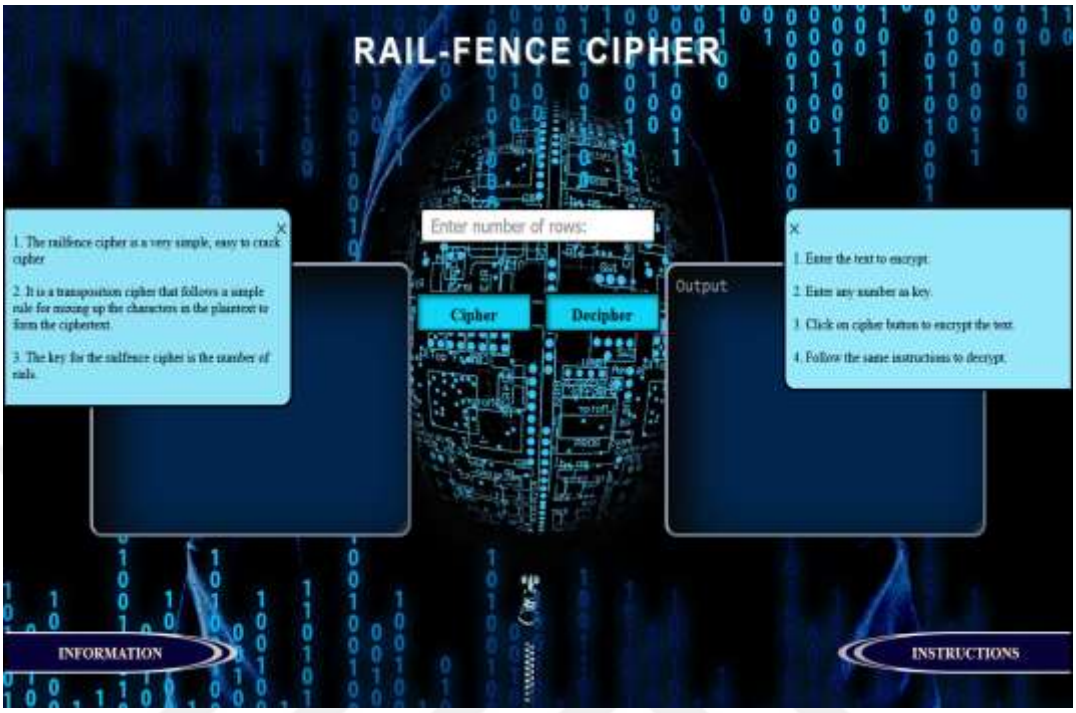

Figure 4.29: Rail-fence cipher.

4.35 REVERSE CIPHER

In cryptography the Reverse cipher can define as the following and shown in Figure 4.30 below.

i. Reverse Cipher uses a pattern of reversing the string of plain text to convert as cipher text.

ii. The process of encryption and decryption is same.

iii. To decrypt cipher text, the user simply needs to reverse the cipher text to get the plain text.

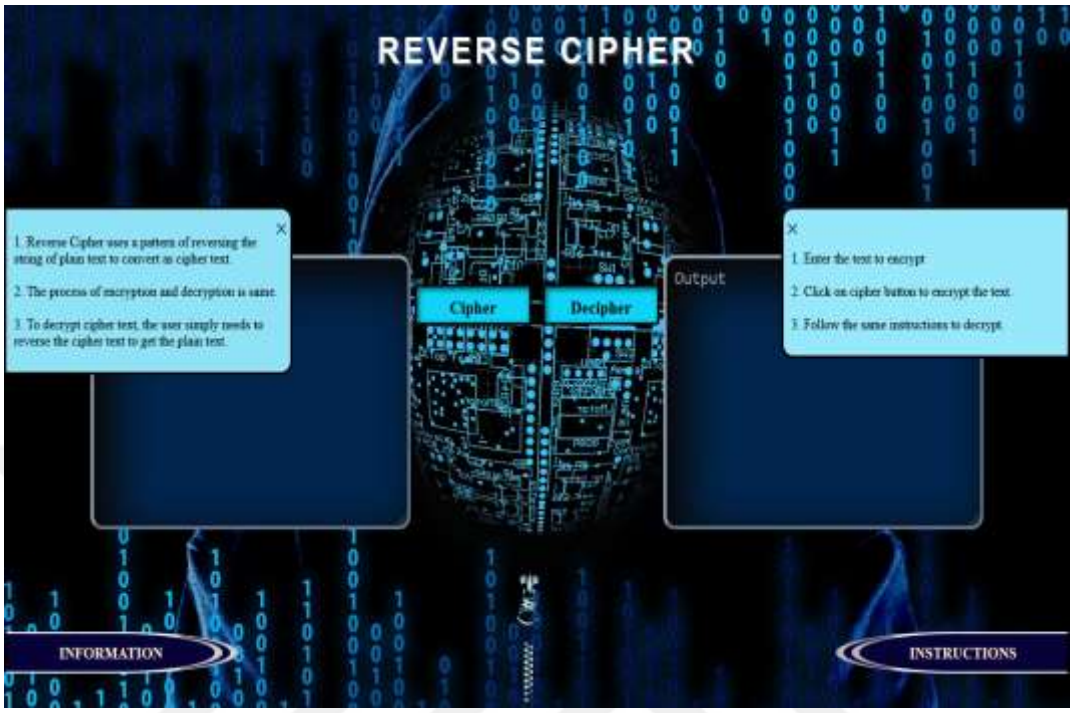

Figure 4.30: Reverse cipher.

4.36 ROT13 CIPHER

In cryptography the ROT13 cipher can define as the following and shown in Figure 4.31 below.

i. The ROT13 cipher is a substitution cipher.

ii. It has a fixed key where the letters of the alphabet are offset 13 places. I.e., all 'A's are replaced with 'N's, all 'B's are replaced with 'O's, and so on. It can also be viewed of as a Caesar cipher with a shift of 13.

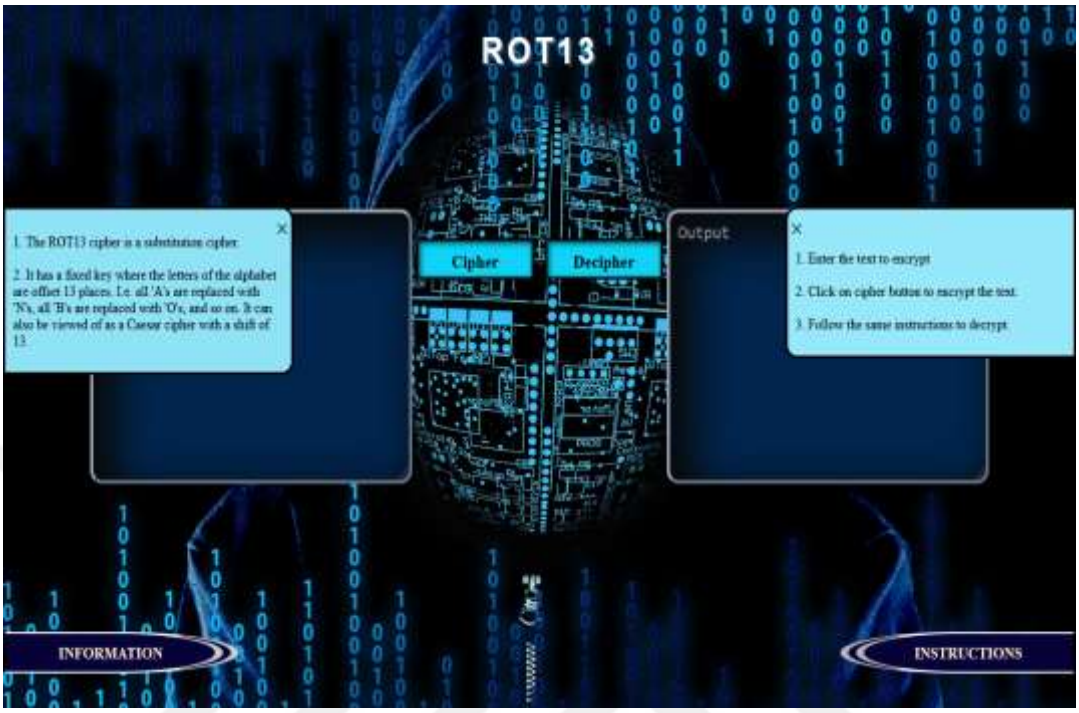

Figure 4.31: ROT13 cipher.

4.37 RUNNING KEY CIPHER

In cryptography the Running Key cipher can define as the following and shown in Figure 4.32 below.

i. The Running Key cipher has the same internal workings as the Vigenere cipher. The difference lies in how the key is chosen; the Vigenere cipher uses a short key that repeats, whereas the running key cipher uses a long key such as an excerpt from a book. This means the key does not repeat, making cryptanalysis more difficult.

ii. The 'key' for a running key cipher is a long piece of text, e.g., an excerpt from a book.

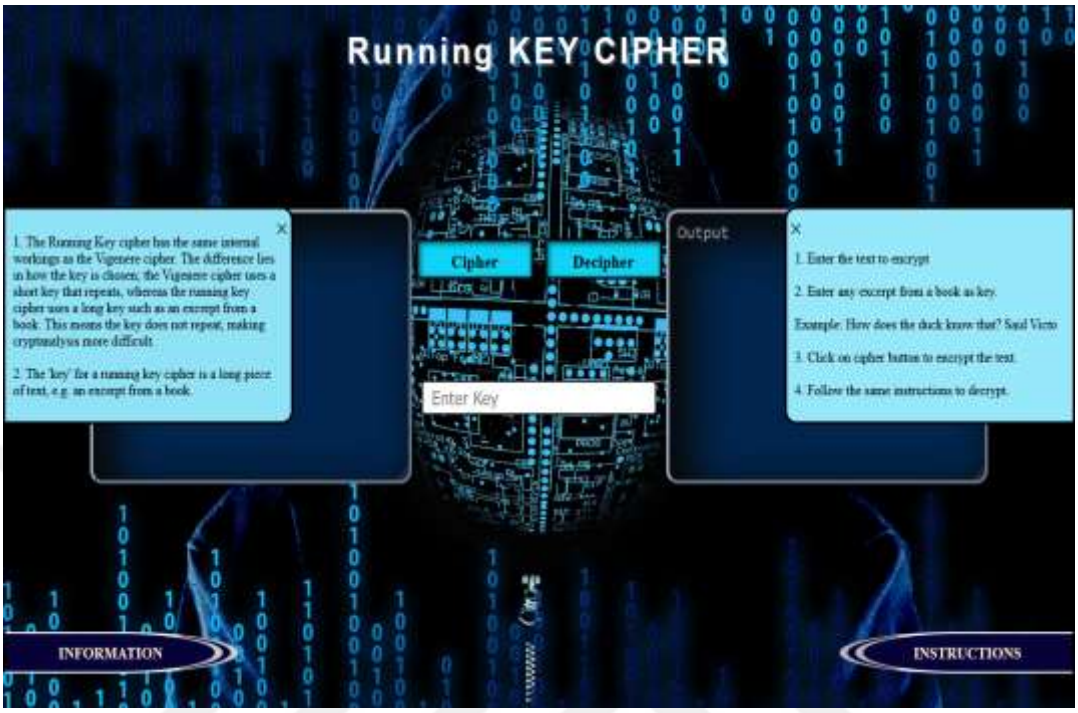

Figure 4.32: Running key cipher.

4.38 STRADDLE CHECKERBOARD CIPHER

In cryptography the Straddle Checkerboard cipher can define as the following and shown in Figure 4.33 below.

i. The straddling checkerboard is a substitution cipher, except that the substitutions are of variable length.

ii. The key for a straddling checkerboard is a permutation of the alphabets along with 2 numbers.

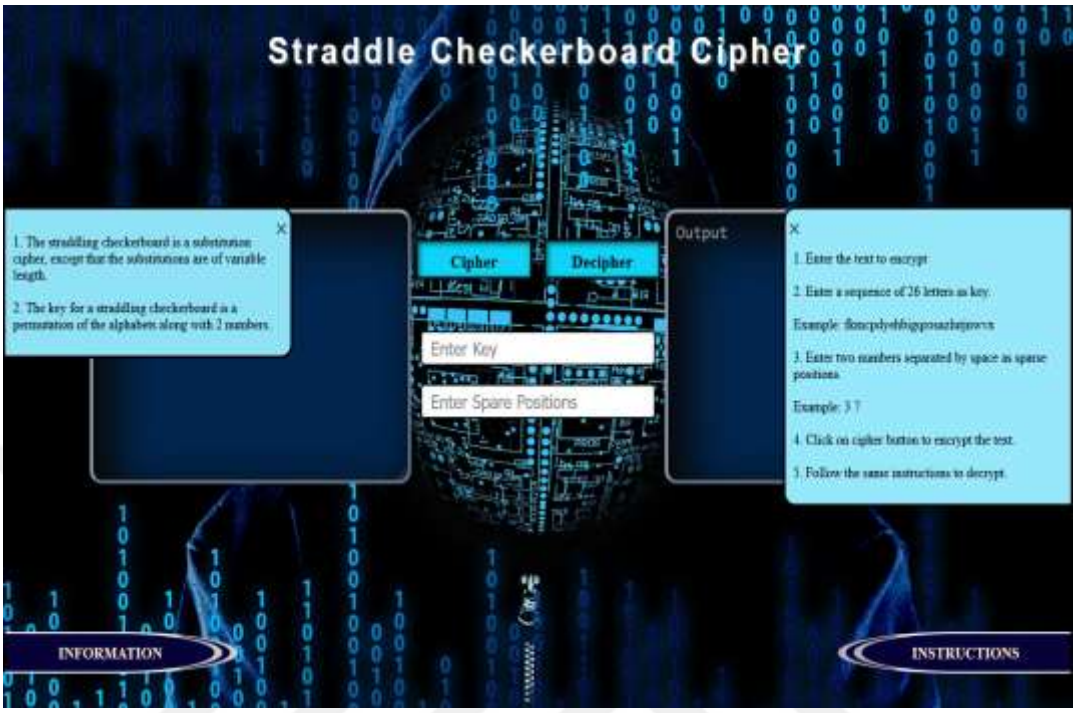

Figure 4.33: Straddle checkerboard cipher.

4.39 VERNAM CIPHER

i. In cryptography the VERNAM cipher can define as the following and shown in Figure 4.34 below.

ii. It is a transposition cipher since each plaintext character is encrypted using its own key instead of a single key,

iii. This key or key stream is randomly generated or is taken from a onetime pad, e.g. a page of a book.

iv. Length of the key to encrypt the plain text should be equal to the length of the plain text

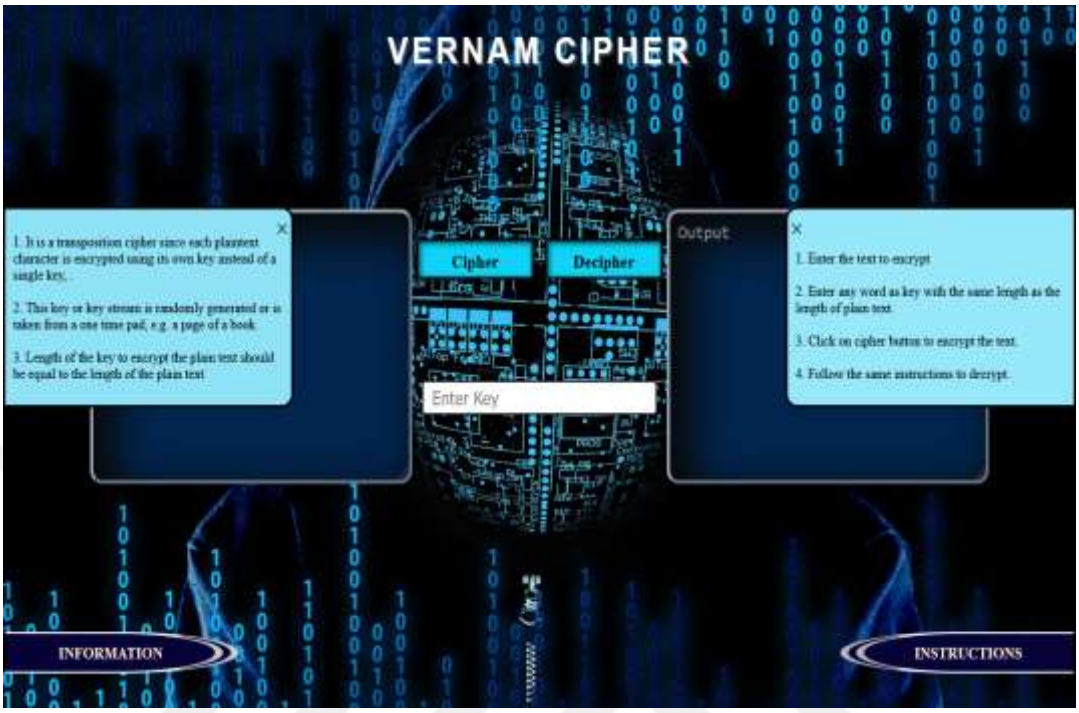

Figure 4.34: VERNAM cipher.

4.40 VIGENERE CIPHER

In cryptography the Vigenère cipher can define as the following and shown in Figure 4.35 below.

i. The Vigenère Cipher is a polyalphabetic substitution cipher.

ii. The method was originally described by Giovan Battista Bellaso in his 1553 book La cifra del. Sig. Giovan Battista Bellaso; however, the scheme was later misattributed to Blaise de Vigenère in the 19th century and is now widely known as the Vigenère cipher.

iii. The Vigenère Cipher was considered le chiffre ind hiffrable (French for the unbreakable cipher) for 300 years, until in 1863.

iv. The 'key' for a vigenere cipher can be any word.

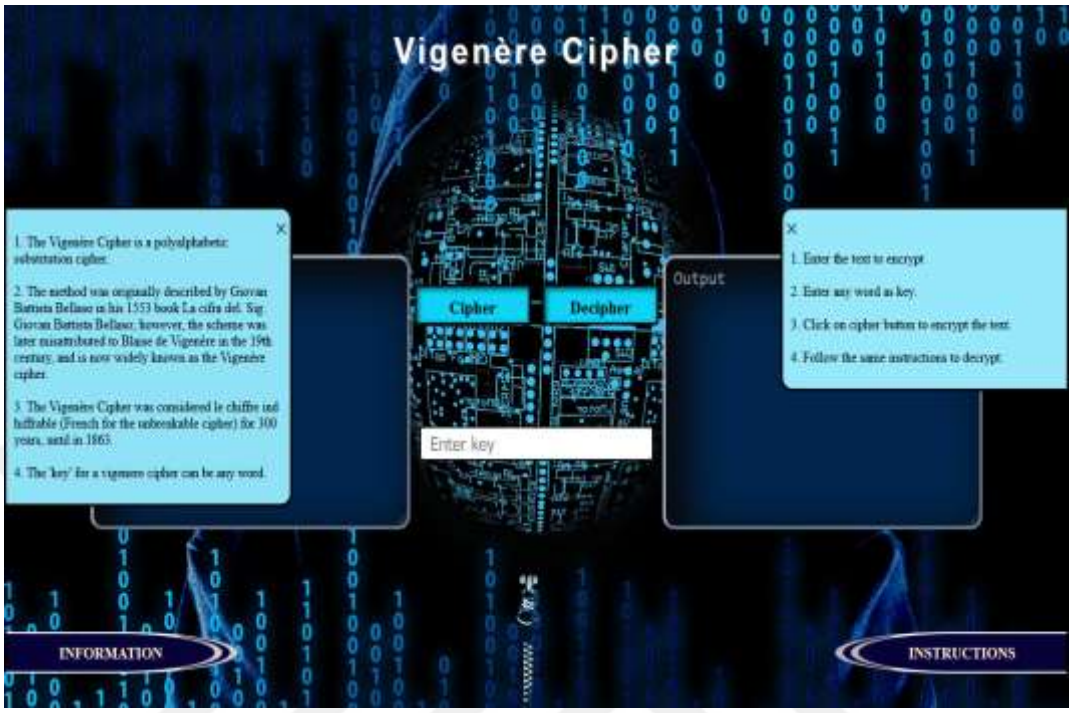

Figure 4.35: Vigenère cipher.

4.41 ZIP CIPHER

In cryptography the Zip cipher can define as the following and shown in Figure 4.36 below.

The Zip cipher is a not a common cipher.

It follows a Unique approach for encryption and decryption.

On encryption. It splits a string in half and zips them together after they have been reversed.

On decryption, it unzips the encoded message, reverses the two strings, and joins them together.

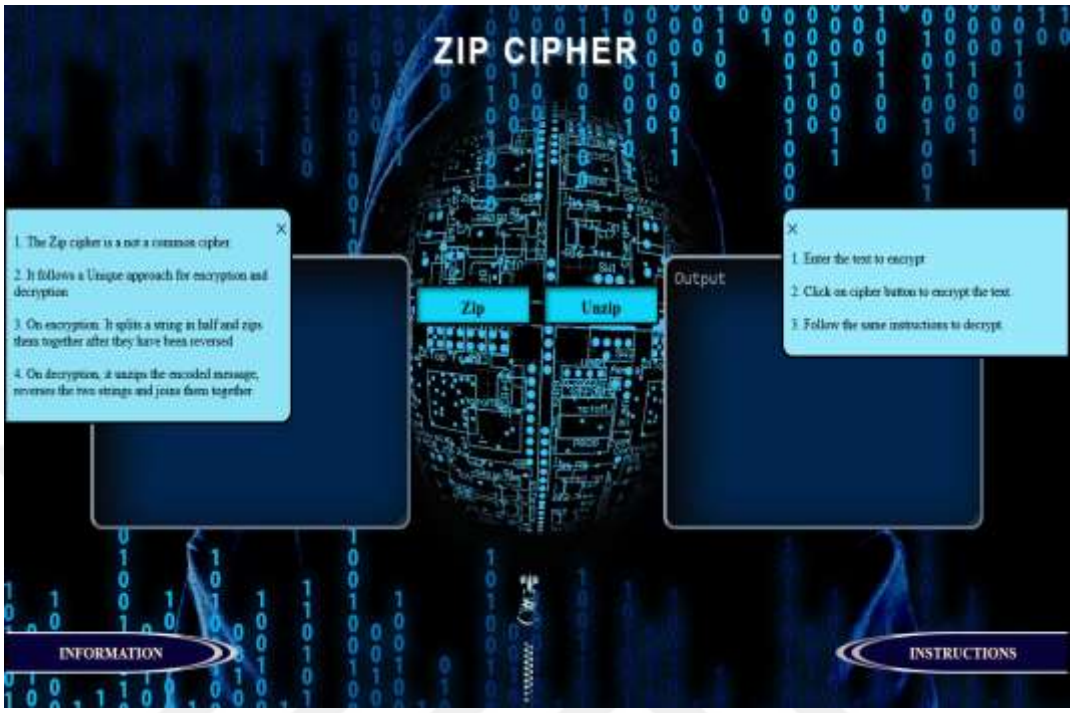

Figure 4.36: Zip cipher.

4.42 SUMMARY

In this chapter, we looked at the study's findings and their potential applications, and we discussed Answers and explanations to the hypotheses and research questions were provided, and the significance of the findings was thoroughly discussed in light of the aforementioned studies and literature. The research findings will be summarized in the following and final chapter. It will talk about the research's design and implementation flaws, and it will suggest directions for future studies based on the findings.

5. CONCLUSION AND FUTURE WORKS

5.1 CONCLUSION

The primary objective of this project is to offer students the opportunity to participate in hands-on experiments, in addition to providing visual representations of encryption algorithms, with the goal of assisting students in gaining a deeper understanding of the theoretical concepts that underlie cryptographic techniques used for network security. It is beneficial to use Cryptographic as a tool to get there because it enables anyone who is interested in cryptography to learn about it without cost, and the source code of the software can be downloaded and modified for specific uses. This makes using Cryptographic an effective tool.

5.2 FUTURE WORKS

Students are able to construct a solid foundation in the fundamentals of cryptography and learn more about the historical context of each algorithm if they begin with the oldest cryptographic methods and work their way up to the most modern ones. This method begins with the oldest and works its way to the most modern. Software such as Cryptographic can assist educators in covering more ground in the field of cryptography by reducing the amount of time spent on theoretical lessons. This frees up more time for them to focus on practical application.

REFERENCES

[1] Afzal, M., Kausar, F., & Masood, A. (2006, 13-14 Nov. 2006). Comparative Analysis of the Structures of eSTREAM Submitted Stream Ciphers. Paper presented at the ICET '06, International Conference on Emerging Technologies, 2006. doi:10.1109/ICET.2006.335958

[2] Agnarsson, G., & Greenlaw, R. (2007). Graph Thory: Modelling, Applications, and Algorithms. New Jersey: Pearson Education Ltd.

[3] Akhter, F. (2015, 26-27 Nov. 2015). A novel Elliptic Curve Cryptography scheme using random sequence. Paper presented at the 2015 International Conference on Computer and Information Engineering (ICCIE). doi:10.1109/CCIE.2015.73993 14

[4] Amara, M., & Siad, A. (2011, 9-11 May 2011). Elliptic Curve Cryptography and its applications. Paper presented at the 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA), 2011. doi:10.1109/WOSSPA.2011.5931464

[5] Amounas, F., & Kinani, E. H. E. (2012, 20-21 April 2012). An elliptic curve cryptography based on matrix scrambling method. Paper presented at the National Days of Network Security and Systems (JNS2), 2012. doi:10.1109/JNS2.2012.6249236

[6] Anderson, R. (2008). Security Engineering: A Guide to Building Dependable Distributed Systems (2nd ed.): John Wiley & Sons.

[7] Arumugam, S., Lakshmanan, R., & Nagar, A. K. (2013). Graph access structures with optimal pixel expansion three. Information and Computation, 230, 67-75. doi:10.1016/J.IC.2013.07.002

[8] Ateniese, G., Blundo, C., De Santis, A., & Stinson, D. R. (1996). Visual Cryptography for General Access Structures. Information and Computation, 129(2), 86-106. doi: 10.1006/INCO.1996.0076

[9] Ateniese, G., Blundo, C., Santis, A. D., & Stinson, D. R. (2001). Extended capabilities for visual cryptography. Theoretical Computer Science, 250(1–2), 143-161. doi:10.1016/S0304-3975(99)00127-9

[10] Austen, J. (2006). Pride and Prejudice. London, Great Britain: Headline Review.

[11] Bai, Q.-h., Zhang, W.-b., Jiang, P., & Lu, X. (2012, 11-13 Aug. 2012). Research on Design Principles of Elliptic Curve Public Key Cryptography and Its Implementation. Paper presented at the International Conference on Computer Science & Service System (CSSS), 2012. doi:10.1109/CSSS.2012.310

[12] Bhat, B., Ali, A. W., & Gupta, A. (2015, 15-16 May 2015). DES and AES performance evaluation. Paper presented at the International Conference on Computing, Communication & Automation (ICCCA), 2015. doi:10.1109/CCAA.2015.714 8500

[13] Blakley, G. R., & Kabatiansky, G. (2011). Secret Sharing Schemes. In H. A. van Tilborg & S. Jajodia (Eds.), Encyclopedia of Cryptography and Security (pp. 1095-1097): Springer US. doi:10.1007/978-1-4419-5906-5_389

[14] Blundo, C., Cimato, S., & De Santis, A. (2006). Visual cryptography schemes with optimal pixel expansion. Theoretical Computer Science, 369(1–3), 169-182. doi:10.1016/J.TCS.2006.08.008

[15] Chan-Hyoung, P., Hong-Yeop, S., & Kyu Tae, P. (1998). Existence and classification of Hadamard matrices. Paper presented at the 1998 Fourth International Conference on Signal Processing Proceedings, 1998. ICSP '98. doi:10.1109/ICOSP.1998. 770165

[16] Chandra, S., Paira, S., Alam, S. S., & Sanyal, G. (2014, 17-18 Nov. 2014). A comparative survey of Symmetric and Asymmetric Key Cryptography. Paper presented at the 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE). doi:10.1109/ICSEMR.2014.7043664

[17] Chen, Y. C., Horng, G., & Tsai, D. S. (2012). Comment on "Cheating Prevention in Visual Cryptography". IEEE Transactions on Image Processing, 21(7), 3319- 3323. doi:10.1109/TIP.2012.2190082

[18] Cohn, P. M. (2000). Introduction to ring theory: Springer Science & Business Media.

[19] Davidoff, G., Sarnak, P., & Valette, A. (2003). Elementary number theory, group theory and Ramanujan graphs (Vol. 55): Cambridge University Press.

[20] De Prisco, R., & De Santis, A. (2014). On the Relation of Random Grid and Deterministic Visual Cryptography. IEEE Transactions on Information Forensics and Security, 9(4), 653-665. doi:10.1109/TIFS.2014.2305574

[21] Delgosha, F., & Fekri, F. (2006). Public-key cryptography using paraunitary matrices. IEEE Transactions on Signal Processing, 54(9), 3489-3504. doi:10.1109/TSP.2006.877670

[22] Deligiannidis, L. (2015, 27-29 May 2015). Elliptic curve cryptography in Java. Paper presented at the IEEE International Conference on Intelligence and Security Informatics (ISI), 2015. doi:10.1109/ISI.2015.7165975

[23] Ding, J., Petzoldt, A., & Wang, L.-C. (2014). The Cubic Simple Matrix Encryption Scheme. Post-Quantum Cryptography, 76.

[24] Ding, J., & Yang, B.-Y. (2009). Multivariate public key cryptography Post-Quantum Cryptography (pp. 193-241): Springer. doi:10.1007/978-3-540-88702-7_6

[25] Droste, S. (1996). New Results on Visual Cryptography. In N. Koblitz (Ed.), Advances in Cryptology — CRYPTO ’96 (Vol. 1109, pp. 401-415): Springer Berlin Heidelberg. doi:10.1007/3-540-68697-5_30

[26] Feng, J.-B., Wu, H.-C., Tsai, C.-S., Chang, Y.-F., & Chu, Y.-P. (2008). Visual secret sharing for multiple secrets. Pattern Recognition, 41(12), 3572-3581. doi:10.1016/J.PATCOG.2008.05.031

[27] Fluhrer, S., Mantin, I., & Shamir, A. (2001). Weaknesses in the key scheduling algorithm of RC4. Paper presented at the International Workshop on Selected Areas in Cryptography.

[28] Galbraith, S., & Menezes, A. (2005). Algebraic curves and cryptography. Finite Fields and Their Applications, 11(3), 544-577. doi:10.1016/J.FFA.2005.05.001

[29] Giorgobiani, G., Kvaratskhelia, V., & Menteshashvili, M. (2015, Sept. 28 2015-Oct. 2 2015). Some properties of Hadamard matrices. Paper presented at the Computer Science and Information Technologies (CSIT), 2015. doi:10.1109/CSITechnol. 2015.7358251

[30] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. Paper presented at the Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, Philadelphia, Pennsylvania, USA. doi:10.1145/237814.237866

[31] Hajiabolhassan, H., & Cheraghi, A. (2010). Bounds for visual cryptography schemes. Discrete Applied Mathematics, 158(6), 659-665. doi:10.1016/j.dam.2009.12.005 Hamming, R. W. (1950). Error detecting and error correcting codes. Bell System Technical Journal, 29(2), 147-160.

[32] Hosnieh, R., Martin von, L., & Christoph, M. (2013). Cryptography in Electronic Mail Theory and Practice of Cryptography Solutions for Secure Information Systems (pp. 406-427). Hershey, PA, USA: IGI Global. doi:10.4018/978-1-4666-4030- 6.ch016

[33] Hou, Y. C., Wei, S. C., & Lin, C. Y. (2014). Random-Grid-Based Visual Cryptography Schemes. IEEE Transactions on Circuits and Systems for Video Technology, 24(5), 733-744. doi:10.1109/TCSVT.2013.2280097

[34] Hu, C. M., & Tzeng, W. G. (2007). Cheating Prevention in Visual Cryptography. IEEE Transactions on Image Processing, 16(1), 36-45. doi:10.1109/TIP.2006.884916Hurley, B., & Hurley, T. (2011). Group ring cryptography. International Journal of Pure and Applied Mathematics, 69(1), 67-86.

[35] Jaya, Malik, S., Aggarwal, A., & Sardana, A. (2011, 11-14 Dec. 2011). Novel authentication system using visual cryptography. Paper presented at the 2011 World Congress on Information and Communication Technologies (WICT). doi:10.1109/WICT.2011.6141416

[36] Jeeva, A., Palanisamy, D. V., & Kanagaram, K. (2012). Comparative analysis of performance efficiency and security measures of some encryption algorithms. International Journal of Engineering Research and Applications (IJERA) ISSN, 2248-9622.

[37] Jie, L., & King, B. (2013, 4-7 Aug. 2013). Smart card fault attacks on elliptic curve cryptography. Paper presented at the IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), 2013. doi:10.1109/MWSCAS.2013.6674882

[38] Joseph, S. K., & Ramesh, R. (2015, 16-19 Dec. 2015). Random grid based visual cryptography using a common share. Paper presented at the 2015 International Conference on Computing and Network Communications (CoCoNet). doi:10.1109/CoCoNet.2015.7411259

[39] Joux, A., & Vitse, V. (2012). Cover and decomposition index calculus on elliptic curves made practical Advances in Cryptology–EUROCRYPT 2012 (pp. 9-26): Springer. doi:10.1007/978-3-642-29011-4_3

[40] Kafri, O., & Keren, E. (1988). Method and apparatus of encryption of optical images: Google Patents.

[41] Kamarulhaili, H. (2010, 7-10 Aug. 2010). Generating Elliptic Curves Modulo p for Cryptography Using Mathematica Software. Paper presented at the Seventh International Conference on Computer Graphics, Imaging and Visualization (CGIV), 2010. doi:10.1109/CGIV.2010.22

[42] Kang, I., Arce, G. R., & Lee, H. K. (2011). Color Extended Visual Cryptography Using Error Diffusion. IEEE Transactions on Image Processing, 20(1), 132-145. doi:10.1109/TIP.2010.2056376

[43] Klein, A. (2008). Attacks on the RC4 stream cipher. Designs, Codes and Cryptography, 48(3), 269-286. doi:10.1007/s10623-008-9206-6

[44] Klein, A., & Wessler, M. (2007). Extended visual cryptography schemes. Information and Computation, 205(5), 716-732. doi:10.1016/j.ic.2006.12.005

[45] Klisowski, M., & Ustimenko, V. (2010, 18-20 Oct. 2010). On the implementation of public keys algorithms based on algebraic graphs over finite commutative rings. Paper presented at the Proceedings of the 2010 International Multiconference on Computer Science and Information Technology (IMCSIT). doi:10.1109/IMCSIT.2010.5679687

[46] Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of computation, 48(177), 203-209.

[47] Kofahi, N. A., Turki, A.-S., & Khalid, A.-Z. (2003, 30-30 Dec. 2003). Performance evaluation of three encryption/decryption algorithms. Paper presented at the IEEE 46th Midwest Symposium on Circuits and Systems, 2003.

[48] Kotorowicz, J., Romanczuk, U., & Ustimenko, V. (2011, 18-21 Sept. 2011). On the implementation of stream ciphers based on a new family of algebraic graphs. Paper presented at the Federated Conference on Computer Science and Information Systems (FedCSIS), 2011.

[49] Krämer, J. (2015). Why Cryptography Should Not Rely on Physical Attack Complexity.Singapore: Springer.

[50] Krebs, M., & Shaheen, A. (2011). Expander Families and Cayley Graphs : A Beginner's Guide. Cary: Oxford University Press.

[51] Leca, C. L., & Rincu, C. I. (2014, 29-31 May 2014). Combining point operations for efficient elliptic curve cryptography scalar multiplication. Paper presented at the 10th

International Conference on Communications (COMM), 2014. doi:10.1109/ICComm.2014.6866676

[52] Liu, F., & Wu, C. (2011). Embedded Extended Visual Cryptography Schemes. IEEE Transactions on Information Forensics and Security, 6(2), 307-322. doi:10.1109/TIFS.2011.2116782

[53] Liu, F., Wu, C., & Lin, X. (2010a). A new definition of the contrast of visual cryptography scheme. Information Processing Letters, 110(7), 241-246. doi:10.1016/j.ipl. 2010.01.003

[54] Liu, F., Wu, C., & Lin, X. (2010b). Step Construction of Visual Cryptography Schemes. IEEE Transactions on Information Forensics and Security, 5(1), 27-38. doi:10.1109/TIFS.2009.2037660

[55] Liu, F., Wu, C. K., & Lin, X. J. (2008). Colour visual cryptography schemes. Information Security, IET, 2(4), 151-165. doi:10.1049/iet-ifs:20080066

[56] Liu, M., Han, L., & Wang, X. (2011). On the equivalent keys in multivariate cryptosystems. Tsinghua Science and Technology, 16(3), 225-232. doi:10.1016/S1007-0214(11)70033-5

[57] Loehr, N. (2014). Advanced Linear Algebra. Bosa Roca: CRC Press.

[58] Lu, S., Manchala, D., & Ostrovsky, R. (2011). Visual cryptography on graphs. J. Comb.

[59] Optim., 21(1), 47-66. doi:10.1007/s10878-009-9241-x

[60] Ma, K., & Wu, K. (2014). Error Detection and Recovery for ECC: A New Approach Against Side-Channel Attacks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 33(4), 627-637. doi:10.1109/TCAD.2013. 2293058

[61] Marin, L., Jara, A., & Skarmeta, A. F. (2012, 4-6 July 2012). Shifting Primes: Optimizing Elliptic Curve Cryptography for Smart Things. Paper presented at the Sixth

International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012. doi:10.1109/IMIS.2012.199

[62] Martin, K. M. (2012). Everyday Cryptography: Fundamental Principles & Applications. New York: Oxford University Press.

[63] Martinez, V. G., & Encinas, L. H. (2013). Implementing ECC with Java Standard Edition

[64] International Journal of Computer Science and Artificial Intelligence, 3(4), 134.

[65] Masadeh, S. R., Aljawarneh, S., Turab, N., & Abuerrub, A. M. (2010, 16-18 Aug. 2010). A comparison of data encryption algorithms with the proposed algorithm: Wireless security. Paper presented at the Sixth International Conference on Networked Computing and Advanced Information Management (NCM), 2010.

[66] Mehta, S., Varadharajan, V., & Nallusamy, R. (2012). Tampering resistant self recoverable watermarking method using error correction codes. International Journal of Information and Computer Security, 5(1), 28-47. doi:doi:10.1504/IJICS.2012.051089

[67] Menezes, A. J., Okamoto, T., & Vanstone, S. A. (1993). Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory, 39(5), 1639-1646.

[68] Miller, V. S. (1985). Use of elliptic curves in cryptography. Paper presented at the Advances in Cryptology—CRYPTO'85 Proceedings.

[69] Modares, H., Moravejosharieh, A., & Salleh, R. (2011, 12-14 Dec. 2011). Wireless Network Security Using Elliptic Curve Cryptography. Paper presented at the First International Conference on Informatics and Computational Intelligence (ICI), 2011. doi:10.1109/ICI.2011.63

[70] Mostaghim, M., & Boostani, R. (2014, 3-4 Sept. 2014). CVC: Chaotic visual cryptography to enhance steganography. Paper presented at the 11th International ISC

Conference on Information Security and Cryptology (ISCISC), 2014. doi:10.1109/ISCISC.2014.6994020

[71] Naor, M., & Shamir, A. (1995). Visual cryptography. In A. De Santis (Ed.), Advances in Cryptology — EUROCRYPT'94 (Vol. 950, pp. 1-12): Springer Berlin Heidelberg. doi:10.1007/BFb0053419

[72] NIST. (2000). FIPS 186-2: Digital Signature Standard (DSS). Gaithersburg MD: National Institute of Standards and Technology.

[73] Ontiveros, B., Soto, I., & Carrasco, R. (2006). Construction of an elliptic curve over finite fields to combine with convolutional code for cryptography. IEEE Proceedings - Circuits, Devices and Systems, 153(4), 299-306. doi:10.1049/ip-cds:20050117

[74] Pal, S. K. (2007, 5-7 March 2007). Fast, Reliable & Secure Digital Communication Using Hadamard Matrices. Paper presented at the International Conference on Computing: Theory and Applications, 2007. ICCTA '07. doi:10.1109/ICCTA. 2007.61

[75] Paszkiewicz, A., Górska, A., Górski, K., Kotulski, Z., Kulesza, K., & Szczepański, J. (2001). Proposals of Graph Based Ciphers, Theory and Implementations. Paper presented at the Proceedings of the Regional Conference on Military Communication and Information Systems. CIS Solutions for an Enlarged NATO, RCMIS.

[76] Pateriya, R. K., & Vasudevan, S. (2011, 3-5 June 2011). Elliptic Curve Cryptography in Constrained Environments: A Review. Paper presented at the International Conference on Communication Systems and Network Technologies (CSNT), 2011. doi:10.1109/CSNT.2011.32

[77] Petit, C., & Quisquater, J.-J. (2012). On polynomial systems arising from a Weil descent

[78] Advances in Cryptology–ASIACRYPT 2012 (pp. 451-466): Springer.

[79] Polak, M., Romańczuk, U., Ustimenko, V., & Wróblewska, A. (2013). On the applications of Extremal Graph Theory to Coding Theory and Cryptography. Electronic Notes in Discrete Mathematics, 43, 329-342.

[80] Polak, M., & Ustimenko, V. (2013, 8-11 Sept. 2013). Examples of Ramanujan and expander graphs for practical applications. Paper presented at the Federated Conference on Computer Science and Information Systems (FedCSIS), 2013.

[81] Prachi, Dewan, S., & Pratibha. (2015, 21-22 Feb. 2015). Comparative Study of Security Protocols to Enhance Security. Paper presented at the Fifth International Conference on Advanced Computing & Communication Technologies (ACCT), 2015. doi:10.1109/ACCT.2015.34

[82] Priyadarsini, P. L. K. (2015). A Survey on some Applications of Graph Theory in Cryptography. Journal of Discrete Mathematical Sciences and Cryptography, 18(3), 209-217. doi:10.1080/09720529.2013.878819

[83] Priyadarsini, P. L. K., & Ayyagari, R. (2013, 22-25 Aug. 2013). Ciphers based on special graphs. Paper presented at the International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013 doi:10.1109/ICACCI.2013.6637215

[84] Qu, Y., & Hu, Z. (2010, 21-24 May 2010). Research and design of elliptic curve cryptography. Paper presented at the 2nd International Conference on Future Computer and Communication (ICFCC), 2010. doi:10.1109/ICFCC.2010. 5497370

[85] Riaz, F., & Ali, K. M. (2011, 26-28 July 2011). Applications of Graph Theory in Computer Science. Paper presented at the Third International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), 2011. doi:10.1109/CICSyN.2011.40

[86] Rivest, R. L., & Schuldt, J. C. (2014). Spritz-a spongy RC4-like stream cipher and hash function. Proceedings of the Charles River Crypto Day, Palo Alto, CA, USA, 24.

[87] Ross, A., & Othman, A. (2011). Visual Cryptography for Biometric Privacy. IEEE Transactions on Information Forensics and Security, 6(1), 70-81. doi:10.1109/TIFS.2010.2097252

[88] Schneier, B. (1993). Description of a new variable-length key, 64-bit block cipher (Blowfish). Paper presented at the International Workshop on Fast Software Encryption.

[89] Setiadi, I., Kistijantoro, A. I., & Miyaji, A. (2015, 19-22 Aug. 2015). Elliptic curve cryptography: Algorithms and implementation analysis over coordinate systems. Paper presented at the 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), 2015. doi:10.1109/ICAICTA. 2015.7335349

[90] Shakespeare, W., & Ackroyd, P. (2006). The Complete Works of William Shakespeare.Glasgow, UK: Harper Collins.

[91] Shankar, P. (1997). Error correcting codes. Resonance, 2(3), 33-47. doi:10.1007/bf02838967

[92] Sharma, M., Garg, R. B., & Dwivedi, S. (2014, 8-10 Oct. 2014). Comparative analysis of NPN algorithm & DES Algorithm. Paper presented at the 3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), 2014. doi:10.1109/ICRITO.2014.7014688

[93] Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. Paper presented at the 35th Annual Symposium on Foundations of Computer Science, 1994.

[94] Shyu, S. J., Huang, S.-Y., Lee, Y.-K., Wang, R.-Z., & Chen, K. (2007). Sharing multiple secrets in visual cryptography. Pattern Recognition, 40(12), 3633-3651. doi:10.1016/j.patcog.2007.03.012

[95] Silverman, J. H., & Suzuki, J. (1998). Elliptic Curve Discrete Logarithms and the Index Calculus. In K. Ohta & D. Pei (Eds.), Advances in Cryptology —

ASIACRYPT'98: International Conference on the Theory and Application of Cryptology and Information Security Beijing, China, October 18–22, 1998 Proceedings (pp. 110- 125). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/3-540-49649- 1_10

[96] Singh, L. D., & Debbarma, T. (2014, 8-10 May 2014). A new approach to Elliptic Curve Cryptography. Paper presented at the International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2014. doi:10.1109/CCAA.2015.7148511

[97] Singhal, N., & Raina, J. (2011). Comparative analysis of AES and RC4 algorithms for better utilization. International Journal of Computer Trends and Technology, 2(6), 177-181.

[98] Stallings, W. (2014). Cryptography and Network Security (6 ed.). New Jersey, USA: Pearson Education Inc.

[99] Sutter, G. D., Deschamps, J. P., & Imana, J. L. (2013). Efficient Elliptic Curve Point Multiplication Using Digit-Serial Binary Field Operations. IEEE Transactions on Industrial Electronics, 60(1), 217-225. doi:10.1109/TIE.2012.2186104

[100] Targhetta, A. D., Owen, D. E., Israel, F. L., & Gratz, P. V. (2015, 18-21 Oct. 2015). Energy-efficient implementations of GF (p) and GF(2m) elliptic curve cryptography. Paper presented at the 33rd IEEE International Conference on Computer Design (ICCD), 2015. doi:10.1109/ICCD.2015.7357184

[101] Tawalbeh, L., Mowafi, M., & Aljoby, W. (2013). Use of elliptic curve cryptography for multimedia encryption. IET Information Security, 7(2), 67-74. doi:10.1049/iet-ifs.2012.0147

[102] Thakur, J., & Kumar, N. (2011). DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis. International journal of emerging technology and advanced engineering, 1(2), 6-12.

[103] Thiranant, N., Kang, Y. J., Kim, T., Jang, W., Park, S., & Lee, H. (2014, 19-21 Dec. 2014). A Design of Elliptic Curve Cryptography-Based Authentication Using QR Code. Paper presented at the IEEE 17th International Conference on Computational Science and Engineering (CSE), 2014. doi:10.1109/CSE.2014. 135

[104] Ustimenko, V. (2007). On Graph-Based Cryptography and Symbolic Computations.Serdica Journal of Computing, 1(2), 131-156.

[105] Ustimenko, V. (2014, 7-10 Sept. 2014). On multivariate cryptosystems based on maps with logarithmically invertible decomposition corresponding to walk on graph. Paper presented at the Federated Conference on Computer Science and Information Systems (FedCSIS), 2014. doi:10.15439/2014F269

[106] Ustimenko, V., & Romańczuk, U. (2013). On extremal graph theory, explicit algebraic constructions of extremal graphs and corresponding Turing encryption machines Artificial Intelligence, Evolutionary Computing and Metaheuristics (pp. 257- 285). Heidelberg, Germany: Springer. doi:10.1007/978-3-642-29694-9_11

[107] Vigila, S., & Muneeswaran, K. (2009, 13-15 Dec. 2009). Implementation of text based cryptosystem using Elliptic Curve Cryptography. Paper presented at the First International Conference on Advanced Computing, 2009. ICAC 2009. doi:10.1109/ICADVC.2009.5378025

[108] Wang, R. Z., & Hsu, S. F. (2011). Tagged Visual Cryptography. IEEE Signal Processing Letters, 18(11), 627-630. doi:10.1109/LSP.2011.2166543

[109] Wang, X., Pei, Q., & Li, H. (2014). A Lossless Tagged Visual Cryptography Scheme.

[110] IEEE Signal Processing Letters, 21(7), 853-856. doi:10.1109/LSP.2014.2317706 Wu, H. (2008). The stream cipher HC-128 New stream cipher designs (pp. 39-47): Springer. doi:10.1007/978-3-540-68351-3_4

[111] Wu, X., & Sun, W. (2013). Generalized Random Grid and Its Applications in Visual Cryptography. IEEE Transactions on Information Forensics and Security, 8(9), 1541-1553. doi:10.1109/TIFS.2013.2274955

[112] Xia, L. (2012, 24-27 June 2012). The application of Elliptic Curve Cryptography in Electronic Commerce. Paper presented at the IEEE Symposium on Electrical & Electronics Engineering (EEESYM), 2012. doi:10.1109/EEESym.2012.6258715

[113] Yan, S. Y. (2008). Cryptanalytic attacks on RSA. New York, USA: Springer US. doi:10.1007/978-0-387-48742-7

[114] Ye, L., & Liu, F. (2011, 24-26 Dec. 2011). Overview of scalar multiplication in elliptic curve cryptography. Paper presented at the International Conference on Computer Science and Network Technology (ICCSNT), 2011. doi:10.1109/ICCSNT.2011. 6182515

[115] Zhi, Z., Arce, G. R., & Di Crescenzo, G. (2006). Halftone visual cryptography. IEEE Transactions on Image Processing, 15(8), 2441-2453. doi:10.1109/TIP.2006. 875249