Proceedings of the Second

University Journal of Research and Innovation

August, 2020

Organized by

University of Computer Studies (Pakokku)

Proceedings of

The Second University Journal of Research and Innovation 2020

Augest, 2020

Organized by

University of Computer Studies (Pakokku)

Department of Higher Education,

Ministry of Education, Myanmar

University Journal of Research and Innovation

Volume 2, Issue 1 2020

University Journal of Research and Innovation 2020

Volume 2, Issue 1, 2020

This journal and individual paper published at www.ucspkku.edu.mm.

Copies:110

All research papers in this journal have undergone rigorous peer-reviewed which is published annually. Full papers submitted for publication are refereed by the Associate Editorial Board through an anonymous referee process.

The authors of the paper bear the responsibility for their content.

Papers presented at the Second University Journal of Research and Innovation (UJRI), University of Computer Studies (Pakokku), August 2020.

UJRI 2020 Editorial Board

UJRI 2020 Editorial Board

Proceedings of

The Second University Journal of

Information and Computing Science 2020

Augest, 2020

Contents

Database Management System & Information Retrieval

Digital Signal Processing

Embedded System

Image Processing

Network & Security

Software Engineering and Web Engineering

Comparison of Data Science Methods for Cyber-security

Aye Pyae Sone
University of Computer Studies, Hpa-An
ayepyaesone@ucshpaan.edu.mm

Kyaw Soe Moe
University of Computer Studies, Pyay
kyawsoemoe@ucspyay.edu.mm

Ohnmar Aung
Computer University, (Mandalay)
mamalay2009@gmail.com

Abstract

The main purpose of this paper is to analyze data science methods and determine which is effective for cyber security problem. Firstly, we need to know about data science which is to extract information and knowledge from large amount of data by using many evolving techniques. Analytic techniques can be applied on cyber-security solutions when data sciences collect and store big data. To control the consequences of threats, machine learning and big data can be applied with the use of analytic models. Modern cyber-security solutions are based on huge data because the more data creates the more accurate analysis. Solving problems and answering questions through data analysis is required to protect users against cyber crimes in data science project. In this paper, we provide the comparison of data science methods KDD process, CRISP-DM and FMDS with their strengths and weakness.

Keywords: CRISP-DM, Cyber-security, Data Science, FMDS, KDD

1. Introduction

Cyber-security solutions are based on pattern identification by comparing a malware and a new threat. It could predict the potential attacks by using data analysis techniques. Data science methods extract knowledge and insights from various structured, unstructured and semi-structured data related to big data and machine learning. Big data is a term of data sets which are too big data processing and require new methods and technologies. By using a large amount of data, it can be the more precise analysis result in solving the security problem. Data analysis techniques involve information protection, intruder detection and business recovery. In this paper, we proposed the methods of data science which are covered for cyber-security and will compare them with their strengths and weakness. This paper is organized as follows. Section-2 provides the overview of data science with their two major concepts. Section-3 describes information about the three methods of data science in detail. A comparison discussion of these methods with their strengths and weakness are delivered n Section-4. In the end, we recommend an effective method that might cover all possible requirements of cyber-security solutions and conclude this paper.

2. Overview of Data Science

Data mining is a concept knowledge discovery in database [5]. Data analysis technique, machine learning and their related methods are applied in data science process to understand and fix data. It is a set of basic concepts to draw out knowledge and information from data. And then, it helps better identify threats, stop intrusions and attacks in cyber-security. The main benefit of using data science for cyber-security process is that big data can be used to detect threats. Various samples of data can be used for deep learning and training purposes in which malware and spam are properly detected. The decision-making process can be concluded by providing data-driven predictions. There are two major concepts of knowledge data discovery in database for cyber-security. These concepts are data driven decision-making, and user data discovery.

2.1. Concepts of Data Science

In ***data-driving decision-making process*** *(DDD*), data processing is needed to support data science tasks. DDD describes the two types of decisions for data science in cyber-security: 1) decisions which are based on data discovery and 2) decisions which are based on frequent decision-making process [2]. Data discovery is the process of transforming complicated data collections into information that users can understand and manage. Frequent decision-making is the process of depending on information evaluation and data analysis. In contrast, DDD means gathering data based on measurable goals, analyzing patterns and facts from these insights and using these facts to develop strategies and activities that provide cyber-security solutions. The ***user data discovery (UDD)*** is the process of predicting user profile from previous information and past details [4]. User profiling can be defined as data collection about an interested domain. The strategy of user profiling is either knowledge-based or behavior-based. The knowledge- based strategy uses a statistical model to classify users based on dynamic attributes. The behavior-based strategy takes the user's behaviors and actions as a model to find patterns by using machine learning technique. By feeding present and past data into machine learning algorithm, the system can exactly detect potential problems. Data scientists can apply their knowledge into the cyber-security field to help protect against attacks and identify suspected manner.

3. Methodology of Data Science

To obtain the analysis result, data sources will pass through several steps. Firstly, raw data is prepared for analysis by using various data preparation methods. Then, the prepared data would be analyzed to abstract and visualize data and develop models using various data analytic techniques. There is typically a practice for predictive model creation, pattern recognition and underlying discovery problems through data analysis. Data science methodology provides the ideal approach to solve problems through a prescribed sequence of steps. Data science methodology covers five main aspects of data science projects, these aspects are:

1. Transforming from problem to approach
2. Transforming from requirements to collection
3. Transforming from understanding to preparation
4. Transforming from modeling to evaluation
5. Transforming from deployment to feedback.

3.1. KDD process

Knowledge discovery in databases, (KDD) which is the process of discovering knowledge in data and focus on high-level applications of specific data mining techniques [3]. The main purpose of the KDD process is to draw out information from data in the context of huge databases. In developing KDD process, it needs to consider the application domain perception.

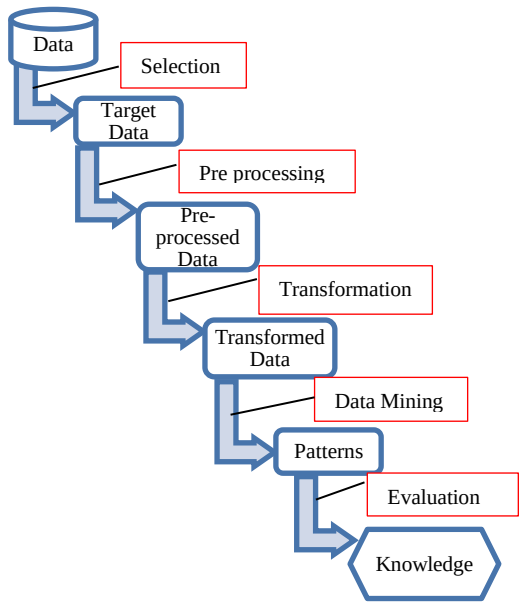

Figure 1. KDD Process

There are five steps in KDD process:

1. ***Selection***: In this step, creating target data and focus on data samples to select the appropriate data is included.
2. ***Pre-processing***: This phase involves consistent data by cleaning selected data.
3. ***Transformation***: This stage transforms data by reducing dimensional on transformation method.
4. ***Data mining***: This phase recognizes interested pattern to be specific form by using data mining techniques.
5. ***Interpretation/evaluation***: This final phase evaluates obtained pattern and ensures it is ready to use.

3.2. CRISP-DM process

CRISP-DM means Cross-Industry standard process for data mining. It is an open standard process model which describes useful ways for data scientists. The analytic model, CRISP-DM is the most widely used today. Although it is a very useful tool, need to update detail. It also supports very useful instructions for the most developed data science projects.

The six phases of CRISP-DM are the following:

1. ***Business understanding:*** This phase is to understand the needs of cyber-security problem and then transforms this perception into description of data mining.
2. ***Data understanding:*** This phase begins with initial data collection, verify data quality and identify interesting sub-groups to improve hidden information. It typically makes a set of data samples.
3. ***Data preparation:*** To remove duplicate data, address missed values, construct required data and integrate all data to format.
4. ***Modeling:*** This phase selects modeling techniques, generate test design, build model and identify specific data to get better results.
5. ***Evaluation:*** Accessing any other data mining results, approved models, review process and determine next steps. The final result is choosing the sufficient model.
6. ***Deployment:*** It will take the evaluation result and determine a strategy for their deployment. It involves all the data preparation and required stages which are needed to set raw data to get the final outcome during model construction.

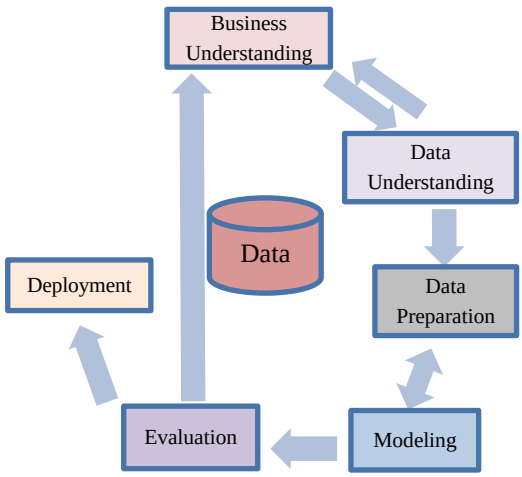

Figure 2. CRISP-DM Life Cycle

3.3. FMDS process

Data scientists need FMDS process because it provides big data set, text and image analytic, deep learning, artificial intelligence and language processing [6]. Foundational Methodology for data science (FMDS) consists of the following steps:

1. ***Business understanding:*** This stage defines problem description, requirements of business perspective and objectives of project. First of all, it tries to understand the resolution of business problem (cyber-security problem) and then provides domain expertise to achieve a project.
2. ***Analytic approach:*** After describing a cyber-security problem clearly, data

scientists can define the analytic approach by identifying the context of statistical and machine learning techniques to achieve the desired outcome.

3. ***Data requirements:*** Choice of analytic approach determines the data requirements, particular data context, formats and representation which are instructed by cyber-security knowledge.
4. ***Data collection:*** This phase is to collect the structured, unstructured and semi-structured data resources which are related to cyber-security problem. If there are data gaps in collection, the data requirements need to be revised and collects more data.
5. ***Data understanding:*** After revisiting the previous step, data gathering might be necessary to close gaps in data understanding. In this phase, descriptive statics and visualization techniques are useful to understand data content, determine data quality and transform initial percept into the data.
6. ***Data preparation:*** This stage employs all activities to build data set that will be used in the modeling stage. This phase comprises of data cleaning (removing invalid values, eliminating duplicate data, formatting data), merging data from multiple data sources (files, tables, platforms) and transforming data into more applicable variables. Feature engineering and text analytic are used in deriving new structured variables to improve model accuracy. Although it is time-consuming step in data science project, high performance and parallel computing system could reduce needed time and prepare data as fast as from huge data sets.
7. ***Modeling:*** The modeling stage insights on developing predict, or descriptive models according to the analytic approach previously defined. Data scientists test a training set (historical data) to build the predictive model. It is highly repeated step as organizations gain intermediate insights. It leads to refine in data preparation and model specification.
8. ***Evaluation:*** To ensure properly and completely addresses the cyber-security problem, data scientists evaluate the model and attempt to understand its quality. Several diagnostic measures are considered to compute evaluation. And other outputs which include tables and graphs are evaluated by using a testing set. However, it is independent the set of training, it follows the same probability distribution and has a precise outcome. The testing set is used to evaluate the model because it can be refined as need as.

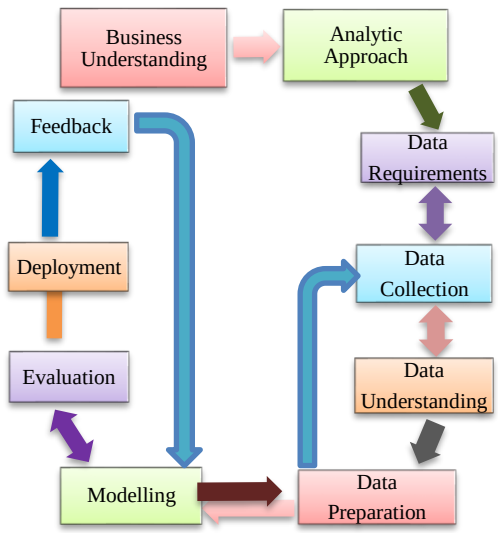

Figure 3. FMDS

9. ***Deployment:*** After approving the developed model in the evaluation phase, it is deployed into a comparable test environment. It produces a simple report with proper suggestions. It also provides managing custom application, embedding the model in a complicated workflow. Deploying a model into a related field (cyber-security problem) involves skills, more applicable form, and technologies within an enterprise.

10. ***Feedback:*** Finally, the results from implemented model are collected in this phase. The function of it is to analyze feedback, performance and capability of deployment phase. It can define the model to improve its accuracy and usefulness by analyzing the feedback.

4. Analysis Process Model

There are various analytic algorithms such as Hadoop and Spark to extract big data sets. In order to predict data, machine learning techniques are used for pattern identification. The historical data are developed through data collection, and then combines these data by using machine learning algorithm. In this paper, we use Hadoop, one of the most suitable tools for analyzing big data. It is a Java based framework and open source software platform for storing and processing big data. Hadoop has been inspired on Google File System (GFS) released by Google in 2003 [7]. It can provide fast access on both structured or unstructured data. Hadoop ecosystem project includes HDFS, Hive, MapReduce, Pig, etc. [9]. Hadoop distributed file system (HDFS) and MapReduce programming model are used in this paper for faster data retrieve from its nodes.

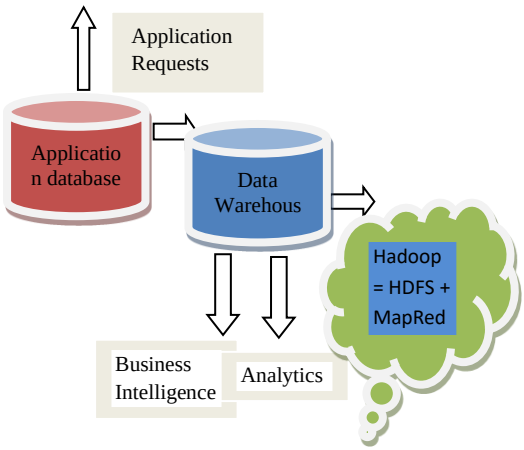

Figure 4. Hadoop Architecture

HDFS is a master / slave architecture, and it breaks data into small blocks. Apache Map Reduce which is software for distributed processing of large data sets. It splits the received data into different datasets which are made to parallel process. Naïve Bayes classification method of machine learning is used for classification.

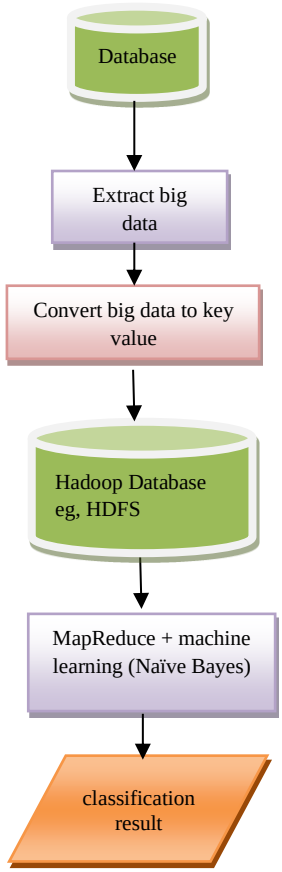

Figure 5. Hadoop MapReduce Data Flow Diagram

4.1. A Comparative Study

All data science methods we discussed above consists of four main iterative stages, which are problem definition/formulation, data gathering, data modeling, and data production KDD.

Comparing the KDD process with the CRISP-DM process, we observed that CRISP-DM involves business understanding and

deployment stage but KDD does not cover these two phases.

- The business understanding phase emphasizes to understand the business objectives and changes this perception into data mining knowledge.
- The deployment is also important because users need to apply the real form of data in business environment. This phase can be identified by presenting this knowledge into cyber-security problem.

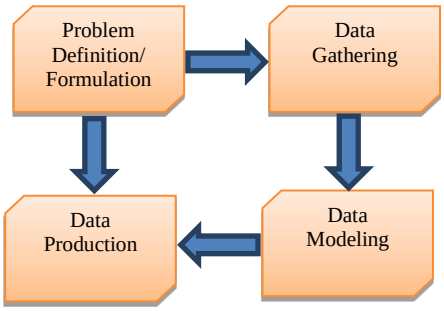

Figure 6. General Data Science Methodology

Concerning the remaining phases, we observed that:

- The data understanding phase in CRISP-DM is similar to the combination of selection and pre-processing phase in KDD.
- The data preparation phase in CRISP-DM can be identified with transformation in KDD.
- The modeling phase in CRISP-DM is similar to data mining in KDD because these two phases include extracting patterns on building models.
- The evaluation phase of CRISP-DM can be identified with Interpretation and evaluation phase in KDD.

Comparing the CRISP-DM process with the FMDS process, this paper suggests that the important stages.
Analytic approach and feedback have been missed in CRISP-DM.

- The analytic approach is needed to choose the suitable machine learning techniques before gathering data gaps. It is a very useful way to determine the appropriate data collection strategy and data resources.
- The feedback phase is applicable to optimize the system to get better results for high performance functionality and effective case.

Table 1. Summary of Data Science Methods

KDD	CRISP-DM	FMDS
-	Business Understanding	Business Understanding
	-	Analytic Approach
	-	Data Requirements
Selection	Data Understanding	Data Collection
Pre processing		Data Understanding
Transformation	Data Preparation	Data Preparation
Data mining	Modeling	Modeling
Interpretation/ Evaluation	Evaluation	Evaluation
-	Deployment	Deployment
-	-	Feedback

Data requirements stage which is concerned with analytic approach and provides required data contents are missed in both KDD process and CRISP-DM process.

This paper provides the comparative study on three data science methods. The KDD process is just knowledge discovering process, it is not completely optimal. Most of the analytic managers use CRISP-DM process because they recognize the need for a repeatable approach, but there are some problems with how CRISP-DM is generally applied. There are four problems of CRISP-DM which are a lack of clarity, mindless re-work, blind hand offs to IT and a failure to iterate.
The strengths of FMDS should be considered for modern cyber-security projects [1].

- Automatic feedback gathering and model assessment, refinement and redeployment steps are used to speed up

and refresh the model process helping to get better results.

- The flow of methodology illustrates the iterative feature of problem-solving process. This can handle data mixing from several sources. FMDS provides this feature in the data preparation and data understanding phase.
- FMDS is platform independent and a useful tool. The entire analytic algorithm should be run not just for sample modeler to ensure the evaluated results are suitable in all situations. Cyber-security project can be more efficient and reliable by testing these models.
- The evaluation, deployment and feedback phases could be better than the simple evaluation phase in both KDD and CRISP-DM. Feedback phase which is included in FMDS might create innovative data science questions to optimize the cyber-security project and make new functionalities for it.
- There is no need to spend excessive time on data preparation or modeling for the accuracy and reliability of cyber-security problem as FMDS is more general and independent of any platform tool.

5. Conclusion

In cyber-security data science project, there are four general steps. The first step is problem definition which is to formulate a security challenge. The second step is necessary to gather required information according to problem definition and the third step uses the collected information to provide a resolution for the defined problem. The final step is a production step which deploys the appropriate modules and a system to run the whole process automatically. In this paper, we compare the three data science methods for cyber-security problem. Considering the presented analysis, we conclude that the FMDS process covers all beneficial attributes of cyber-security project. It is also platform and tool independent. It could be recommended in any cyber-security project because it is designed in detail with clearly divided steps.

References

[1] Farhad Foroughi and Peter Luksch, “Data Science Methodology for Cybersecurity project” https://www.researchgate.net>publication

[2] F. Provost and T. Fawcett, "Data science and its relationship to big data and data-driven decision making," Big Data, vol. 1, no. 1, pp. 51-59, 2013.

[3] U.M. Fayyad, "Data mining and knowledge discovery: Making sense out of data," IEEE Expert: Intelligent Systems and Their Applications, vol. 11, no. 5, pp. 20-25, 1996.

[4] S.Kanoje, S.Girase, and D.Mukhopadhyay, "User profiling trends, techniques and applications," arXiv preprint arXiv:1503.07474, 2015.

[5] U. M. Fayyd, G. P. Shapiro, and P. Smyth, "From data mining to knowledge discovery: an overview," 1996.

[6] http://bigdatauniversity.com/bdu-wp/bdu-course/ data-science-methodology from IBM

[7] Introduction To Hadoop –NYOUG” https://nyoug.org>SIG>DataWarehousing

[8] https://hadoop.apache.org/core/docs/current/hdfs design.html

[9] https://wiki.apache.org/hadoop/FAQ

A Comparative Study of Password Strength Using Password Meter, Kaspersky and NordPass

Lai Yi Aung, San San Nwel, Khaing Khaing Soe, Mya Mya Htay
University of Computer Studies (Pakokku)
laiyiaung@ucspkku.edu.mm, sansannwel@ucspkku.edu.mm,
khaingkhaingsoe@ucspkku.edu.mm

Abstract

Unauthorized access is a potentially major problem for anyone who uses a computer or high-tech devices such as smartphones or tablets. A strong password provides essential protection from financial fraud and identity theft. In the proposed system, we analyze the strength of password with different lengths (8, 9, 10, 11, and 12 characters) and different character combinations (only small or capital, small and capital, only numeric, only special characters, small/capital and numeric, small and capital and numeric, and all types of character). We also show scores of strength and five levels of complexity for all types of password with different lengths and character combinations. This system use much password checkers such as Password Meter, Kaspersky and NordPass. Finally, this system reveals the comparison result for scores of password strength resulted by these three-password checkers.

Keywords: Unauthorized access complex password, password strength, password length, characters combination.

1. Introduction

Passwords have been an essential user authentication method of years despite the availability of stronger authentication mechanisms [1]. However, there is one major dilemma associated with password-based authentication. That is, a password must be easy about the owner to remember but it must be hard for others to guess. Naturally, many users prefer memorability to security, choosing weak passwords. According to a survey conducted by TeleSign in 2015 [2], 3 out of 4 users chose weak passwords, and 40 percent of them experienced an issue with their account security in the past year. Passwords still are even though much has been said about their weaknesses, we have used the passwords in accounts such as face book, banks, viber, etc. Because, the passwords have an inherent trade-off between usability and security. Nowadays, everybody has at least one password. So, people should know the password security. We have conducted the study of password-strength meters, finding that meters did affect user behavior and security [7]. In this paper, the scores of the password strength based on the different lengths and different character combinations by using three password checkers such as Password Meter, Kaspersky and NordPass are compared.

2. Password Strength Checker

Password strength checkers is becoming quite popular as the fight for more secure online profiles gets more important. There are several types of password strength checker to check password strength. In this system, the three best password security checkers is implemented [5,6].

2.1. NordPass Password Strength Checker

The NordPass Password Checker is our first option and was built by the same team that built NordVPN, one of the most popular commercial

VPN services on the market this tool is great because it not only checks the strength of the password you input, it also checks if the password has been exposed to any data breaches[5].

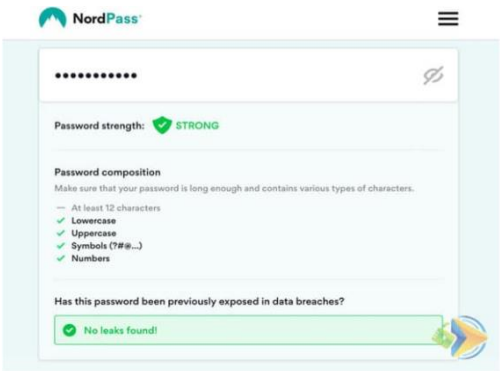

Figure1. NordPass Password Strength Checker

2.2. Kaspersky Password Strength Meter

The Kaspersky password checker is another popular tool built by one of the bigger names in computer security and anti-virus. There's no feedback on the quality of the password other than a red/yellow/green meter and a "time to crack". The upside here is that user knows Kaspersky is going to be a secure place to check user's password. The downside is that they don't give user much here [6].

Figure 2. Kaspersky Password Strength Meter

2.3. The Password Meter

A password strength meter is a Graphical User Interface (GUI) element that is displayed during password creation and offers visual feedback on the strength of the inputted password. Traditional strength meters have been the focus on previous research, regarding both their correctness and their effectiveness. As there is not yet a strict specification, different password meter implementations use different algorithms to measure a password's strength or guess ability. Many passwords meter guide users toward, but do not strictly require, complex passwords. This is why we propose using password strength meters, which can additionally base their complexity of defined of the score [3].

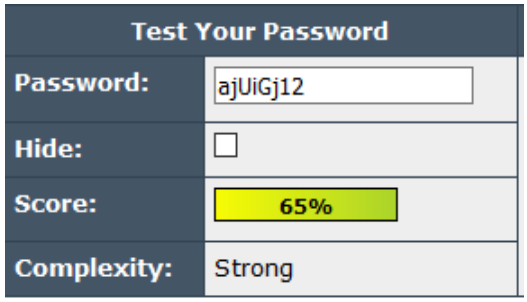

Figure 3. Password Meter

3. System Design

In our proposed system, user firstly enters user's password. Then, password strength of user password is estimated according to the length and different character combination. And then, the system displays the score of password strength as comparison result by much password checkers such as Password Meter, Kaspersky and NordPass. Finally, the system shows the five complexity levels based on score of password strength.

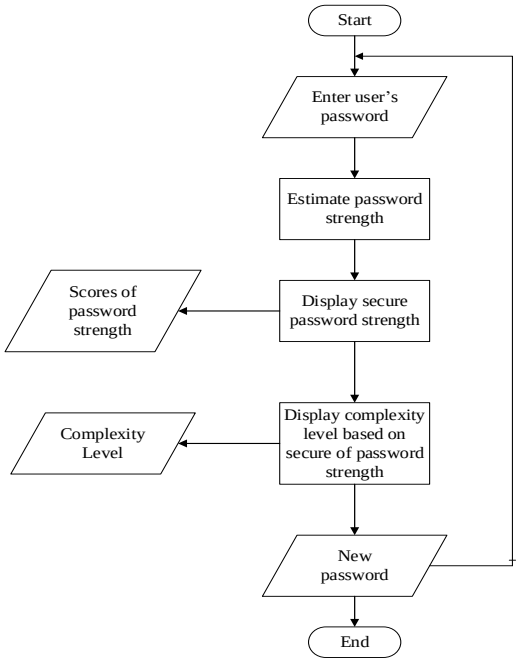

Figure 4. System design for password strength

4. Calculation Result

The score of strength for user's password is calculated by password checker. The calculation for a sample password is shown in following.

Sample input password = Art47021

1. Count the small, capital and digits in the given password.

2. Get the character count and calculate bonus.

Addition Bonus = length * total no: of characters
= 8*4=32;

Addition Bonus 1 = (length - no: of small characters) *2
= (8-2) * 2=12;

Addition Bonus 2 = (length - no: of capital characters) *2
= (8-1) * 2=14;

Addition Bonus 3 = no: of digits*4
= 5*4=20;

Addition Bonus 4 = no: of middle*2
= 4*2=8;

Addition Bonus 4 = no: of requirement*2
= 4*2=8;

Total Addition Bonus=94;

Deduction Bonus 1 = no: Consecutive capital*2
= 0*2=0;

Deduction Bonus 2 = no: Consecutive small*2
= 1*2=2;

Deduction Bonus 3 = no: Consecutive digit *2
= 4*2=8;

Total Deduction Bonus=10;

3. Calculate the total addition bonus and deduction bonus.

4. Get the scores by subtraction total deduction bonus from total addition bonus.

Scores = Total Addition Bonus-Total Deduction Bonus
= 94-10=84;(Very Strong)

5. Experimental Result

In the experiment, we tested the seven types of password (only small or capital, small and capital, only numeric, only special characters, small/capital and numeric, small and capital and numeric, and all types of character) with different password lengths (8, 9, 10, 11 and 12 characters).We also measured scores of strength and complexities of all types of password with different lengths and character combinations using three types of password checker such as Password Meter, Kaspersky and NordPass. The tested results are as shown in Figure 5.

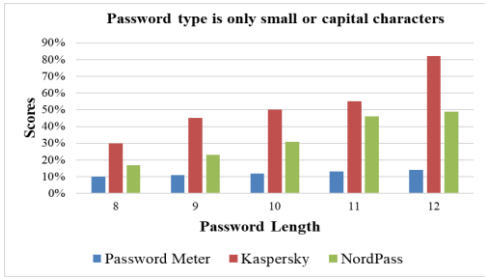

Figure 5 (a). Password type is only small or capital characters

In Figure 5 (a), user password is tested by Password Meter, Kaspersky and NordPass password checkers. The scores of password length **8** are **10%**, **30%** and **17%** respectively. For password length **12**, **14%**, **82%** and **49%** respectively. As shown in Figure 5 (a), Kaspersky is more scores than Password Meter and NordPass.

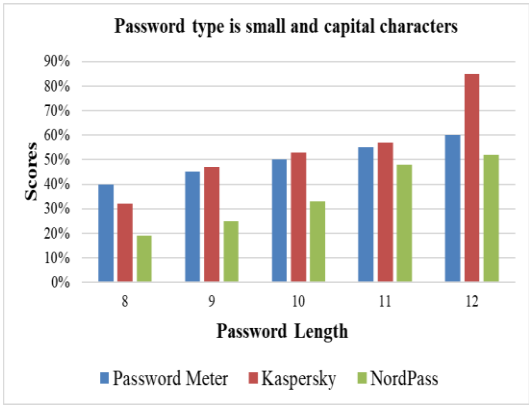

Figure 5 (b). Password type is small and capital characters

According to Figure 5 (b), the scores of password length **8** are **40%, 32%** and **19%** tested by Password Meter, Kaspersky and NordPass respectively. And password length **12** of scores is **60%, 85%** and **52%.** The scores of Kaspersky are more than Password Meter and NordPass except password length **8**.

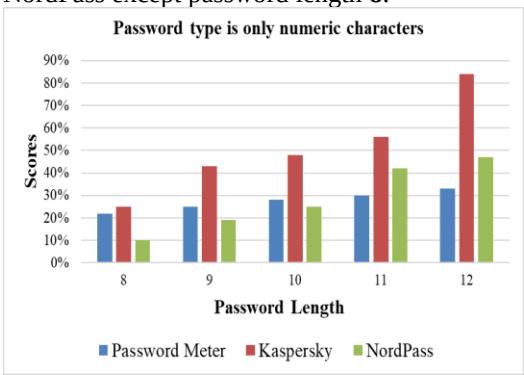

Figure 5 (c). Password type is only numeric characters

In Figure 5 (c), password length **8** the scores of strength are **22%, 25%** and **10%** tested by Password Meter, Kaspersky and NordPass respectively. For password length **12** of scores is **33%, 84%** and **47%.** In Figure 5 (c), Kaspersky is more scores than Password Meter and NordPass.

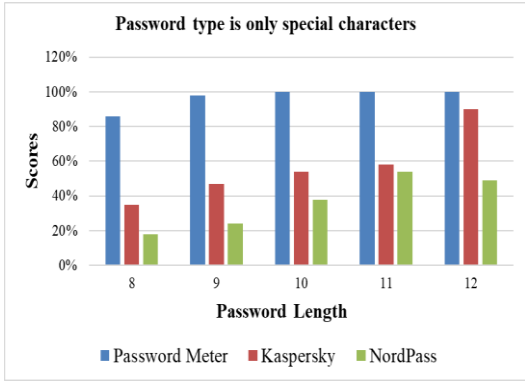

Figure 5 (d). Password type is only special characters

As shown in Figure 5 (d), the scores of password strength for password length **8** are **86%, 35%** and **18%** tested by Password Meter, Kaspersky and NordPass respectively. The scores of password length **12** are **100%, 90%** and **49%**. But, this type of password is less of usage in our environment. In Figure 5 (d), Password Meter is more scores than Kaspersky and NordPass.

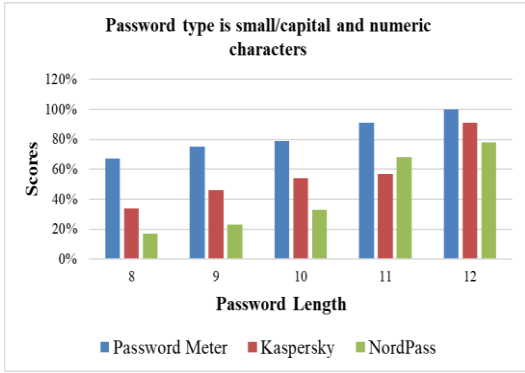

Figure 5 (e). Password type is small/capital and numeric characters

In Figure 5 (e), the scores of password length **8** are **41%, 33%** and **17%** by using Password Meter, Kaspersky and NordPass respectively. And then, the scores of password length **12** are **73%, 86%** and **48%**. According to Figure 5 (e), Password Meter is more scores than Kaspersky and NordPass.

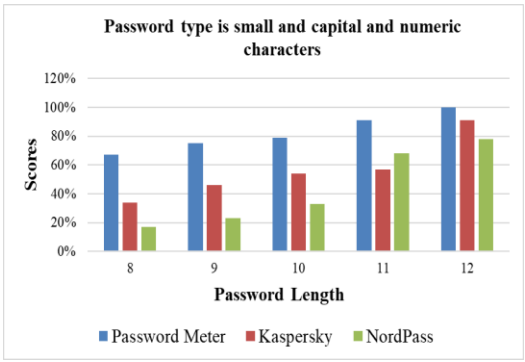

Figure 5 (f). Password type is small and capital and numeric characters

According to in Figure 5 (f), the password length **8**, the scores of password strength are **67%**, **34%** and **17%** by tested with Password Meter, Kaspersky and NordPass respectively. For password length **12,** the length of scores is **100%, 91%** and **78%**. As shown in Figure 5 (f), Password Meter is more scores than Kaspersky and NordPass. This type of password is used in our education site.

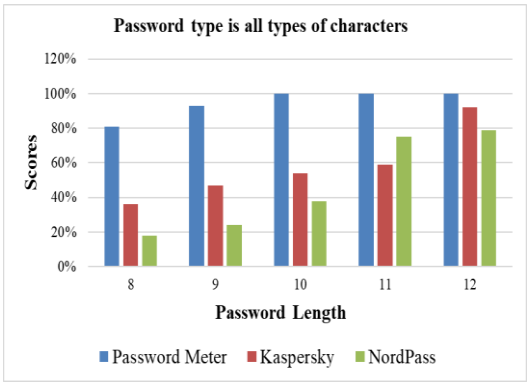

Figure 5 (g) Password type is all types of characters

In Figure 5 (g), password length **8** the score of password strength is **81%, 36%** and **18%** by Password Meter, Kaspersky and NordPass respectively. For password length **12**, the scores are **100%, 92%** and **79%.** According to the experimental result, the score of Password Meter is more than Kaspersky and NordPass but the scores by NordPass is less except for password length **11**.

5.1. Complexities of Password Strength

The scores resulted by much password strength checkers can be divided into complexities range. There are five levels of complexity such as very weak, weak, good, strong and very strong. These levels of complexity show in the following Table 1.

Table 1. Five levels of complexity

Range	Complexities				
	Very Weak	Weak	Good	Strong	Very Strong
Score < 20%	√	–	–	–	–
Score < 40%	–	√	–	–	–
Score < 60%	–	–	√	–	–
Score < 80%	–	–	–	√	–
Score < = 100%	–	–	–	–	√

6. Conclusion

In this work, we proposed a novel way to measure the strength of user-selected passwords. A high enough security level can be reached by the cheapest and simplest method, by applying passwords. According to the experimental results, the proposed system reveals that Password Meter is more scores of password strength than the other password checkers such as Kaspersky and NordPass. But, Kaspersky is more scores of password strength for the password types (only small or capital and only numeric). And then, NordPass is more scoring of strength in the password types which combined small and capital and numeric and all types of

character for password length 11 and 12. The network security is becoming more significant as the volume of data being exchanged with net increases day by day [4]. Therefore, the proposed system can be applied to secure computers in the traditional sense and also to control access to mobile phones, homes, ATMs (automatic teller machines) and many more to enhance security.

References

[1] M. Zviran and W. J. Haga, "Password security: an empirical study," Journal of Management Information Systems, pp. 161–185, 1999.

[2] A. Adams and M. A. Sasse, "Users are not the enemy," *Commun. ACM*,vol. 42, no. 12, pp. 40–46, December 1999.

[3] S. Schechter, C. Herley, and M. Mitzenmacher. Popularity is everything: a new approach to protecting passwords from statistical-guessing attacks. In Proceedings of the 5th USENIX conference on Hot topics in security, pages 1–8. USENIX Association, 2010.

[4] Dell'Amico, M. Michiardi, P.; Roudier, Y.: "Password Strength: An Empirical Analysis". INFOCOM, 2010 Proceedings IEEE, pp. 1-9. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5461951&abstract Access=no&userType=inst.

[5] M. Bishop, "Improving system security via proactive password checking," Computers & Security, vol. 14, no. 3, pp. 233–249, 1995.

[6] J. J. Yan, "A note on proactive password checking," in Proc. NSPW '01. ACM, 2001, pp. 127–135.

[7] W. E. Burr, D. F. Dodson, and W. T. Polk. Electronic authentication guideline. Technical report, NIST, 2006.

Comparative Study of Huffman and LZW Text Compression for Efficient Transmission and Storage of Data

Mi Mi Hlaing, San San Nwel, Tin Tin Thein
University of Computer Studies (Pakokku)
mimihlaing@ucspkku.edu.mm, sansannwel@ucspkku.edu.mm

Abstract

Today, text compression is an important technology for all computer users. Compression is an easy way to store large amounts of data. Due to cost issues, data compression is important for business data processing. Save data and provide large amounts of data processed by many business applications for efficient data transfer and efficient storage. This study focuses on the performance analysis of data compression algorithms for Huffman and Lempel-Ziv Welch (LZW). The experimental results of the Huffman algorithm have better data compression performance than the LZW algorithm. According to the testing result, Huffman algorithm can reduce the data size by an average of 55 % and LZW algorithms can reduce the data size by 61%. In the process, Huffman is faster than the LZW algorithm.

Keywords: data compression, transmission, storage, Huffman, LZW, performance

1. Introduction

Data compression means reducing the space required to store and reducing the time required to transfer data. The size of data can be reduced by removing the excessive information. The purpose of data compression is to express a number with fewer bits while satisfying the original restored [1].

Data compression is lossless and only the original data can be correctly rebuilt from the compressed version. This lossless technology is used when the raw data of the source is important and we cannot lose detail. Huffman and LZW are both lossless data compression technologies [2].

Another type of compression algorithm is called a lossy algorithm because they irreversibly delete some data and can only reconstruct approximations of the original data. Approximate rebuilds may be ideal because it may lead to more efficient compression. However, it is often necessary to strike a good balance between visual quality and computational complexity. Because of the way human visual and auditory systems work, data such as multimedia images, video, and audio are easier to compress with lossy compression [2, 3].

In lossless algorithms, lossy algorithms have an improved compression effect, but lossy compression is limited to audio, images, and video, and some losses are acceptable. Two great technologies, “lossless" or "lossless" are meaningless because each method has its use, lossless technology may be good, or lossless technology may be excellent [4, 5].

In this study, two different data compression methods were analyzed, namely the Huffman algorithm and Lempel-Ziv Welch (LZW) algorithm. The purpose of this work is to determine the method that may result in the highest compression ratio and performance.

The organization of the paper is as follow. The section I introduces the system. In section II, two compression techniques are presented. Experimental result and conclusion of the system are presented in section III and IV, respectively.

2. Compression Techniques

Data compression methods and algorithms

can be divided into two different ways: lossless and lossy [6].

- Lossless compression is a class of data compression method that allows the original data to be perfectly reconstructed from the compressed data.
- Lossy compression is a type of compression that removes unnecessary components of a file to reduce the file's size.

2.1. Huffman Compression Technique

Huffman compression is a lossless compression algorithm that is ideal for compressing text and program files. This may explain why it is commonly used to compress programs such as ZIP and ARJ.

Huffman compression belongs to a series of algorithms with variable code word lengths. This means that one symbol, such as a character in a text file, is replaced with a bit sequence of different lengths. Therefore, symbols that occur frequently in a file are given short sequences, and other symbols rarely get long bit sequences [7].

The example of Huffman compression techniques is presented as following. Suppose the piece of data "ACDABA" are compressed.

Since there are 6 characters, this text is 6 bytes or 48 bits long. With Huffman encoding, the file is searched for the most frequently appearing symbols (in this case the character 'A' occurs 3 times) and then a tree has been built and the symbols are replaced by shorter bit sequences. In this particular case, the algorithm would use the following substitution table: A=0, B=10, C=110, D=111. If these codes are used to compress the file, the compressed data looks like this:

01101110100

This mean that 11 bits are used instead of 48, a compression ratio of 4 to 1 for this particular file.

A Huffman tree or Huffman coding tree is defined as a complete binary tree, where each leaf of the tree corresponds to the letter in a given alphabet.

The Huffman tree is considered a binary tree associated with the minimum outer path weight and means the minimum weighted path length and associated binary tree for a given leaf set. Therefore, the goal is to build a tree with the smallest external path weight.

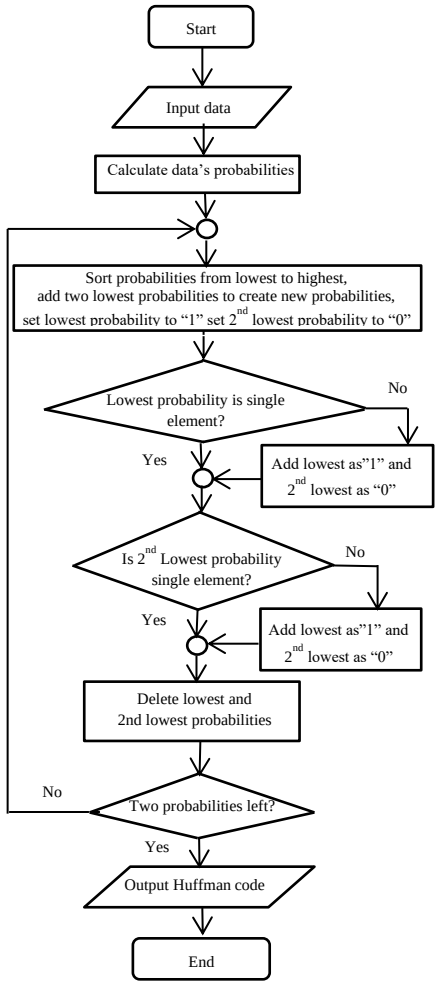

Figure 1. Huffman Encoder

The Huffman encoder uses the frequency of instructions to determine the length of the code word that replaces the original code word. Commonly used instructions use short code words instead of infrequently [8]. The flowcharts for Huffman encoders and decoders are shown in figure 1 and 2 respectively.

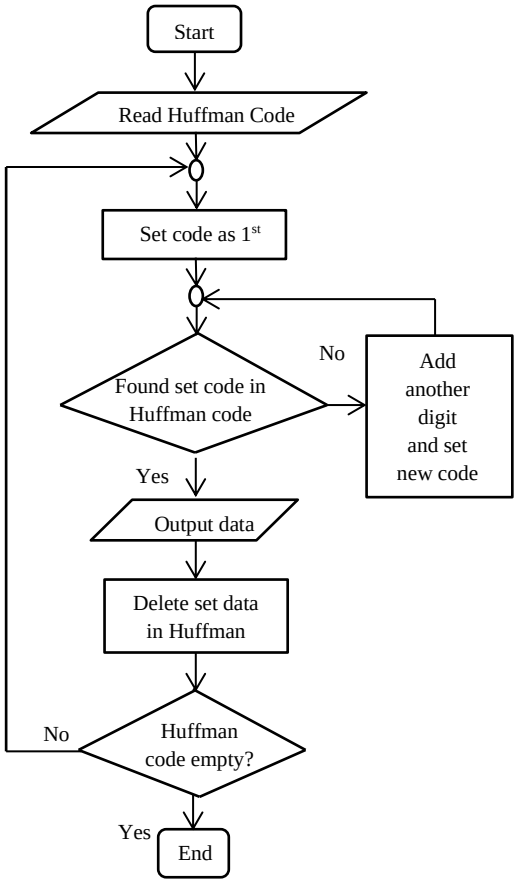

Figure 2. Huffman Decoder

2.2. LZW Coding Technique

A typical file data compression algorithm is called LZW - Lempel, Ziv, and Welch encoding. Variations in this algorithm are used in many file compression schemes, such as GIF files. These are lossless compression algorithms that do not lose data and allow the original file to be completely reconstructed from the encoded message file [9].

The LZW algorithm is a greedy algorithm that recognizes and encodes increasingly long repeating phrases. The definition of each phrase has a prefix with a prefix equal to the previously encoded phrase and additional letters of the alphabet. Note: "Alphabet" refers to the legal character set in the file. For a normal text file, this is the ASCII character set. For a 256 gray-level gray level image, this is an 8-digit number that represents the gray level of the pixel [9, 10].

In many texts, some character sequences are displayed frequently. For example, in English, the word occurs more often than any other sequences of three characters: with *and*, *ion*, and *ing* closed behind. Including space characters, there are other very common sequences that contain long sequences. We can't improve Huffman encoding by assigning a fixed encoding to each character, but we can do better by encoding the entire sequence of characters using a few bits. These methods take advantage of frequently occurring length character sequences.

Typically, a representation smaller than the Huffman tree is generated, and unlike basic Huffman encoding, 1) the text is read only once, and 2) no additional overhead space is required for the compressed representation. This algorithm uses a dictionary that contains a sequence of dynamically selected characters from the text. Each sequence of characters associates a dictionary with a number. Numeric code words are called s code numbers or code numbers. All codes have the same length in bits; a typical code size is twelve bits, which permits a maximum dictionary size of 2^{12} = 4096 characters sequences [9].

The example of LZW compression technique is presented as following. Suppose the piece of data " a a b a b a c b a a c b a a d a a " are compressed. The dictionary is initialized with all possible single-character sequences, that is, elements of the text alphabet (assuming the N symbol of the alphabet) are assigned code numbers 0 through N-1, and all other code numbers are not initially assigned. Text 'w' is encoded using greedy heuristics: at each step determines the longest prefix 'p' for 'w' in the dictionary, prints the code number of 'p', removes 'p' before w, and invokes the current match of 'p'. Each step modifies the dictionary by adding new string, assigning an unused code number to the next. The string we add consists of the first character of the rest of 'w' and the current match concatenated. It is easier to add this string to the

next step. In the first step, the string is not added to the dictionary [11].

The LZW encoder and decoder flow charts are shown in figure 3 and 4 respectively.

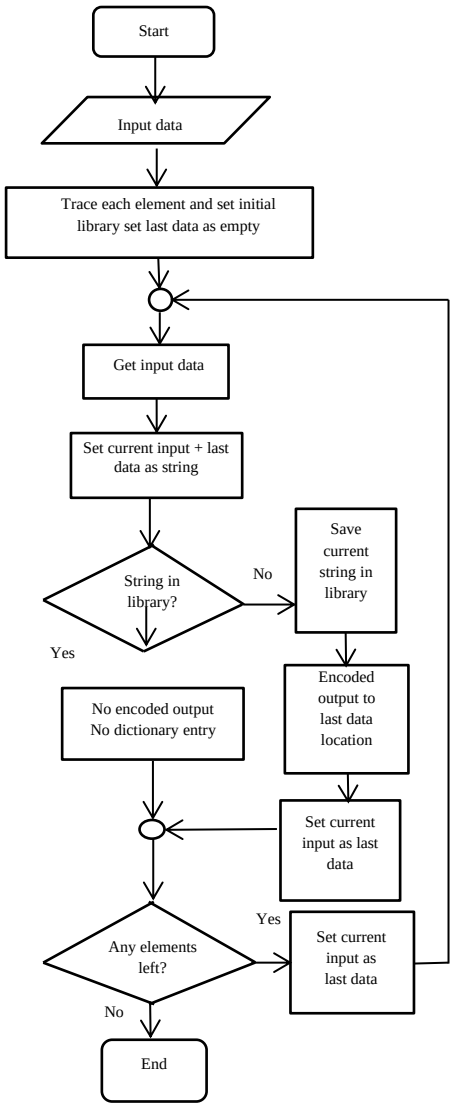

Figure 3. LZW Encoder

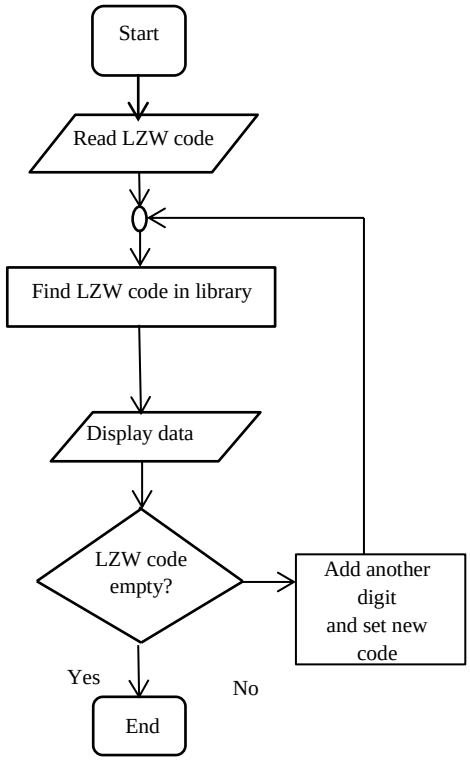

Figure 4. LZW Decoder

3. Compression Performances

Different criteria are measured, depending on the nature of the application. In the compression algorithm, the main concern when measuring performance are the space efficiency (compression ratio) and time efficiency (compression time).

3.1. Compression Ratio

Compression Ratio is the ratio between the size of the compressed file and the size of the source file.

The compression ratio are calculated using the following equation (1).

$$\textbf{Compression Ratio} = \frac{\textbf{size after compression}}{\textbf{size before compression}} \quad \textbf{(1)}$$

3.2. Compression Time

Compression and decompression times must be considered separately. The compression time and decompression time are increased as file size increases. Both compression and decompression time are measured according to the algorithms used such as Huffman and LZW. With the development of high-speed computer accessories, this factor may give very small value. They may depend on the performance of the computer. We measure the compression and decompression time of different sized text files on Core i5 (1.80 GHz) processor with Windows 10.

4. Experimental Results

In this section, we focus on comparing the performance of compression techniques (Huffman and LZW). The efficiency of compression algorithm can be evaluated using two criteria. One is the amount of compression obtained, and the other is the time used by the encoding and decoding algorithms.

Figure 5 shows the comparative results between Huffman and LZW compression techniques. In this experiment, the ten different data sizes of text file are tested. The tested file sizes are 10K, 20K, 40K, 80K, 160K, 320K, 640K, 1280K, 2560K and 5120K bytes.

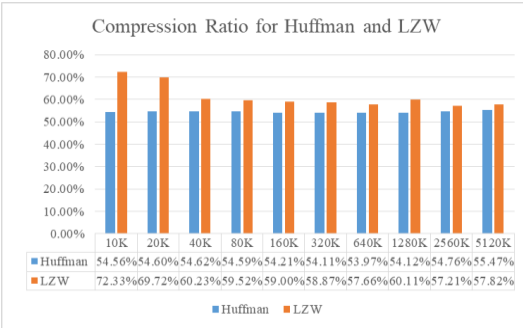

Figure 5. Compression Ratio for Huffman and LZW

The row of Figure 5 represents the compression ratio in percent and the column represents the file size in bytes. In the 10K bytes file size, the compression ratio of Huffman algorithm is 54.56% and the LZW is 72.33%. In 5120K bytes, the Huffman is 55.47% and the LZW is 57.12%. Therefore, according to the tested results, Huffman is very efficient in compression process for data storage and data transmission than LZW.

The results of the time taken to compress and decompress different data sizes using the Huffman and LZW algorithms are shown in figure 6 and 7. According to the tested results, the Huffman algorithm is faster encoding and decoding than the LZW algorithm.

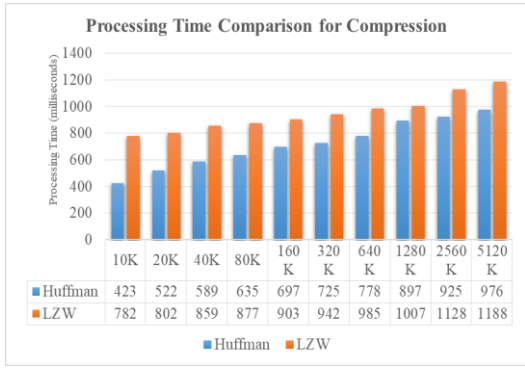

Figure 6. Compression Processing Time

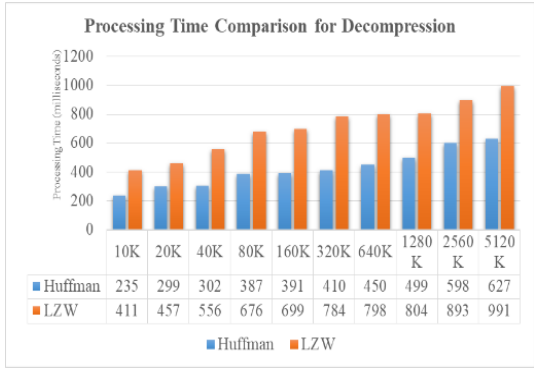

Figure 7. Decompression Processing Time

5. Conclusion

Today, many data processing applications require storage of large volume of data. At the same time, the proliferation of computer

communication networks is resulting in massive transfer of data over communication links. When the amount of data to be transmitted is reduced, the capacity of the communication channel can be increased. Moreover, lossless data compression technique become an important field as it significantly can reduce storage requirement and communication cost.

This study analyzed the compression performance of the Huffman algorithm and LZW algorithm using different data sizes. According to the experimental results, in compression ratio, Huffman algorithm can reduce data size by 55% on average but LZW algorithm can reduce data size by 70% on average. For certain test data, the Huffman algorithm performs better in terms of data transfer and data storage compression rate and computation time than LZW.

References

[1] Senthil Shanmugasundaram, Robert Lourdusamy, "A Comparative Study Of Text Compression Algorithms", International Journal of Wisdom Based Computing, Vol. 1 (3), December 2011.

[2] Khalid Sayood, “Introduction to Data Compression”, 2nd Edition, San Francisco, CA, Morgan Kaufmann, 2000.

[3] https://en.wikipedia.org/wiki/Lossy_ compression.

[4] Seyyed Mahdi Najmabadi, Philipp Offenhäuser, "Analyzing the Effect and Performance of Lossy Compression on Aeroacoustic Simulation of Gas Injector", Journal of Computational Science and Engineering, 2017.

[5] https://ww.geeksforgeeks.org/difference-between-lossy-cmpression-and-lossless-compression/

[6] K.A. Ramya, M.Pushpa, "A Survey on Lossless and Lossy Data Compression Methods", International Journal of Computer Science and Engineering Communications, Vol.4, Issue.1, Page.1277-1280, (2016)

[7] https://www.prepressure.com/library/compression-algorithm/huffman

[8] Gonzalez, R.C. and R.E. Woods. "Digital Image Processing.", 1st Edn., Prentice Hall, Upper Saddle River, ISBN-10: 013168728X, pp: 954,2008.

[9] Harper Collins, "Data Structures & Their Algorithms", Publishers, Harry R. Lewis and Larry Denenberg, 1991.

[10] A. Drozdek, "Data Structures and Algorithms", Brooks/Cole 2001.

[11] Shahbahrami, A., R. Bahrampour, M.S. Rostami and M.A.Mobarhan, "Evaluation of huffman and arithmetic algorithms for multimedia compression standards.", Int. J. Comput. Sci., Eng. Appli., 1: 34-47 2011.

Analysis of Quantum Cryptography

Mya Thandar Phyu
UCS (Hpa-An)
mtdphyu@gmail.com

Nan Myint Myint Htwe
UCS (Thaton)
htwelay08@gmail.com

Nan Sandar Thin
UCS (Hpa-An)
kyawsandaaung@gmail.com

Abstract

The internet has become a global mean of communication, turning our reality upside down. Change communication to the level of communication standard. There are many ways to protect information from unwanted access. For security, an important issue of personal information can be stored with computer in the cloud. There are many cryptographic techniques which are used to provide security. However, all cryptographic techniques will be ineffective if the key distribution mechanism is weak. Quantum cryptography can be used for unconditional, secure communications using the laws of quantum physics. This paper proposes the analysis of key security for different Quantum cryptography protocols and compare of these protocols. The importance of this paper is to mark the rise of quantum cryptography, its elements and quantum key distribution.

Keywords: photons polarization, quantum cryptography, quantum key distribution protocol, qubits

1. Introduction

Quantum cryptography is a combination of quantum mechanics and classical cryptography, which is an important part of cryptography. Communication with Heisenberg's uncertainty principles and quantum theory can be guaranteed. The main function of quantum cryptography is to generate secure keys. This paper is focused on quantum cryptography protocols in various parameters and summarize the weakness of Quantum over traditional cryptography.

The structures of this paper are the following: Section 2 presents the related works. Section 3 presents the attribute of Quantum Cryptography. Section 4 discusses about the probability of a photon transmitted and their difference which shown in the figure. Section 5 discusses the analysis of different Quantum Cryptography protocols. Section 6 presents the comparison of different Quantum Cryptography protocols. Finally, section 7 gives a Conclusion.

2. Related Works

Quantum cryptography is derived from the concept of quantum money, which was proposed by Wiesner in 1969. The level of technology is limited in the history of creativity and this creativity cannot be realized, which is not published until 1983[16]. Quantum cryptographic protocol is also called Quantum Key Distribution (QKD) protocol.

In 1991, Eker proposed the protocol [2] is based on Bells theorem. Note [2] employs a pair of quantum bits (i.e, an ERP pair), which is essentially the same as [6].Subsequently, in 1992, the improvement [5] of the scheme [6] was put forward by Bennett. Employing any two nonorthogonal states, progress is simpler and more effective. After that, many QKD protocols [4, 7] using the basic principles of quantum mechanics have been finalized. First, practical of QKD protocol [6] was proposed by Bennett and Brassard, in 2011. Quantum cryptography protocols are really single photon polarization, implementation pioneers of their key distribution protocol. After that, a lot of effort was put into QKD to improve security and effectiveness. BB84 protocol has been experimentally demonstrated to act correct with bit rate of

1Mbit/s over 20 km and 10 kbit/s over 100 km of fiber optic cable [15].

3. Quantum Cryptography

Quantum Cryptography is based on the principles of quantum physics, giving us the safest way to transmit data to secure communications based on quantum mechanics laws. Quantum mechanics is the mathematical framework or set of rules which is used for the construction of physical theories. The two important elements of quantum cryptography on which quantum cryptography depends are: Heisenberg Uncertainty Principle and Photon Polarization Principle [10].

3.1. Heisenberg Uncertainty Principle

This principle says that if you measure something, you measure something else exactly. It's not available. For example, if you apply this principle to people, you can measure a person's height, but the measure of his weight is not available. This law applies to photons. The photons have a wave-like structure and are polarized or in some directions. This principle is central to the attacker publishing quantum encryption [3].

3.2. Photon Polarization Principle

In this case, the hacker could not replicate any of the different "qubits" (ie, quantum states). If you try to measure any properties, it will interfere with other information. The photon is randomly polarized at a base and is known as qubit. The photon is polarized randomly in one of the bases to represent a bit known as a qubit. A 0° polarization of photon in the rectilinear basis or 45° in the diagonal basis is used to represent a binary 0. A 90° polarization in the rectilinear basis or 135° in diagonal basis is used to represent a binary 1 as shown in Figure. 1 [13].

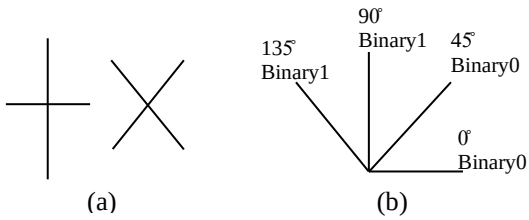

Figure 1. (a) Rectilinear and diagonal bases (b) Polarization of photons to represent bits

3.3. Qubits and Quantum States

The underlying unit of quantum cryptography is qubit. It has two states, labeled as $|0\rangle$ and $|1\rangle$ (vertical bars | and angle brackets ⟩) and referred as a state, a Ket or a Dirac notation named after its framer Paul Dirac who had instigated this notation in 1939 [19].

A bit can be in the state 0 or 1 whereas a qubit can occur in the state $|0\rangle$ or$|1\rangle$. It can also occur in superposition state which is a linear combination of the states $|0\rangle$ and$|1\rangle$. A state can be labeled as $|\Psi\rangle$. The state in superposition is noted as $|\Psi\rangle$ = state $\alpha|0\rangle+\beta|1\rangle$ where α, β are complex numbers [19].

Perhaps a qubit occurs in a superposition state $|0\rangle$ and $|1\rangle$, but this state cannot be measured. Certainly, when a qubit is measured, it will occur in the state $|0\rangle$ or in the state$|1\rangle$.

The probability of obtaining the state $|0\rangle$ or $|1\rangle$ qubit is the modulus squared of α, β respectively according to quantum mechanics laws. The represent of the probability of obtaining $|\Psi\rangle$ in $|0\rangle$ state is $|\alpha|^2$ and the probability of obtaining $|\Psi\rangle$ in $|1\rangle$ state is$|\beta|^2$. The probability of getting result of a measurement is obtained by squaring the coefficients. The condition is $|\alpha|^2+|\beta|^2 = 1$ [19]. In table 1, shown the representation of quantum bits (Qubits).

Table 1. Representation of quantum bits (Qubits)

Quantum Bits (Qubits)	
Symbols	**Binary number**
α	00
β	01
μ	10
€	11

3.4. Notations

$$|0\rangle = |-\rangle \quad (1)$$

$$|1\rangle = |\rangle \quad (2)$$

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |/\rangle \quad (3)$$

$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |\backslash\rangle \quad (4)$$

Equation (1) and (2) are represents state 0° and state 90° in the rectilinear basis respectively. Equation (3) and (4) are state 45° and state 135° in diagonal basis respectively [17].

4. Probability of a Photon

There are many probabilities of a photon which gives the transmission in one angle and produces in same/other angle.

4.1. Probability of a Photon Transmitted and Produced in Same Basis and State

The probability of an unpolarized light transmits through vertical polarizer or basis which can be gained up to vertical polarizer is 100%, as shown in figure 2. This is for the same direction of polarized light.

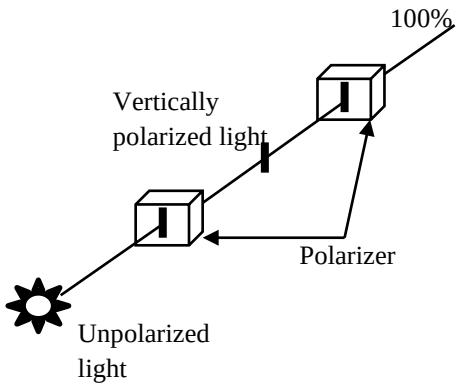

Figure 2. An unpolarized light transmitted through vertical polarizer and produced in vertical polarizer

4.2. Probability of a Photon Transmitted and Produced in Same Basis but Prepared in Difference states

The probability of an unpolarized light transmitted through vertical polarizer prepared at an angle 90° of rectilinear basis and produced in horizontal polarizer at 0° of rectilinear basis is 0% that is shown in figure 3.

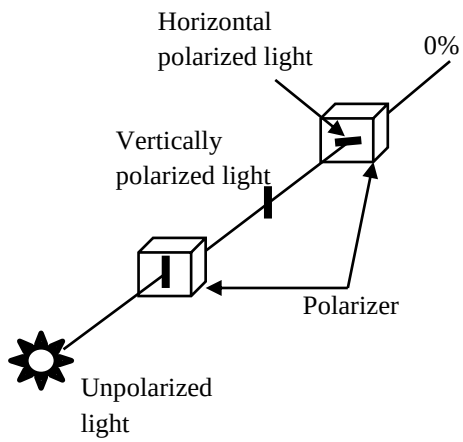

Figure 3. An unpolarized light transmitted through vertical polarizer and produced in horizontal polarizer

4.3. Probability of a Photon Transmitted and Produced in Different states

The probability of an unpolarized light transmits through vertical polarizer and produce 45° diagonal basis 50% shown in figure 4.

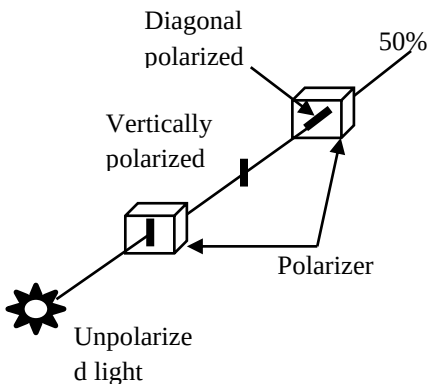

Figure 4. An unpolarized light transmits through vertical polarizer and produces in diagonal polarizer

4.4. Probability of a Photon Transmitted in One Basis and Produces at an Angle

The probability of an unpolarized light transmits through vertical polarizer and produces through a polarizer an angle θ in the vertical direction of $\cos^2 \theta$ [9][17] shown in figure 5.

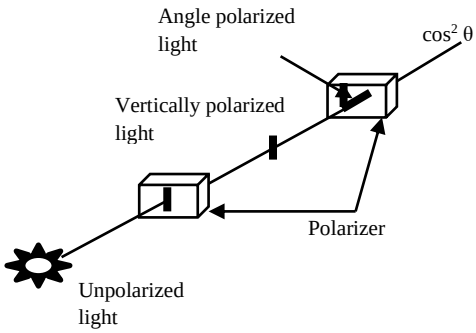

Figure 5. An unpolarized light transmits in rectilinear polarizer and produces at an angle

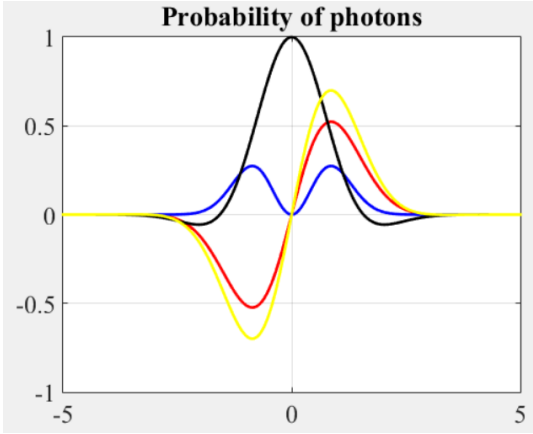

Figure 6. Analysis the probability of photons in different conditions

Finally, different conditions of Probability of photons are shown in figure 6 by using matlab. The red line represents the vertical polarizer, the black line represents the diagonal polarizer, the blue line represents the angle and the yellow line represents the horizontal polarization.

5. Analysis of Different Quantum Cryptography Protocols

5.1. BB84 Protocol

In 1984, Brassard and Bennett distributed their BB84 protocol, which is currently the most famous Quantum Key Distribution (QKD). The BB84 protocol involves the four polarization states: horizontal | h>, vertical | v>, left circular polarization | lcp>, right circular polarization | rcp> in state These four polarization, horizontal polarization status and circular polarization status of the left circle indicate '0' and the vertical status of polarization and the right circular polarization status which show up to a '1' [14].

5.2. BB92 Protocol

It is not important to use two orthogonal bases for encoding and decoding data according to state polarization. The BB92 protocol uses the same procedure as BB84. However, using only

two non-orthogonal quantum states | h> instead of '0' and | rcp> instead of '1' half of the BB84 protocol to send the key [14].

5.3. ERP Protocol

In 1991, Artur Ekert explained about a quantum particle pair, ERP pair. The ERP pair states as solution of double particles that spatial, the ERP pair is released. The situation states a relationship of choosing and measuring a particle which leads to the direction and dimension of the particle. This conduct is called "action at a distance" because the particles are separated at far distance [11].

5.4. SARG04 Protocol

The BB84 protocol is used for four polarization states. If these four states are encoded with different records, it will give a new protocol called SARG04, which is strong against photon-number-split attacks, when it is reduced pluses of a single photon laser sources.

Alice sent a 'photon' bit of length through the public channel to Bob. Bob chose his basis when calculating the qubits communicated by the sender. If Bob matches the same bit sent by the sender, he announces openly that he has received Alice's transmission. If there are errors in the key length on sides, Alice and Bob can clear the process and start over [12].

5.5. S13 Protocol

Eduin H, Sema, introduced this protocol in 2013 to prevent the loss of data packets through the signal channel. S13 is created in a seed state called random seeds and public key encryption. It has been proven safe by creating secret keys of the same size in many responses. This protocol differs from BB84 only [1].

5.6. One Time Pad-OTP Protocol

In this protocol, real key pairs of the same length are used randomly when plain text is used to send to Bob and known as the key before encryption. This key is mixed by (XOR-ing) bit by bit and combining the keys with the mixed bit of plain text to create a bit of cipher text. Next, the encrypted messages will be mixed (XOR-ed) with a duplicate of One Time Key, after that plain text will be recovered [18].

5.7. AK15 Protocol

Transmitted information using the deception status into the stream of qubits is more secure and makes it more difficult for hackers to capture information based on that content. The receiver can easily receive sequences of qubits that are transmitted over the narrow classic channel without requiring additional communication. Therefore, this protocol is resistant to Intercept-Resend - Attack (IRK), MIMA and waste time [1].

The figure 7 is shows the analysis of quantum cryptograph protocols for key securities. The BB84 protocol has a 96% secure key and BB92 protocol is 96.57%. The key pair security of the ERP protocol is 96.83% while SARG04 has 98% security. The key security of S13, OTP and AK15 is 98.56%, 99.23% and 99.89%.

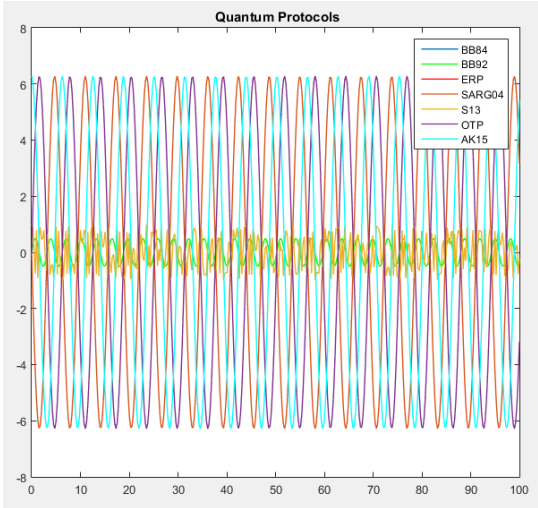

Figure 7. Analysis of quantum cryptography protocols for key securities.

6. Compare the Difference between Quantum Cryptography Protocols

BB84, S13 and AK15 protocols are polarization situation which has 2 orthogonal. BB92 polarization has 1 orthogonal. SAR04 polarization is coded with bits. ERP and OTP have no polarization. BB84, BB92, SAR04, S13 and AK15 protocols depend on Heisenberg Uncertainly principle. However, ERP and OTP depend on Quantum Entanglement principle. All about of these protocols do not have Bell's-in-equality except AK15. OTP and AK15 protocols do not have classical channel. However, other protocols should include in this channel. While the process states of BB84, SARG04 and S13 are 4, ERP has 3 states, BB92 has 2 states, AK15 has n states and OTP has no states. BB84 protocol is vulnerable to man-in-the-middle attack. BB92, SARG04, ERP and AK15 protocols are robust against Intercept-Resend.

7. Conclusion

Techniques adapted from classical science can be applied to quantum key distribution protocols, as appropriate signals that quantum cryptography are new areas of research. Quantum cryptography is considered the future model of traditional cryptography. Quantum cryptography is built on quantum values of physics and can be thoroughly proven by creating secret keys. This paper presents the analysis of secure keys for quantum cryptography protocols and probability of photon for different conditions and compares the quantum cryptography protocols.

Acknowledgements

I want to very special thank for inviting in University Journal of Research and Innovation, 2020 of University of Computer Studies (Pakokku). Special thanks to teacher Daw Thu Thu Han and teacher Daw Su Su Khine from the English Department for checking the English language and grammar storage. Thanks to all our teachers and friends for their support in this paper.

References

[1] A. Abuhgra and K. Elleithy, "QKDP's Comparison Based upon Quantum Cryptography Rules," in IEEE, 2016.

[2] A. K. Ekert, "Quantum cryptography based on Bellas theorem", Physical Review Letter, vol. 67, no. 6, pp. 661-663,1991.

[3] A. Peres and L. E. Ballentine, "Quantum Theory: Concepts and Methods", American Journal of Physics, vol.63,no. 3, pp. 285– 286,1995.

[4] B. Huttner, N. Imoto, N. Gisin, and T. Mor, "Quantum cryptography with coherent states, "Physical Review A: Atomic, Molecular and Optical Physics, vol.51, no.3,pp.1863-1869,1995.

[5] C.H. Bennett, "Quantum cryptography using any two nonorthogonal states", Physical Review Letters, vol. 68, no. 21, pp. 3121-3124, 1992.

[6] C. H. Bennett and G. Brassard, "WITHDRAWN: Quantum cryptograph: Public key distribution and coin tossing", Theoretical Computer Science, 2011.

[7] D. BruB, "Optimal eavesdropping in quantum cryptography with six states", Physical Review Letters, vol. 81, no.14, pp. 3018-3021,1998.

[8] D. N. Kartheek, G. Amamath, P. Venkateswarlu Reddy, "Security in quantum computing using quantum key distribution protocols", 978-4673-5090-7/13/$31.00©2013 IEEE.

[9] Dr. PhysicsA, Quantum Mechanis Concepts: 1 Dirac Notation and Photon Polarization, Published on Aug 20, 2013, Available: https://www. youtube.com/ watch?v=pBhzXqbh5JQ.

[10] Harshad R.Pawar, Dr.Dinesh G. Harkut, "Classical and quantum cryptography for image encryption and decryption", 978-1-5386-2599-6/18/$31.00©2018 IEEE.

[11] M. Elboukhari, M. Azizi and A. Azizi, "Quantum Key Distribution Protocols: A Survey", International Journal of Universal Computer Sciences, vol. Vol.1, 2010.

[12] M. Lopes and D. Sarwade, " On the performance of quantum cryptographic protocols SARG04 and KMB09", i"n IEEE, 2015.

[13] Ms. V. Padmavathi, Dr. B. Vishnu Vardhan, Dr. A. V. N.Krishna, "Quantum cryptography and quantum key distribution protocols: A survey", 978-1-4673-8286-1/16$31.00©2016 IEEE.

[14] P. Techateerawat, “A Review on Quantum Cryptography Technology”, International Transaction Journal of Engg, Mangmt, Applied Sciences & Technologies, vol. Volume 1, 2010.
[15] P.Winiarczyk, W.Zabierowski, “BB84 analysis of operation and practical considerations and implementations of quantum key distribution systems”, CADSM’2011,23-25 February, 2011, Polyana-Svalyava (Zakarpattya, UKRAINE.
[16] S. Wiesner, “Conjugate coding”. ACM SIGACT News, vol.15,no.1,pp. 78-88,1983.
[17] Wiesner, S., “Conjugate Coding”. Sigact News, Vol. 15. No. 1. 1983. pp. 78-88; original manuscript written circa 1969.
[18] “mils electronic”, 1947. [online]. Available: https://www. mils.com/.
[19] https://www.researchgate.net/publication/314190-604_Quantum_Cryptography_A_Review

University of Computer Studies (Pakokku)
Department of Higher Education
Ministry of Education
Myanmar