



The Effect of Weight Factors Characters on Selecting a Password

تأثير عوامل وزن الأحرف على اختيار كلمة السر

**Prepared by
Ghadeer Ali Shaheen
200810372**

**Supervisor Prof.
Dr. Ahmad Al-Jaber**

**Thesis Submitted in Partial Fulfillment of the
Requirement for the Degree of Master in
Computer Science**

**Department of Computer Science
College of Computer Science and Informatics
Amman Arab University**

September, 2011

I, the undersigned "Ghadeer Ali Mohammad Shaheen" hereby authorize Amman Arab University to provide copies of this thesis to libraries, institutions and any other parties upon their request.

Name: Ghadeer Ali Mohammad Shaheen

Signature:



Date: 28/9 / 2011

Resolution of the Examining Committee

This thesis titled " The effect of weight factors characters in selecting password & password recovery ". Has been defended and approved on 28 /09 /2011.

Examining Committee

Dr. Moayad abd al razaq

Dr. Ala Alaa Al Hamami

Dr. Ahmad Al Jaber

Signature

M. A. Razaq

Alahamami

A. Aljaber

Dedication

To my husband thanks for his help and support.

To my father , to my mother the gardenia flower

To my sisters and brother.

To my supervisor Dr. Ahmad Al Jabir .

and to all my friends especially Mr. Aladdin Qaraqish

Abstract

With today's technology, a user can store his/her whole life on computer hard drive. With password protection, everything is safe. A password works as a strong lock that protects information and data in any personal computer or large system.

Humans can easily create memorable passwords, but this also creates the problem that the generating process is guessable either online/ offline, or by dictionary attacks, brute force attacks, Rainbow Tables or by Social Engineering techniques.

In this thesis, an effective method to generate random passwords has been created and implemented using Huffman coding algorithms.

In the creation phase, the user enters six characters from four different groups (capital characters, small characters, digits, and symbols), the program generates different passwords.

The generated passwords are classified into five scores according to Huffman Coding entropy scores.

The passwords which we obtain from Huffman Coding results will be checked again according to Password Meter checking weight schemas.

Therefore, the good results obtained through the system indicate its ability to create a safe and strong password.

Arabic summary

الخلاصة

مع التكنولوجيا الحالية ، يمكن للمستخدم تخزين كامل تفاصيل حياته على جهاز الكمبيوتر الثابت. كل شيء آمن عند استخدام كلمة العبور/السر . كلمة السر تعمل على تأمين حماية قوية للمعلومات والبيانات سواء على الجهاز الخاص أو على الأنظمة الواسعة.

يمكن للانسان بسهولة اختيار كلمات عبور لا تنسى، ولكن في الوقت نفسه يكون من السهل اختراقها إما عن طريق التخمين الفوري أو التخمين الغير متصل بالإنترنت ، أو عن طريق استخدام القاموس الهجمات و التقنيات الهندسية للكشف عن كلمات العبور.

في هذه الأطروحة، تم إيجاد وسيلة فعالة لإنشاء كلمات السر العشوائية وتطبيقها على Huffman Coding algorithms.

في مرحلة الإنشاء ، على المستخدم إدخال ستة أحرف من أربع مجموعات مختلفة (حروف كبيرة ، الحروف الصغيرة، الأرقام والرموز) ، يقوم البرنامج بإنشاء كلمات مرور مختلفة.

كلمات المرور المتولدة يتم تصنيفها إلى خمس درجات وفقا Huffman Coding entropy scores.

يتم التحقق من كلمات المرور التي تم الحصول عليها من Huffman Coding مرة أخرى وفقا ل Password Meter checking weight schemas.

النتائج التي تم الحصول عليها من Huffman Coding و Password Meter checking ، يتم تحليلها وتقديم اقتراحات عدة للمستخدم لاختيار كلمات العبور/السر.

النظام يظهر مقدرة في إنشاء كلمات مرور آمنة وقوية.

Contents

Dedication	IV
Abstract.....	V
Arabic summary	VI
Contents	VII
List of Figures	IX
List of Tables	XIV
List of abbreviations	XVII
Chapter one Introduction	1
1.1. History of the Internet	1
1.2. History of Password	4
1.3. History of Hacking	4
1.4. Applications	7
1.5. Statement of the Problem	7
1.6. Previous Work.....	8
1.6.1. Password Security: A Case History (1979).....	8
1.6.2. Password Generation and Search Space Reduction (2009).....	9
1.6.3. Password Protection?(2008).....	10
1.6.4. An Image of the Future: Graphical Passwords (2006).....	11
1.6.5. Design and evaluation of a shoulder – Surfing Resistant Graphical password Scheme.....	12
1.6.6. Memorability of alternative password systems (2009).....	13
1.6.7. Hybrid Password Authentication Scheme Based on Shape and Text (2010).....	16
1.7. Objectives of the Thesis	19
1.8. Outline of Forthcoming Chapters	19
Chapter Two Password.....	20
2.1. Internet Security.....	20
2.2. Computer Crimes	20
2.3. Password	22
2.3.2. Store Password in Computer system	29
2.3.3. Password Strength.....	30
2.3.4. Tools available for Password Strength Checking	34
2.3.5. Password and Hacking	39
Chapter Three Huffman Coding.....	43

3.1 Huffman Coding.....	43
3.2 Implement Huffman Coding on Passwords.....	52
Chapter Four Result and Discussion	57
4.1 Experimental Results	57
Chapter Five Conclusion and Recommended Future Tasks.....	71
5.1. Conclusion.....	71
5.2 Recommended Future Tasks	74
References	75

List of Figures

1.	Five pass	– 9
1	icons.....	
1.	Example of a convex hull with 5 pass	– 9
2	icons.....	
1.	Example of a symbolic	1
3	password.....	0
1.	Password set	1
4	procedure.....	2
1.	Login	1
5	interface.....	2
	...	
1.	Original stroke on the	1
6	interface.....	3
1.	Stroke variants of	1
7	interface.....	3
2.	Graphical password scheme suggested by	1
1	Blonder.....	9
2.	Passfaces.....	2
2	0

2.3	The	story	2
	scheme.....		1
		
2.4	Password	in	Grid-Based
	schemes.....		2
2.5	Selection	panel	in
	step.....		graphical
			2
			3
2.6	Frequencies	of	letters
		2
			5
2.7	2 or 3 letters appearing more often in a normal		2
	sentence.....		6
2.8	Password	Meter	3
	program.....		0
2.9	Password	Meter	3
	program.....		0
2.1	Password	Meter	3
0	program.....		0
2.1	Password	Meter	3
1	program.....		0
2.1	Password	Meter	3
2	program.....		1
2.1	Microsoft	password	checker
3	program.....		3
			1

2.14	Microsoft program.....	password	checker	31
2.15	Microsoft program.....	password	checker	31
2.16	Microsoft program.....	password	checker	32
2.17	Microsoft program.....	password	checker	32
3.1	building tree.....	Huffman	coding	38
3.2	building tree.....	Huffman	coding	38
3.3	building tree.....	Huffman	coding	38
3.4	building tree.....	Huffman	coding	39
3.5	building tree.....	Huffman	coding	39
3.6	building tree.....	Huffman	coding	39
3.7	building tree.....	Huffman	coding	40

3.8	building	Huffman	coding	4
	tree.....			0
3.9	Huffman		Coding	4
	algorithm.....			0
3.1	Final	Huffman	Coding	4
0	Tree.....			1
3.1	Password	generator	program	4
1	interface.....			4
3.1	Password		generator	4
2	algorithm.....			5
3.1	Huffman	Coding	Checking	4
3	algorithm.....			5
3.1	Password	Meter	Checking	4
4	Algorithm.....			6
3.1	Software			4
5	processes.....			7
			
4.1	Password		generator	4
	interface.....			8

4.	Same	class	5
2	Classification.....		2
4.	Positive results very closed to each		5
3	other.....		4
4.	Negative		5
4	results.....		6
	..		

List of Tables

2.	Top	10	2
1	password.....		8
	...		
2.	Scheme of weights		2
2	assigned.....		9
2.	Password Classification for the final		2
3	score.....		9
3.	Characters in ASCII		3
1	Code.....		6
3.	Characters in Unicode		3
2	Code.....		6
3.	Frequency of Characters in a data		3
3	file.....		7
3.	Huffman final		4
4	table.....		1
3.	Final		4
5	result.....		2
	...		
3.	Characters and its		4
6	probabilities.....		3

3.7	Huffman	coding	result	43
	(1).....			
3.8	Huffman	coding	result	43
	(2).....			
3.9	Huffman	Coding	general	45
	classifications.....			
3.10	Password	Meter	General	46
	Classifications.....			
4.1	Huffman	Coding	results	49
			
4.2	Huffman	Coding	general	49
	classifications.....			
4.3	Password		Meter	50
	results.....			
4.4	Password	Meter	general	classifications
			51
4.5	New	classification	for	password
			results
				51
4.6	passwords	results	with	the
	classification.....			same
				class
				52
4.7	characters		distribution	53
			
4.8	characters		distribution	53
			

4.9	positive results very close to each other.....	54
4.10	Positive result in Huffman Coding checking.....	55
4.11	Password's results.....	negative 56

List of abbreviations

ARPA	Advanced Research Projects Agency
ASCII	American Standard Code for Information Interchange
BBN	Bolt, Beranak, and Newman
BBS	Bulletin Board System
CERN	Conseil Européen pour la Recherche Nucléaire
CHC	Convex Hull Click
DARPA	Defense advanced Research Project Agency
DNS	Domain Name System
FTP	File Transfer Protocol
HTML	Hyper Text Markup Language
HTTP	Hypertext Text Transfer Protocol
ICCB	Internet Configuration Control Board
MIT	Massachusetts Institute of Technology
NCSA	National Center for Supercomputing Applications
POP	Post Office Protocol
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol

UDP User Datagram Protocol

WWW World Wide Web

Chapter one Introduction

Life these days has become largely dependent on passwords for many purposes: logging into computer accounts, retrieving emails from servers, transferring funds, shopping online, accessing programs, databases, networks, web sites and even reading the morning newspaper online^[1].

The problem of selecting and using good passwords is becoming more important every day. The importance of services that are provided through computers and networks increases dramatically and in many cases such services require passwords or other forms of security concerns.

This chapter talks briefly about history of the internet , history of password , history of hacking , Applications , statement of the problem, previous password work, objectives of the thesis, and chapters' outlines.

1.1. History of the Internet

Back in the early 1960's, the internet was still unknown, computer networks weren't discovered, and no e-mail facility yet existed. So, when people were extremely busy to communicate, they would use the post, telephones or telegrams. Therefore, the birth of the internet has dramatically changed the role of all these mediums^[2].

The Internet represents one of the most successful examples of the benefits of investment and commitment to research and development of information infrastructure^[3]. Internet was the result of some genius ideas by scientists in the early 1960s that lead to a great potential value in allowing computers to share information on research and

development in scientific and military applications; the first proposed global network of computers was at Massachusetts Institute of Technology (MIT) in 1962. It moved over to the Defense advanced Research Project Agency (DARPA) in late 1962^[2]. In 1965 Thomas Merrill and Lawrence G. Roberts connected the TX-2 computer in Massachusetts Institute of Technology (MIT) to Q-32 in California with a low speed dial up telephone line creating the first wide area computer network^[4]. The result of this experiment was the realization that the time-shared computers could work well together, running programs and retrieving data on the remote machine, but the circuit switched telephone system was inadequate for the job. So Kleinrock was convinced of the Theoretical feasibility of communications using packets rather than circuits^[5]. The way Packet switching worked was by dividing a message up into small sized units called packets; each packet contains a unit of data as well as information about its origin, its destination, and the procedure to be followed to reassemble the message.

In 1968 Advanced Research Projects Agency (ARPA) awarded a contract to Bolt, Beranak, and Newman (BBN), a technology consulting firm, to build a testbed network called ARPANET. ARPANET connected only four computers: University of California at Santa Barbara, University of Utah, University of California At Los Angeles and SRI in Stanford^[6].

The development of packets switching pushed ARPANET to start growing and to connect 15 host together. New network overlapping protocols and systems were created such as Transmission Control Protocol (TCP) which enabled the expansion to a worldwide internet and the number of hosts to rise to 62 hosts.

To push the development of the internet, the Internet Configuration Control Board (ICCB) was changed to the internet Activities Board (IAB). Due of the fast growth of the internet (around 1000 hosts) the Domain Name System (DNS) was created and all hosts were connected to it.

The invention of the World Wide Web (WWW) had the most important impact to the Internet's growth and development and the first web browser was called "Nexus". National Center for Supercomputing Applications (NCSA) released the first fully featured graphical web browser called "Mosaic". Later the famous Netscape Navigator came and then Microsoft internet Explorer was created.

By 1996, the internet connected up to 13 million computers together in 195 countries, reaching all continents ^[7]. The internet has become an integral part of many lives in the industrialized world of the 21st century. The user can communicate with friends and family through e-mail and pass long text messages, documents, photographs and audio/video clips as well.

The internet has not just made life easier for the individual. Some of the features were popular with businesses eager to keep up with the global market place. It has revolutionized the way that many companies do business. The internet has contributed to the modern trend toward globalization, so that businesses no longer operate only locally or nationally, but internationally as well ^[30]. Since the internet is accessible to anyone with a computer and a network connection, individuals and organizations can reach any point on the network regardless to national or international boundaries or time. Therefore Internet should be secure and safe^[8].

1.2. History of Password

A password (watchwords) is a secret word or string of characters that is used for authentication, to prove identity or gain access to a resource, it has been used since ancient times, in Roman military; it was the way in which they secure the passing round of the watchword for the night. Passwords have been used with computers since the earliest days of computing. It was introduced in 1961 in the Compatible Time-Sharing System for The Massachusetts Institute of Technology (MIT). There was a LOGIN command that asks the user to type his password. The system then turns off the printing mechanism, so that the user can type the password with privacy. Robert Morris invented the idea of storing login passwords in a hashed form as part of the Unix operating system. His algorithm, known as crypt(3), used a 12-bit salt and invoked a modified form of the DES algorithm 25 times to reduce the risk of pre-computed dictionary attacks^[1].

1.3. History of Hacking

Access to internet can also pose hazards, as with most technological advances, there is also a dark side: criminal hackers. Government, companies, and citizens around the world are anxious to be a part of this reevaluation, but at the same time they are afraid that some hackers will break into their web server and replace their logos or read their e-mail, steal their credit card numbers from an online shopping site, or implant a software that will secretly transmit their organization's secrets to the open internet.

The real meaning of hacking is to expand the capabilities of any electronic device: to use them beyond the original intentions of the manufactures. As a matter of fact, the first hackers appeared in the 1960's at the

Massachusetts Institute of Technology (MIT), and the first victims were electric trains. They wanted them to perform faster and more efficiently^[9].

During the 1970's, a different kind of hacker appeared: the phone hackers or phreaks. They learned ways to hack the telephonic system and make phone calls for free. John Draper, found that he could make long distance calls with a toy whistle. He built a "blue Box" that, when used in conjunction with the whistle and sounded into a phone receiver, allows phreaks to make free calls^[10].

By the 1980's, phreaks started to migrate to computers, and the first Bulletin Board System (BBS) appeared. BBS are like the yahoo groups of today, were people posted messages of any kind of topics. The BBS used by hackers specialized in tips on how to break into computers, how to use stolen credit card numbers and share stolen computer passwords^[9].

In 1988, Robert Tappan Morris, a Cornell grad University student, created the first Internet worm; a destructive program that replicates itself and moves through a computer network at breakneck speed. He tried to show how an MIT security system was vulnerable to attack; he wrote a software program that exploited a glitch in a Unix email program and allegedly intended the worm program to infect only the MIT network. But during a 12-hour period, it spread rapidly, infecting thousands of systems and forcing some universities to shut down their computers altogether and to spend thousands of dollars to fix the infected ones^[11].

In 1991, The General Accounting Office reveals that Dutch teenagers gained access to Defense Department computers during the Gulf War, changing or copying

classified sensitive information related to war operations, including data of military personnel, the amount of military equipment being moved to the gulf and the development of important weapons systems.

In 1994, Two hackers identified as "Data Stream" and "Kuji" broke into Griffith Air Force base and hundreds of the systems, including computers at NASA and Korean Atomic Research Institute. And in 1999 a Norwegian group cracked a key to decoding DVD copy protection. The group created a DVD decoder program for distribution on the Web, a move that spurred a flurry of lawsuits from the entertainment industry^[12].

In 2000 Yahoo, eBay, Amazon, Datek and dozens of other high-profile Web sites were knocked offline for up to several hours following a series of so-called "distributed denial-of-service attacks." Investigators later discovered that the DDOS attacks in which a target system is disabled by a flood of traffic from hundreds of computers simultaneously were orchestrated when the hackers co-opted powerful computers at the University of California-Santa Barbara^[13].

Recently, being a hacker has taken on a new meaning – someone who breaks into systems for malicious intent. They are out for personal gain: fame, profit and even revenge. They Modify, delete, and steal critical information, often making other people miserable^[14].

To guarantee security of personal information and to prevent unauthorized access to important data, user accounts, such as computers and email accounts, passwords are an important security tool for any system. It is a strong lock that will protect all the information on personal and public sites.

1.4. Applications

The world of today is ruled by the internet, where everyone from individuals to institutions store their information on it^[15]. They realize that they need to hide some important information such as private correspondence , report draft , and account information. The best possible way to hide such information is to set up a password for its access.

Passwords are one of the important things for any system. It will help a user to maintain his/her identity so that others will not be able to view user's account.

Passwords have been used for a long time in many applications, such as logging in to computer accounts , e-mail , banks , shopping online, transferring funds, accessing program, database, networks , portals dating and social networking sites which all require passwords^[1].

1.5. Statement of the Problem

Due to the limitation of human memory, people are inclined to choose easily guessable passwords e.g. phone numbers, birthdays, names of family or friends, or words in human languages or dictionary words, which might lead to severe security problems. Though it was commonly believed that secure passwords were difficult to remember, easy-to remember passwords were insecure.

Passwords are considered weak when they are guessable or not resistant against dictionary driven attacks and, as a consequence, not secure in terms of brute force attacks.

Current password checking is considered the best approach for avoiding selection of weak passwords . Unfortunately it mainly depends on dictionary-based checking, and it often fail to filter some weak passwords from strong passwords. Meanwhile, using strong passwords reduce overall risk of a security breach; however, strong passwords don't replace the need for effective security controls.

The creation of an effective password can be through authentication system software that takes any hacking programs and techniques(like dictionary attacks software)into consideration , and that can be done by creating random password from four different groups (capital characters, small characters , digits and symbols) using Huffman Coding compressing algorithm.

1.6. Previous Work

In this thesis, there are many researches related to the history of passwords and password development, some of which are the following:

1.6.1. Password Security: A Case History (1979)

This article is an overview of the history of password security scheme on the UNIX time-sharing system. The UNIX system was the first to have a file that contained the actual passwords of all users. But that was insufficient; anyone could reach the password file, edit or modify it, even make a copy of this file.

It was necessary to devise a system in which neither the password file nor the password program can be read by anyone, and the only way to do this is to find means of

encryption that are very difficult to invert. The first convenient encryption method existed on an M-209 machine used by the U.S Army during World War II. The password was not used to be encrypted as the text but as the key, and a constant was encrypted using this key. The encrypted result was entered into the password file^[16].

1.6.2. Password Generation and Search Space Reduction (2009)

Random passwords can be produced by random password generators, but it might be difficult to remember, especially if someone has many accounts and many different passwords. People can easily create memorable passwords but they'll mostly be guessable. So experts generally recommend a system for evaluating each password against some metric and rejecting the weak ones, rather than mandating a certain number of characters from some character set, such as “use at least 2 digits, 2 lower case letters, 2 upper case letters and 2 special characters”, this might misleadingly guide users into designing passwords with the exact same number of characters and in the same order, such as 12asLK!?

Text based passwords are divided into three categories:

- Non-word passwords: character strings that don't contain any real words that are found in the dictionary, name, or location.
- Mixture passwords: character strings containing both word and none word part. Example: T!today65?
- Word passwords: strings which are pure dictionary words like (password) or modifications of them like (P@\$WORD).

The paper shows that in order to design good passwords, it must query to the following requirements:

- For non-word password design :
A password should be longer than 8 characters, and consist of all character sets, with varying the number of characters from each set.
- For mixture password design:
A password should be longer than 10 characters, using short words and many extra characters from a large character set.
- For word passwords:
A password should be more than 12 characters, having a lot of short and modified words, while using variation when modifying, and even using combinations of different languages^[17].

1.6.3. Password Protection?(2008)

Passwords are the gates of information, it's very important to make those gates very invisible so bad guys can't get through them. These days most of us are aware of the proper structure for a strong password, but the stronger the password is, the more likely one would forget it.

In case that happens, a procedure involves the use of "secret question" developed to retrieve the forgotten password, but this question that is supposed to help you remember should be carefully chosen, an answer for an easy question like 'what was your first employer? ' can be retrieved from one's profile on social networking sites such as Facebook, My Space and Live Journal, especially that most of us already have an account in at least one of these sites.

In this regard, it is highly advised to take extreme caution when it comes to the amount and type of personal information that you expose on these sites^[18].

1.6.4. An Image of the Future: Graphical Passwords (2006)

With text password getting more complex, it's becoming harder to remember them all especially when you have 10 – 20 different passwords.

Consequently the idea of graphical passwords came up to the surface, this type is image-based in which you should select specific areas in a picture in a certain order to unlock the machine. This method is quick and clean but also susceptible to shoulder surfing, so the icon- based password has emerged which it's impossible to be exposed by shoulder-surfers but remains too slow of a method, so it's unlikely to be used any time soon, it can be helpful in certain cases, though, where time of logging doesn't matter and keeping the password confidential is the top priority^[19].

1.6.5. Design and evaluation of a shoulder – Surfing Resistant Graphical password Scheme

The study first shows the advantages and disadvantages of both alphanumeric and image-based passwords. Disadvantages can include the exposure to visual hackers and the low speed of logging in. To avoid that, the Convex Hull Click (CHC) has been suggested. It is a graphical password scheme that guards against shoulder- surfing attacks by human observation, video recording or electronic capture. In CHC the graphical elements are shown in a window on the screen and the user must choose several icons from the portfolio to be his or her pass-icons.



Figure 1.1 Five pass – icons

She/he actually has to click within the convex hull of the pass-icons. A convex hull is the area encompassed by the edge joining a set of three or more points. The users won't click on the pass icons themselves and so the attacker cannot tell specifically which icons are the pass-icons.



Figure 1.2 Example of a convex hull with 5 pass – icons.

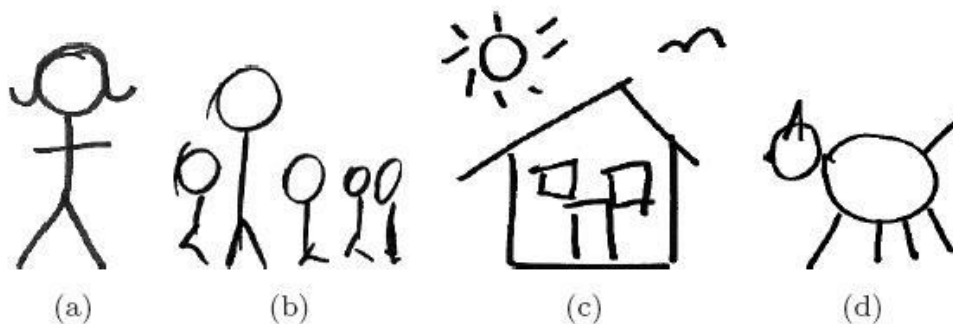
The password window changes between the rounds of the challenges. The non-pass-icons will move to new random positions in the window, and some of them randomly leave the window and a random number of new ones enter the window. If the user responds correctly to each of the challenges, he/she will be authenticated, otherwise the login fails.

The main consideration about the security of CHC is that the password space can be very large (and more secure) by increasing the number of icons, the number of pass icons, or both. And the brute force attack is infeasible. The main drawback for CHC is the longer time taken to input the password^[20].

1.6.6. Memorability of alternative password systems (2009)

The study shows a need to develop and evaluate new forms of memorable passwords, since the task of remembering alphanumeric passwords is often complicated by such factors as the need for multiple passwords to access different systems and the requirement to change passwords that expire at different times. This work investigates the efficacy of proposed new systems in the “real world” based upon theories of memory that involve self-generated meaningful symbols. It shows that whatever the nature of the meaning and its relationship to memory, a meaningful password is easier to learn and better remembered than a meaningless password. The meaningful password should be learnt after only one presentation and one trial of learning.

An experiment done on twenty participants to select 4 faces they thought they could remember from 36 faces images (these faces were selected on the basis of ethnicity with the absence of distinctive facial features) to form their face-based password. Participants were not told that they would be required to remember the faces in any particular order, and the time taken to select four faces was determined and recorded , then the participants were asked to draw four black and white symbols that hold personal meanings to them, and the time they took to decide and draw their four symbols was recorded. They weren't told to remember their symbols in any particular order and were not asked to explain their symbols.



Figur.1.3Example of a symbolic password. a=18month female child;b= family comprising mother, father and three sons; c=family home; d=family pet (dog).

The participants were also asked to generate a four-character alphanumeric password with the restrictions that it must have no simple repetition and wasn't a password they're currently using. The time taken to produce that was also recorded. Then, random four-character alphanumeric passwords were computer-generated and each participant was randomly assigned one of them.

Participants were tested on days 1,3,7,17,28 and 48 after the training. It was found that the time required to generate face and symbol based passwords is more than the time required to produce an alphanumeric password.

Since alphanumeric passwords are restricted to 62 characters, there are potentially an infinite number of symbols-based passwords, which is an important attribute of the overall success of a password system. File size is also important in determining the success of a password system. Alphanumeric passwords utilize the smallest file size followed by symbol-based passwords with face-based passwords requiring the largest.

Participants found it easier to self-generate an alphanumeric password than a symbolic password, but they enjoyed doing the symbolic password task more. Contrary to expectation, self-generated and meaningful symbol-based passwords were not the best remembered of the password type studied but may still be a viable overall alternative^[21].

1.6.7. Hybrid Password Authentication Scheme Based on Shape and Text (2010)

Hybrid password scheme can be an alternative solution for text-based and graphical passwords. The study explains the necessity to make a bridge between the graphical and text password, a grid with characters is adopted to construct the new system, and that is known as hybrid password authentication scheme which is based on the shape and text used in computers and mobile devices. With this new authentication scheme, users can only remember the shapes and strokes they like as their password. Then the system authenticates the shape of passwords just with text on the grid and their input order during the process.

If, for example, the user chooses one of the characters of his name, for example “N”, as the shape of the password, he/she has ability to design the stroke on the grid as he/she likes, and the shape is finally represented by a number of blocks on the grid.

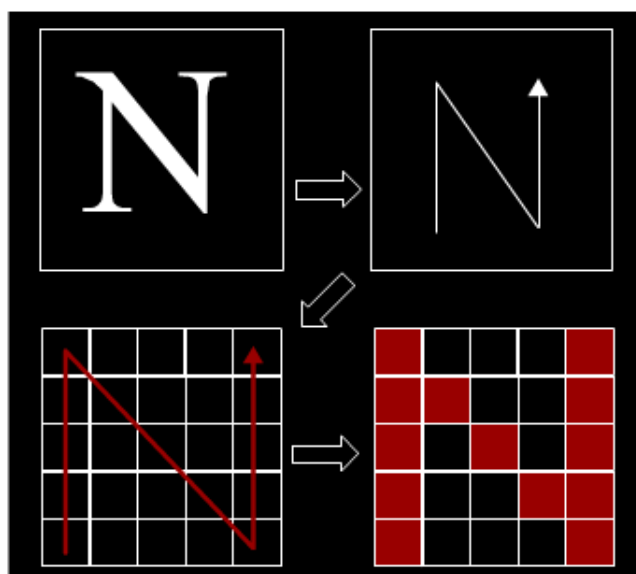


Figure 1.4 Password set procedure

The set of elements that produce this shape of password (referred to it as U) is either 0 or 1, so $U = \{0, 1\}$, and the system will choose the symbols randomly from U to fill every grid.

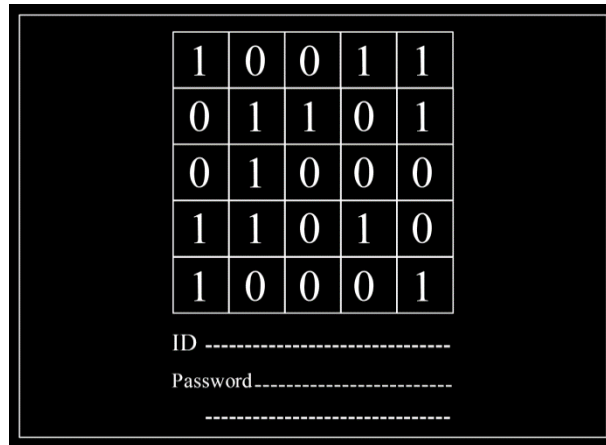


Figure 1.5 Login interface

Only "0" and "1" from keyboard keys will be used to input the password, so the right order of password symbols will be as follows: [1100110110011], if the password entered is not correct, then the system will generate another login interface grid, and the symbols from U appeared in the grid varies at each login step which means that the shape and the sequence of shape will not vary but the mapped text will not be the same in the new interface.

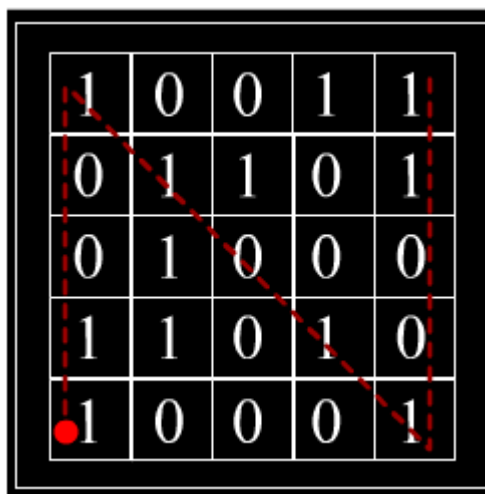


Figure 1.6 Original stroke on the interface

The advantages of hybrid password authentication scheme is that users cannot only choose the character but can also adopt the geometric shapes, the number shapes, the symbol shapes and even the arbitrary. One shape can even have different styles, which means a conceptual shape will generate various specific styles, take the triangle shape for example, the stroke of the shape can have several variants.

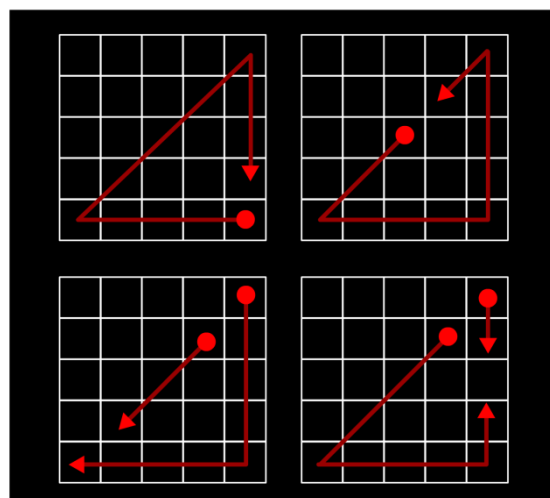


Figure 1.7 Stroke variants of triangle

Plus, hybrid password scheme is highly resistant to shoulder surfing because the login step does not reflect the password shape directly and does not require the user to click on the screen. However, the system still has some drawbacks. This method takes longer time than graphical or textual schemes, and it's relatively unfamiliar to the general public so the user may adopt the simplest and weakest strokes as their passwords. Also, if the password creation process has been recorded, the whole system can be attacked easily^[22].

1.7. Objectives of the Thesis

The main objectives of this thesis are:

1. Construct password generator software based on Huffman Code compressing algorithm.
2. Compare passwords with password strength checking (Password Meter).
3. Evaluate the performance and strength for a new system.

1.8. Outline of Forthcoming Chapters

Chapter two mainly describes internet security, passwords, passwords and Hacking, in more details. Chapter three is concerned with Huffman Coding, building a new system for password generators based on Huffman Coding. Chapter four lists experimental analysis and results. Finally chapter five is concerned with conclusion and future works.

Chapter Two Password

With the escalating popularity and usability of the Internet, it is only normal that issues like Internet security and internet crimes are being discussed , full description of password are also presented in this chapter.

2.1. Internet Security

Confidentiality, integrity and availability are the three fundamental concepts of internet security. When an unauthorized person reads or copies the information, it results in a failure of confidentiality.

When the information is modified in an unbalanced manner, it results in loss of integrity and when information is removed or becomes inaccessible, it results in loss of availability.

Authentication and authorization are the main procedures of internet security systems by which organizations make information available to those who want it and who can be trusted with it ^[23].

With the existence of the Internet, computer users can now work in an open system and security has become much more complicated. Also, several number of types of computer crimes may be susceptible if sufficient security measures are not in place.

2.2. Computer Crimes

a) Unauthorized Entry

Unauthorized entry is what happens when a criminal enters the company's computer system. The criminal uses the codes or passwords of an authorized user to gain illegal access to the system, the criminal could also search for points into a program or a network to gain access to the system.

b) Computer Viruses

Organizations' computer systems and networks can also be vulnerable to external attacks by computer viruses. These programs or pieces of codes are loaded onto the computer without the user's knowledge and against his/her wishes .altering the way that the computer operates or modifying the data or programs that are stored on these computers. Simple viruses can be self replicating bits of code that use up a computer's memory or otherwise disable a computer. More complex viruses can transmit themselves across a network and bypass the security system to infect other computers or systems.

c) Intentional Damage.

Data leakage is the intentional removal of files or even entire databases from a system without leaving any trace that they have been removed or even that they existed. Another type of intentional damage to the system's data is scavenging. It can be done through searching the trash can in the computer to find discarded data or other information about the system's programs or processes.

d) Stealing or Capturing Data

There are numerous examples of methods to stealing or capturing data in the enterprise's system. Eavesdropping is the use of electronic surveillance devices to either listen to or capture the content of electronic transmissions. Wiretapping is the use of any device to electronically capture data during transmission or to listen to conversations that take place over the network. Both wireless transmissions and those that occur over copper wire are susceptible to wiretapping ^[8].

2.3. Password

2.3.1 2.3.1.Type of Password:

A Password can either be textual or graphical or combining textual and Graphical.

1. *textual password:*

The password that consists of English letters, numbers and symbols. Text passwords have been widely used for user authentication. Human-generated text-based passwords can be divided into four categories: Non-word passwords, Mixture passwords, Word passwords and passcodes.

a) *Non – word passwords:*

character strings, which do not contain any real words that are found in the dictionary, names or locations, for example (NT*Ke0).

b) *Mixture passwords:*

Character strings containing both word and non-word parts. Example (T!today65?) this password has two non-word parts around the word part in the middle.

c) *Word passwords.*

Word passwords are strings which are either pure dictionary word(s). Example: (password,SkiingIsTheBestIKnow) or modifications of them, e.g. P@\$\$WORD [17].

d) *passcode*

The term passcode is sometimes used when the password is purely numeric, such as the personal identification number (PIN) commonly used for ATM access [1].

2. Graphical password

The ubiquity of graphical user interfaces and input devices, such as the mouse, stylus, and touch screen, which permit other than typed input, has enabled the emergence of graphical passwords. The main difference to textual passwords is the use of a device with graphical input, the user enters the password by clicking on a set of images, specific pixels in images, or by drawing a pattern in a pre-defined and secret order^[24].

Graphical password techniques can be largely classified into three categories: recognition-based, cued recall and recall based.

a) Recognition-based system

A series of images are presented to the user and a successful authentication requires the correct images being clicked in the right order.

b) Recall based system

The user is asked to reproduce something that he or she created or selected earlier during the registration.

c) Cued recall system

This is a component of a memory task in which the subject is asked to recall items that were presented to them on an initial training, or initial presentation list.

However, it is slightly different from recall based system because the subject is given a hint, or a cue, about the items on the original list^[32].

Graphical password schemes are divided into two major categories: image-based schemes and grid-based schemes.

1. Image based schemes

Image-based schemes use images, including photographs, artificial pictures, or other kinds of images as background. Based on the number of images displayed, image-based schemes are divided into two subclasses: single-image schemes and multiple-image schemes.

- **Single image schemes:**

Single-image based schemes use one single image as a background and require a user to repeat several actions with an input device, such as clicking or dragging, in the same manner as in the registration stage [26]. An example of this scheme is a PassPoint.

- **PassPoints:**

This is a famous graphical password scheme which allows arbitrary images to be used; a user can click on any place on an image to create a password. A picture contains hundreds to thousands of memorable points. So the possible password space is quite large [27].

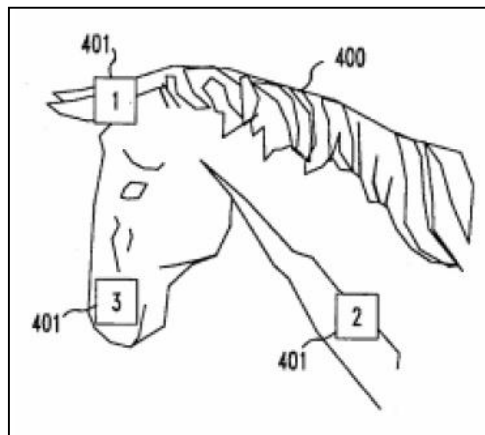


Figure 2.1 Graphical password scheme suggested by Blonder

In multiple-image schemes, multiple images are presented and a user is required to recognize and identify one or more of them, which are previously seen and selected by the user. There are two famous types of this scheme: face scheme (Passfaces) and Story scheme.

- **The Face Scheme:**

Passfaces is a commercial authentication product based on the graphical password. Users are given a random set of faces to serve as their secret authentication code. During the authentication process users should pick out their assigned faces, one at a time, from a successive group of nine faces.

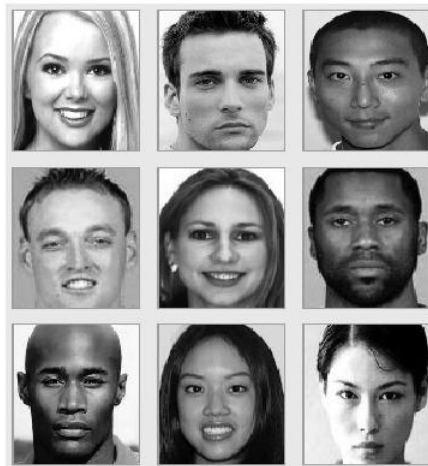


Figure 2.2 Passfaces

One problem with Passfaces is that some faces displayed might not be welcomed by certain users. In other words, if a user has to look at some faces he/she does not like or even dislike, the login process will become unpleasant. Another drawback of Passfaces is that it cannot be used by people who are face-blind (a disease which affects a person's ability to tell faces apart).

- **The story scheme:**

A password is a sequence of images selected by the user to make a story . the images are drawn from categories that depict everyday objects, food , automobiles , animals , children , sports , scenic locations and male and female models.



Figure 2.3 The story scheme

2. Grid-based schemes

Proposed by Jermyn et al in year 1999 [26]. led graphical passwords to a grid background. Users under this scheme can create their secret password by drawing their secret password as a free-form image on a grid .At login time, a user is required to draw the same pattern of image on the grid to gain access to the system. This algorithm involves storing the coordinates of grid cells where the user puts his pen down, draws a line and then lifts his pen up. Each pen up has a specific value. The bit string generated from the drawing is hashed using a one way hash function, and stored. This hash is then matched with the stored hash in order to authenticate the user[27]. Using a grid as background has several advantages:

First, it eliminates the need to store a graphical database on the server side and removes the overhead to transfer images through network.

Second, as a grid is a simple object, such schemes minimize the quality requirement for displays, which is an essential factor in image-based schemes.^[26]

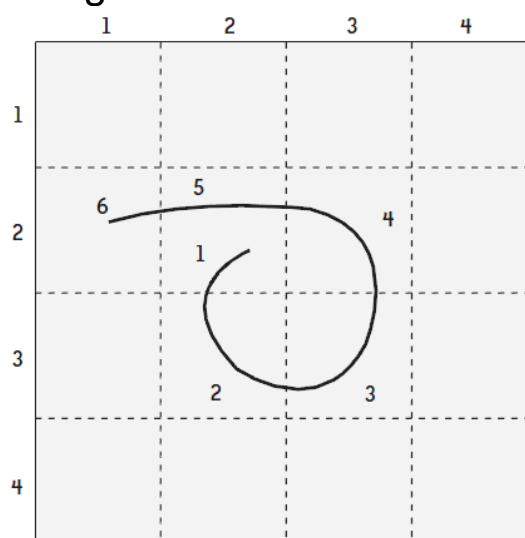


Figure 2.4 Password in Grid-Based Schemes

3. Text and Graphical Passwords

Combining text and graphical passwords can be done by following two steps: In step one a user is asked for her user name and text password.

In step two the user is presented with an image portfolio. The user must correctly select all images (one or more) preregistered for this account in each round of graphical password verification. If the user does not like a particular image portfolio, he may request a new one or upload her own images to be included in a portfolio.

Each image is associated with an index number, images along with their index numbers are displayed in a random order on the screen. To select an image the user identifies the image and then clicks the corresponding index number on the lower selection panel.

After the user completes all rounds of verification, if both the text password and all graphical passwords were correct, he is granted account access. Otherwise , access is denied^[25].



Figure 2.5 Selection panel in graphical step

and we shall focus , in this very study, on text-based password.

2.3.2. Store Password in Computer system

The computer system stores textual passwords using three basic methods: plaintext, encryption and hashes.

1. *Plaintext method*

The first and most obvious method that stores a password exactly as the user entered it. This plaintext method stores the plain data without any obfuscation, encryption or encoding. When the user log into a computer or a network account, the system compares the password entered with the copy stored in a database. If they match, it lets the user in.

2. *Encryption method*

In this method each password is encrypted before storing it in the database. Encryption combines plain text with another secret key to create a garbled string that can be retrieved only by using that same key. Encryption is just storing a password protected by a password.

3. *Hash method*

The widely accepted solution for storing passwords is to use password hash. A hash is the result of an algorithm in a complex formula that modifies plain text in a complicated manner to produce a garbled string that represents the password. Hashing algorithms are one way formulas because there is no reasonable way to calculate the original password from its hash. It can't just reverse the formula.

To check the user password, a computer system will take the user entry, run it through the same hashing algorithm, and then compare the result with the data stored in the hash database. If they match, the system knows that the two passwords must have the same hash to produce the same result.^[28]

2.3.3. Password Strength

Password strength is a measurement of the effectiveness of a password in resisting guessing and brute force attacks^[29]. To grade password strength , various basic tests must be run , which include a letter frequency analysis , a character type analysis , a length distribution analysis and a common password analysis^[30].

1. Letter frequency analysis:

In every language, some letters are used more often than others on average or some combinations of 2 or 3 letters appearing more often in a normal sentence.

The attacker gathers statistical information of the occurrences of each letter and uses this information to aid his guesswork when trying to break the cipher or the password.

For example "Robert Edward Lewand the author of Cryptological Mathematics book" analyzed roughly 15000 characters , or roughly 2700 words from three separate sources and found the following frequencies of English letters:

a	0.08167
b	0.01492
c	0.02782
d	0.04253
e	0.12702
f	0.02228
g	0.02015
h	0.06094
i	0.06966
j	0.00153
k	0.00772
l	0.04025

m	0.02406
n	0.06749
o	0.07507
p	0.01929
q	0.00095
r	0.05987
s	0.06327
t	0.09056
u	0.02758
v	0.00978
w	0.02360
x	0.00150
y	0.01974
z	0.00074
n	0.06749

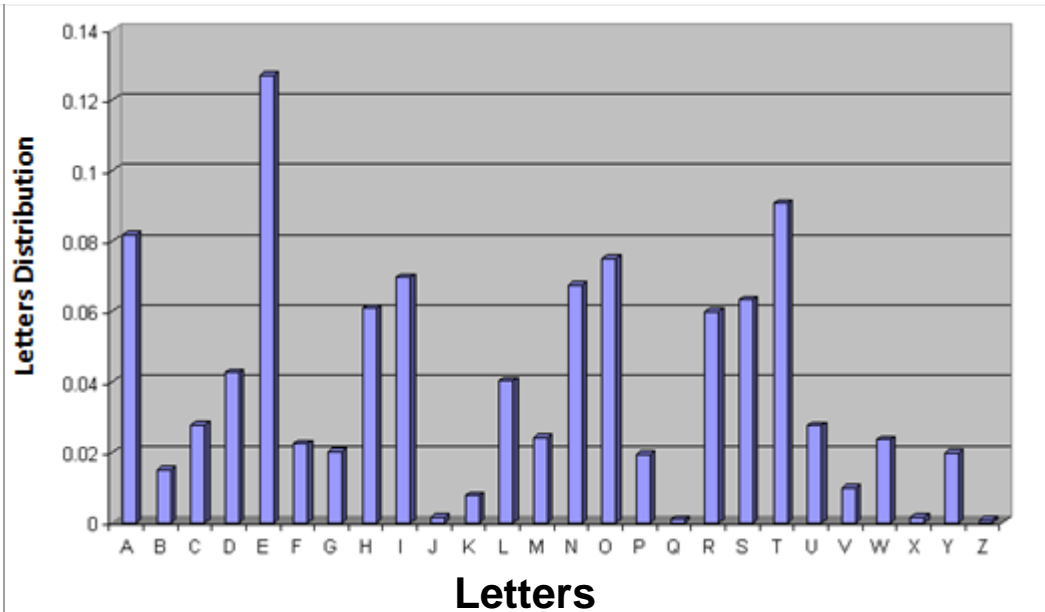


Figure 2.6 Frequencies of Letters^[31].

Also he analyzed combinations of 2 or 3 letters which appearing more often in a normal sentence , and found the following results:

The most common digrams	th	The most common trigrams^[32]	The
	he		And
	at		Tha
	st		Hat
	an		Ent
	in		lon
	ea		For
	nd		Tio
	er		Has
	en		Edt
	re		Tis
	nt		Ers

	to		Res
	es		Ter
	on		Con
	ed		Ing
	is		Men
	ti		The
	th		And
	he		Tha
	at		Hat
	St		Ent
	An		Ion
	In		
	Ea		
	Nd		
	th		

Figure 2.7 2 or 3 letters appearing more often in a normal sentence.

Letter frequency analysis helps us in various ways. Knowing the frequency of each letter gives ability for measuring password strength. Furthermore , letter frequency analysis allows us to measure the degree in which the passwords conform to actual words. It helps us to better understand if the passwords follow a language and what language this might be.

2. Character type analysis

Character type analysis looks at every password and flags what kind of characters make up the password, there are five categories of character types categories :

- Lowercase : which are the standard lowercase letters of the alphabet.
- Uppercase: which are the standard uppercase letters of the alphabet.

- Digits: which are the digits zero through nine.
- Symbols: which are any characters found in the non – extended ASCII set that do not belong to the above categories. Most of these can be found on a standard American keyboard.
- Unicode: which are any characters that do not belong to the above categories. Like the euro sign and the Japanese alphabet.

It is found that half of the passwords consist solely out of lowercase characters, this is troublesome, considering that most passwords are only 6 to 8 characters long and conform to a language. And also the passwords which consist solely out of digits are limited in complexity.

3. Length distribution analysis

Password length is simply the number of individual characters used in the creation of a password. Length distribution analysis gives us insight in what the common length of a user chosen password is. Many policies require a minimum password length, typically 8 characters. Some systems impose a maximum length for compatibility with other systems.

4. Common password analysis

A common password is a password that is widely used and most likely one that can be logically guessed.

Table 2.1 Top 10 password

	Password
1	Password
2	12345
3	Princess
4	1234567
5	Abc123
6	Monkey
7	Password1
8	(your first name)
9	Iloveyou
10	123456

In other words, if an attacker would try to gain access to a system by trying the top 10 most common passwords using a listing of known accounts . He/she could expect to succeed within 25 accounts, costing him only 250 guesses^[30].

2.3.4. Tools available for Password Strength Checking

Commercial tools available for password strength checking include the Password Meter (Password Meter , 2008) and Microsoft password checker (Microsoft , 2008) . These password meters use lexical rules.

The Password Meter which used by Google is a Java Script Function That checks the strengths of passwords with a well-defined algorithm.

It is based on dealing with a weighting method, and a weight is adopted for computing the strength of the password.

The strength is decided based on the overall score which is determined using positive and negative weightages based on the scheme given in Table 2.2, the final score is capped with minimum of zero and a maximum of 100. The features that make the password strong are given more weightage and the features that weaken the password are given negative weightage.

Table 2.2 Scheme of weights assigned

Additions	Weight Assigned
Number of characters in the password	Number of characters*4
Number of Lowercase characters	(length – number of lowercase characters) * 2
Number of Uppercase characters	(length – number of lowercase characters) * 2
Number of digits	(number of digits * 4)
Number of symbols	(symbolcount * 6)
Number of Middle number /symbols	(number/symbolcount * 2)
Deductions	
Characters only	- 1 * number of characters
Digits only	- 1 * number of digits
Number of repeated characters (n)	- (n (n –1))
Number of consecutive uppercase characters (n)	- (n * 2)
Number of consecutive Lowercase characters (n)	- (n * 2)
Number of sequential characters	- (n * 3)

Requirements (n) 1. Minimum 8 characters in length 2. Contains 3/4 of the following items: - Uppercase Letters - Lowercase Letters - Numbers - Symbols	- (n * 2)
--	-------------

The final score is the cumulative result of all bonuses and deductions, and the final score is capped with minimum of 0 and a maximum of 100.

The password strength analysis has been carried out using SVM ^{light}. It is a training algorithm for learning classification and regression rules from data. SVM is most suitable for working accurately and efficiently with high dimensionality feature space. SVM is based on strong mathematical foundations and results are simple and very powerful algorithms^[29].

Table 2.3 Password Classification for the final score

Class	Score
Very Weak	Less Than 20
Weak	20 – 39
Good	40 – 59
Strong	60 – 79
Very Strong	Greater than 80

When the researcher implemented the previous features at the various passwords ,we got the following result^[33]:

1.Password: ghadeer

Test Your Password		Minimum Requirements
Password:	●●●●●●●	<ul style="list-style-type: none">• Minimum 8 characters in length• Contains 3/4 of the following items:<ul style="list-style-type: none">- Uppercase Letters- Lowercase Letters- Numbers- Symbols
Hide:	<input checked="" type="checkbox"/>	
Score:	7%	
Complexity:	Very Weak	

Figure 2.8 Password Meter program

2.Password: ghadeer5

Test Your Password		Minimum Requirements
Password:	●●●●●●●●	<ul style="list-style-type: none">• Minimum 8 characters in length• Contains 3/4 of the following items:<ul style="list-style-type: none">- Uppercase Letters- Lowercase Letters- Numbers- Symbols
Hide:	<input checked="" type="checkbox"/>	
Score:	33%	
Complexity:	Weak	

Figure 2.9 Password Meter program

3.Password: ghadeer555

Test Your Password		Minimum Requirements
Password:	●●●●●●●●●	<ul style="list-style-type: none">• Minimum 8 characters in length• Contains 3/4 of the following items:<ul style="list-style-type: none">- Uppercase Letters- Lowercase Letters- Numbers- Symbols
Hide:	<input checked="" type="checkbox"/>	
Score:	42%	
Complexity:	Good	

Figure 2.10 Password Meter program

4.Password: ghadeer^555

Test Your Password		Minimum Requirements
Password:	●●●●●●●●●	<ul style="list-style-type: none">• Minimum 8 characters in length• Contains 3/4 of the following items:<ul style="list-style-type: none">- Uppercase Letters- Lowercase Letters- Numbers- Symbols
Hide:	<input checked="" type="checkbox"/>	
Score:	64%	
Complexity:	Strong	

Figure 2.11 Password Meter program

5.Password: **GHras^NG^35**

Test Your Password		Minimum Requirements
Password:	<input type="password" value="....."/>	<ul style="list-style-type: none">• Minimum 8 characters in length• Contains 3/4 of the following items:<ul style="list-style-type: none">- Uppercase Letters- Lowercase Letters- Numbers- Symbols
Hide:	<input checked="" type="checkbox"/>	
Score:	83%	
Complexity:	Very Strong	

Figure 2.12 Password Meter program

Microsoft password checker, is a program that measures the password strength online, the user just types his password and gets an instant strength rating: Weak, Medium, Strong, or Best.

1.Password: **Ghadeer**

Password:

Strength: **Weak**

Figure 2.13 Microsoft password checker program

2.Password: **ghadeer5**

Password:

Strength: **Weak**

Figure 2.14 Microsoft password checker program


3.Password: **ghadeer555**

Password:

Strength: **Weak**

Figure 2.15 Microsoft password checker program

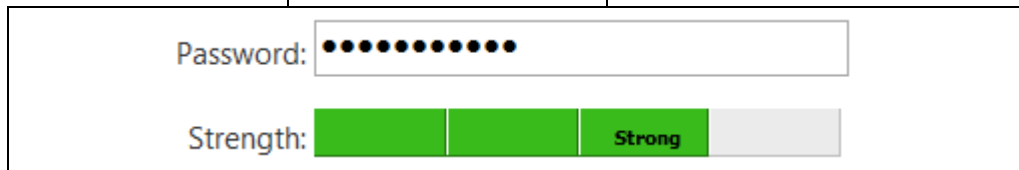
4.Password: **ghadeer^555**



The screenshot shows a password input field containing 'ghadeer^555'. Below the field, the password is represented by ten black dots. A strength indicator bar is shown below the dots, divided into four segments: the first two are yellow and labeled 'Medium', and the last two are grey.

Figure 2.16 Microsoft password checker program

5.Password: **GHras^NG^35**



The screenshot shows a password input field containing 'GHras^NG^35'. Below the field, the password is represented by ten black dots. A strength indicator bar is shown below the dots, divided into four segments: the first three are green and labeled 'Strong', and the last one is grey.

Figure 2.17 Microsoft password checker program

Password Checker does not collect, store, or transmit information beyond the computer that the user uses to access Password Checker. The image works on user computer desktop until the user navigates away from the page.

The security of the passwords entered into Password Checker is similar to the security of the user password entered when the user logs into Windows. The password is checked and validated on your computer, but is not sent over the Internet.

2.3.5. Password and Hacking

Attackers use online and offline methods to gain the user password, such as Smart Guesses, Dictionary Attacks , Brute Force Attacks , Rainbow Tables Social Engineering and Other Techniques.

1. **Smart Guesses**

The easiest method to gain user password is simply to guess it, many hackers simply try the five most common passwords for a particular system. They might also try a blank password or a password that is the same as the users' names or their children, spouse, pet or car model as

their password. If they get nothing they just move on the next account and keep trying until they find the account with weak passwords. These methods work by attempting them on large numbers of accounts. Hackers often use automated tools that allow for large scale attacks. ^[34]

Guessing password can be either done through an online attack or Offline Attack.

a) Online password attacks

The most straightforward password attack is online guessing. The attacker finds himself sitting in front of a login prompt, knows one or more legal login names, and begins trying to guess the passwords for these login names. If the attacker is familiar with the owner of the account he is trying to crack, he may be able to narrow the search space of possible passwords.

The attackers look specifically for accounts that have rarely been used. These accounts may still be using default passwords that have never been changed by the account's owner. The word "password" is commonly used as a default password.

But online guessing attacks have obvious problems. A common problem is that many accounts are automatically disabled after an incorrect password is entered five times in a row.

Another problem with password guessing is that it is a slow process. Also guessing attacks are susceptible to logging by a system administrator, unless the attacker is successful in compromising the system and then cleaning these logs, the system administrator will know the system has been attacked. The administrator will take the right action to the machine and the rest of the network.

b) Offline Password Attacks

Offline guessing of password is another approach to cracking passwords. Some Unix systems store the encrypted version of every user's password in a word readable file.

Usually the users choose simple, easily guessed passwords; the only potential obstacle is to acquire the password file for the target system. The attacker then feeds the password file to the latest version of his favorite password-cracking program, which will output the login name and associated password for each password it successfully cracks.

2. Dictionary attacks

Dictionary attacks are usually offline attacks against a password, with this attack, the attacker will use a program that will try every possible words in the dictionary. Dictionary attacks can be done either by repeated logging into systems, or by collecting encrypted passwords and attempting to find a match by similarly encrypting all the passwords in the dictionary. Attackers usually have a copy of the English dictionary as well as foreign language dictionaries for this purpose. They all use additional dictionary like database, such as names and lists of common passwords ^[35].

Many software tools are available to automate dictionary attacks against various systems. Most of these tools are smart enough to try simple variants of dictionary words, such as words followed by one or two numbers or simple letter substitutions.

3. Brute force attacks

Brute force attacks are more tedious but more complete versions of dictionary attacks. Brute force attacks also involve trying millions of passwords, but they work by trying every combination of every letter and every punctuation symbol until a password is found^[34]. A short 4-letter password consisting of lower case letters can be cracked in just few minutes. A long 7-character password consisting of upper and lower cases as well as numbers and punctuation can take months to crack assuming trying a million combinations in a second^[43]. Brute force attacks are slow and time consuming, but still quite common.

4. Rainbow Tables

Offline attacks work by hashing millions of passwords in order to find hashes that match those of the target. Rainbow Tables facilitate these attacks by pre-computing the hashes for billions of passwords. These tables take a very long time to generate, but once the attacker has the tables, he can crack a large number of passwords in a matter of seconds.

5. Social Engineering

Sometimes a hacker can get the password simply by asking for it. This technique is still quite effective.

Hackers might pose as help desk or support staff and try to trick the user into revealing the password. They might send to a user an e-mail claiming that the user account in some sites is suspended, providing a place to enter his password. Or they might even take advantage of user greediness by providing some trick to get rich quick or to take advantage of other users.^[34]

Chapter Three Huffman Coding

3.1 Huffman Coding

The standard model of storing data uses fixed length codes, the most common way to represent characters and numbers in computing is by using the ASCII Code. This is based on a string of 8 bits where each bit can be either '1' or '0'.

Table 3.1 Characters in ASCII Code

Character	ASCII Code
A	01000001
B	01000010
C	01000011

1	00110001
2	00110010
3	00110011

There is a modern model which allows a much wider range of language to be represented in Unicode. Unicode allocates 16 bits to each character and any ASCII character can be converted to Unicode by prefixing it with the zero byte.

Table 3.2 Characters in Unicode Code

Character	ASCII Code
A	0000000001000001
B	0000000001000010
C	0000000001000011

1	000000000110001
2	000000000110010
3	000000000110011

There are certain advantages to these two systems. When reading a file, it always reads 8 bits or 16 bits at a time to read a single character. But these coding schemes have disadvantage. because some characters are more frequently used than other characters^[37]. For example if we have a data file, and this file contains only 8 characters, appearing with the following frequencies in Table 3.3:

Table 3.3 frequency of Characters in a data file

Character	C	D	E	F	K	L	U	Z
Frequency	32	42	120	24	7	42	37	2

The actual message length for data file contains 8 characters is 918 bits ^[38].

So, using Huffman Code can find the optimal way to take advantage of varying character frequencies in a particular file. Using Huffman coding on files can shrink them anywhere from 10% to 30% depending on the character distribution.

The idea behind the coding is to give less frequent characters and groups of characters longer codes. Also the coding is constructed in such a way that no two constructed codes are prefixes of each other ^[37].

Huffman coding was developed by David A. Huffman while he was a Ph.D. student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes"^[39]. The Huffman method is somewhat similar to the Shannon-Fano method, proposed independently by Claude Shannon and Robert Fano in the late 1940s. The main difference between the two methods is that Shannon – Fano constructs its codes from top to bottom and the bits of each codeword are constructed from left to right, while Huffman constructs a code tree from the bottom up, and the bits of each codeword are constructed from right to left.

Huffman made significant contributions in several areas, mostly information theory and coding, signal designs for radar and communications, and design procedures for asynchronous logical circuits^[40].

The easiest way to see how a Huffman algorithm works is to work through previous example shown in table 3.3:

- The first step is to construct a priority Queue and insert each frequency – character pair into the queue.

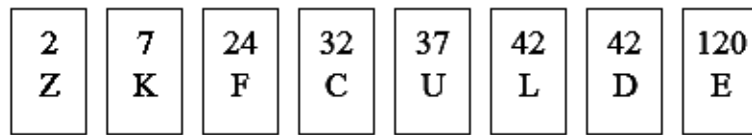


Figure 3.1 building Huffman coding tree

- In the second step, the two items with the lowest key values are removed from the priority queue.
 - A new binary Tree is created with the lowest key item as the left external node, and the second lowest key item to the right external node.
 - The new tree is then inserted back into the priority queue.

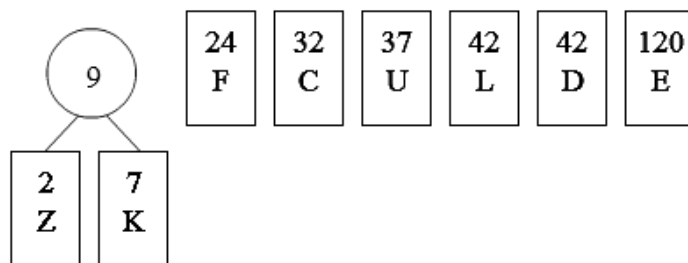


Figure 3.2 building Huffman coding tree

- The process is continued until only one node is left in the priority queue.

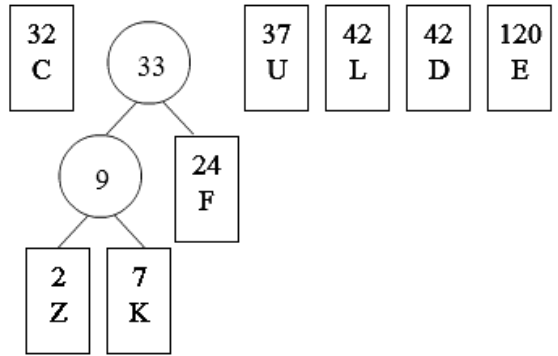


Figure 3.3 building Huffman coding tree

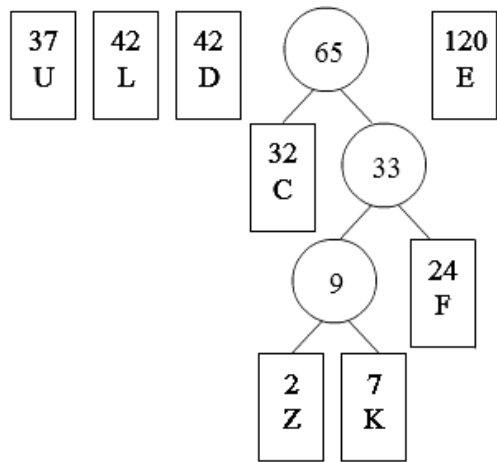


Figure 3.4 building Huffman coding tree

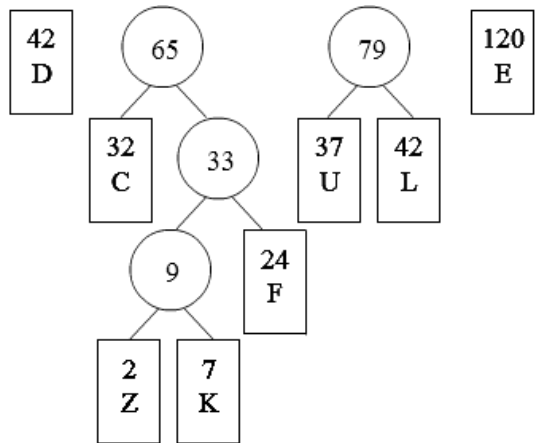


Figure 3.5 building Huffman coding tree

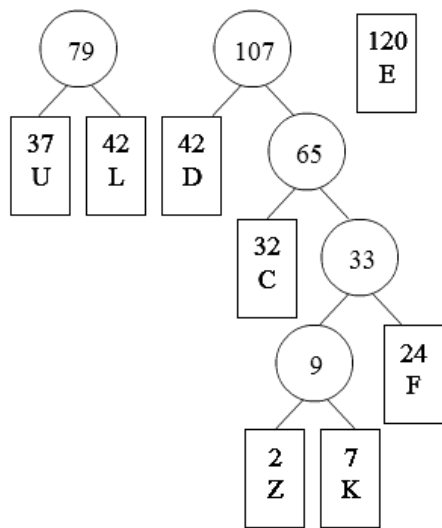


Figure 3.6 building Huffman coding tree

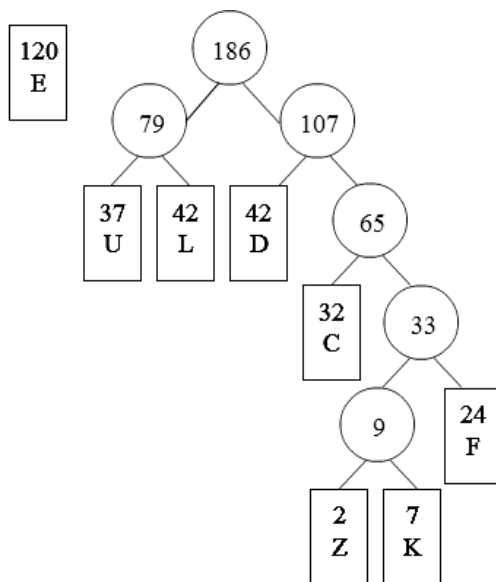


Figure 3.7 building Huffman coding tree

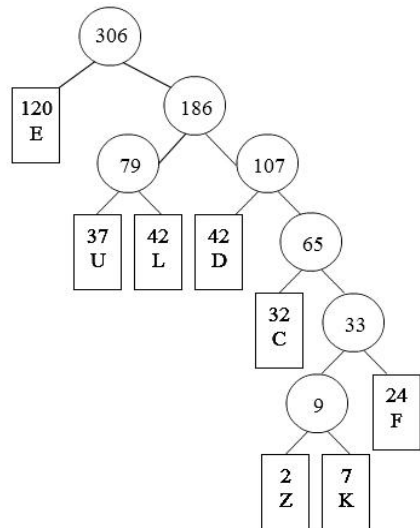


Figure 3.8 building Huffman coding tree

Huffman Algorithm (X):

Input : String X of length n.

Output: Coding Tree For X.

Compute frequency $f(c)$ of each character c in X.

Initialize a priority queue Q.

For each character c in X do

Create a single – node tree T storing c .

Insert T into Q with key $f(c)$.

While $Q.size() > 1$ do

$F1 \leftarrow Q.minkey()$

$T1 \leftarrow Q.removeMinElement()$

$F2 \leftarrow Q.minKey()$

$T2 \leftarrow Q.removeMinElement()$

Create a new tree T with left subtree T1 and right subtree T2.

Insert T into Q with Key $F1 + F2$

Return $Q.removeMinElement()$ ⁽⁴⁵⁾

Figure 3.9 Huffman Coding algorithm.

Once the tree is built, each leaf node corresponds to a character with a code. To determine the code for a particular node, walk a standard search path from the root

to the leaf node in question. For each step to the left, append a (0) to the code and for each right append a (1). As shown in the final Huffman Coding tree.

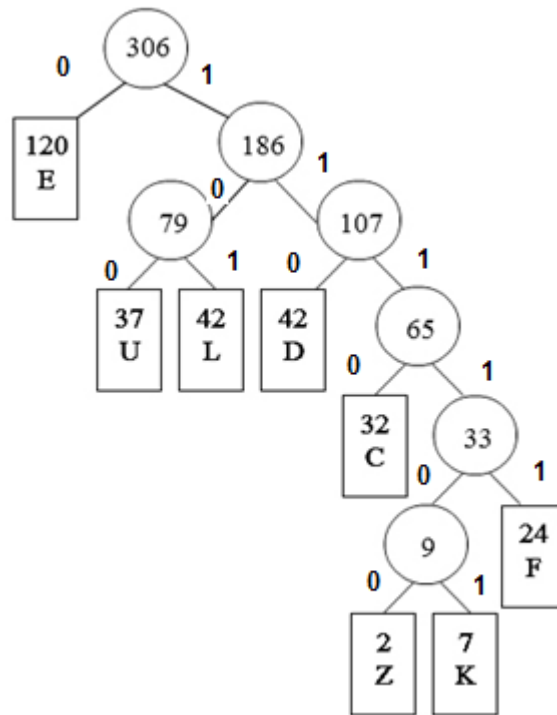


Figure 3.10 Final Huffman coding tree

Thus for the tree above we get the following codes^[37].

Table 3.4 Huffman Final table

Character	frequency	Code	No of bit
C	32	1110	4
D	42	110	3
E	120	0	1
F	24	11111	5
K	7	111101	5
L	42	101	3
Z	37	111100	6
U	2	100	3

When we're talking about Huffman Code there are some definitions that must be introduced.

1- **Probability for each symbol (Pi)** : the frequencies of occurrences of the symbols in the input string,^[40] and it can be found by the following formula:

$$P(i) = x = \frac{\text{frequency of symbol}}{\text{total number of frequencies}}, \text{ Where } \sum p_i = 1$$

2- **Codeword**: sequence of bits representing a coded symbol or string, for example in the previous example the Codeword for letter “C” is 1110.

3- **Average length**: the average length of symbols in bits. and it can be found by the following formula:

$$L_i = \sum p_i * \text{no of bit in symbol.}$$

4- **Entropy**: smallest number of bits needed on average to represent a symbol^[41] or the minimum number of bits to represent given symbols^[42].

Entropy for a set of character s1, s2, s3...sn with probability p1, p2...pn = $H(s1...s2) = - \sum p_i \log_2 p_i$

Huffman coding is optimal according to information theory when the probability of each input symbol is a negative power of two (2^{-1}).

Table 3.5 Final result

Character	frequency	probability	Codeword	No of bit
C	32	0.104	1110	4
D	42	0.137	110	3
E	120	0.392	0	1
F	24	0.078	11111	5
K	7	0.029	111101	5
L	42	0.137	101	3
Z	37	0.121	111100	6
U	2	0.007	100	3

The entropy and average size for the previous example is:

$$\text{Entropy} = - \sum p_i \log_2 p_i$$

$$\begin{aligned} \text{Entropy} = & - 0.104 * \log_2 * 0.104 - 0.137 * \log_2 * 0.137 - \\ & 0.392 * \log_2 * 0.392 - 0.078 * \log_2 * 0.078 - 0.029 * \log_2 * 0.029 \\ & - 0.137 * \log_2 * 0.137 - 0.121 * \log_2 * 0.121 - 0.007 * \log_2 * \\ & 0.007 = 2.51 \text{ bit/symbol.} \end{aligned}$$

The average size (avg) = $L_i = \sum p_i * \text{no of bit in symbol}$
 (avg) = $4 * 0.104 + 3 * 0.137 + 1 * 0.392 + 5 * 0.078 + 5 * 0.029 +$
 $3 * 0.137 + 6 * 0.121 + 3 * 0.007 = 2.11 \text{ bit/symbol.}$

Huffman code is not unique, which means we can get different Huffman codes for the same symbols. Suppose we have the following symbols with its probability:

Table 3.6 Characters and its probabilities

Character	Probability
A1	0.4
A2	0.2
A3	0.2
A4	0.1
A5	0.1

The same five symbols can be combined differently to obtain a different Huffman codes:

Table 3.7 Huffman coding result (1)

Letter	Probability	Codeword	No of bit
A1	0.4	0	1
A2	0.2	10	2
A3	0.2	111	3
A4	0.1	1101	4
A5	0.1	1100	4

The average size of table 3.7 is = $1*0.4+2*0.2+3*0.2+4*0.1+4*0.1=2.2$ bitsymbol.

Table 3.8 Huffman coding result (2)

Letter	probability	Codeword	No of bit
A1	0.4	100	3
A2	0.2	101	3
A3	0.2	00	2
A4	0.1	01	2
A5	0.1	11	2

The average size of table 3.8 is= $3*0.4+3*0.2+2*0.2+2*0.1+2*0.1=2.2$ bit/symbol

The best Huffman code is the one with the smallest variance. The variance of a code measures the sizes of the individual codewords which deviates from the average size. The variance of the code of table (3.7) is:
 $0.4(1-2.2)^2 + 0.2(2-2.2)^2 + 0.2(3-2.2)^2 + 0.1(4-2.2)^2 + 0.1(4-2.2)^2 = 1.36$

While the variance of the code of table (3.8) is :
 $0.4(3-2.2)^2 + 0.2(3-2.2)^2 + 0.2(2-2.2)^2 + 0.1(2-2.2)^2 + 0.1(2-2.2)^2 = 0.4^{[40]}$.

3.2 Implement Huffman Coding on Passwords.

Humans can easily create memorable passwords, such as meaningful words, names of the user's family members, spouse, or pet, but this also creates the problem that their generation process is guessable, especially by Attackers methods such as Smart Guesses, Dictionary Attacksand Brute Force Attacks.Passwords which are difficult to remember will reduce the security of a system because the user might need to write down or electronically store the password, therefore the password must be

meaningless and at the same time easy to remember.

The researcher is going to introduce an effective software that generates a password for the user by the following steps

Step no (1): the user is going to enter his/her six Characters from character four groups (Lowercase, Uppercase, digit and symbol).



Enter 6 Characters (Capital, Small, Digit and Symbol)
to generate your password

Generate

Dont repeat characters

Figure 3.11 Password generator program interface

Step no (2): The program starts generating the password according to the following algorithm:

Password Generator Algorithm:

```
begin
proc findPermutations(elemints:Array, len:int)
    permutationsNum=Math.pow(elements.length, len));
    // elements=8; and len=8 then the result is
    8^8=16777216 word
    check();
end

begin
proc check()
    permutations:Array;
    while permutations.length<permutationsNum
        perm:Array;

        while ( perm.length<len )
            ind:int = Math.random() *
elements.length;
```

```

perm.push(elements[ind]);

permstr:String=perm.join(',')// to create
single letters to on word
do if (permstr is not in permutations)
permutations.push(permstr);
end

```

Figure 3.12 Password Generator Algorithm

Step no(3): Findpasswords' strength byusing Huffman Coding Checking algorithm:

Huffman Coding Checking Algorithm.

1. Find Huffman Coding for each password.
2. Calculate the entropy for each password according to formula:
Entropy = - $\sum p_i \log_2 p_i$
3. Create five Huffman Coding Classification (Very Strong, Strong, Good , Weak , Very Weak)
4. Distribute each password according to its entropy among these five groups.
5. Save the first 20 results from each Group.

Figure 3.13 Huffman Coding Checking Algorithm.

Huffman Coding Classification created according to the following Scores:

Table 3.9 Huffman Coding General Classifications

Class	Score
Very Weak	2.21 – 2.5
Weak	1.91 – 2.20
Good	1.61 – 1.90
Strong	1.31 – 1.60
Very Strong	1.00 – 1.30

Step no (4): Generated passwords which obtained from figure (3.13) will be checked again according to Password Meter checking schemas.

Password Meter Checking Algorithm:

1. Find the weight for each password by computing the overall score which is determined in schemes:
 - Number of characters in a Password.
 - Number of lowercase characters.
 - Number of uppercase characters
 - Number of digits
 - Number of symbols.
 - Number of middle number/symbols.
 - Character Only
 - Digit Only
 - Number of repeated Characters
 - Number of consecutive uppercase characters
 - Number of consecutive lowercase characters.
 - Number of sequential characters.
 - Requirements
2. Create five Password Meter Classifications (Very Strong, Strong, good, Weak, Very weak).
3. Distribute each password according to its weight among these five groups.
4. Save the first 20 results from each group.

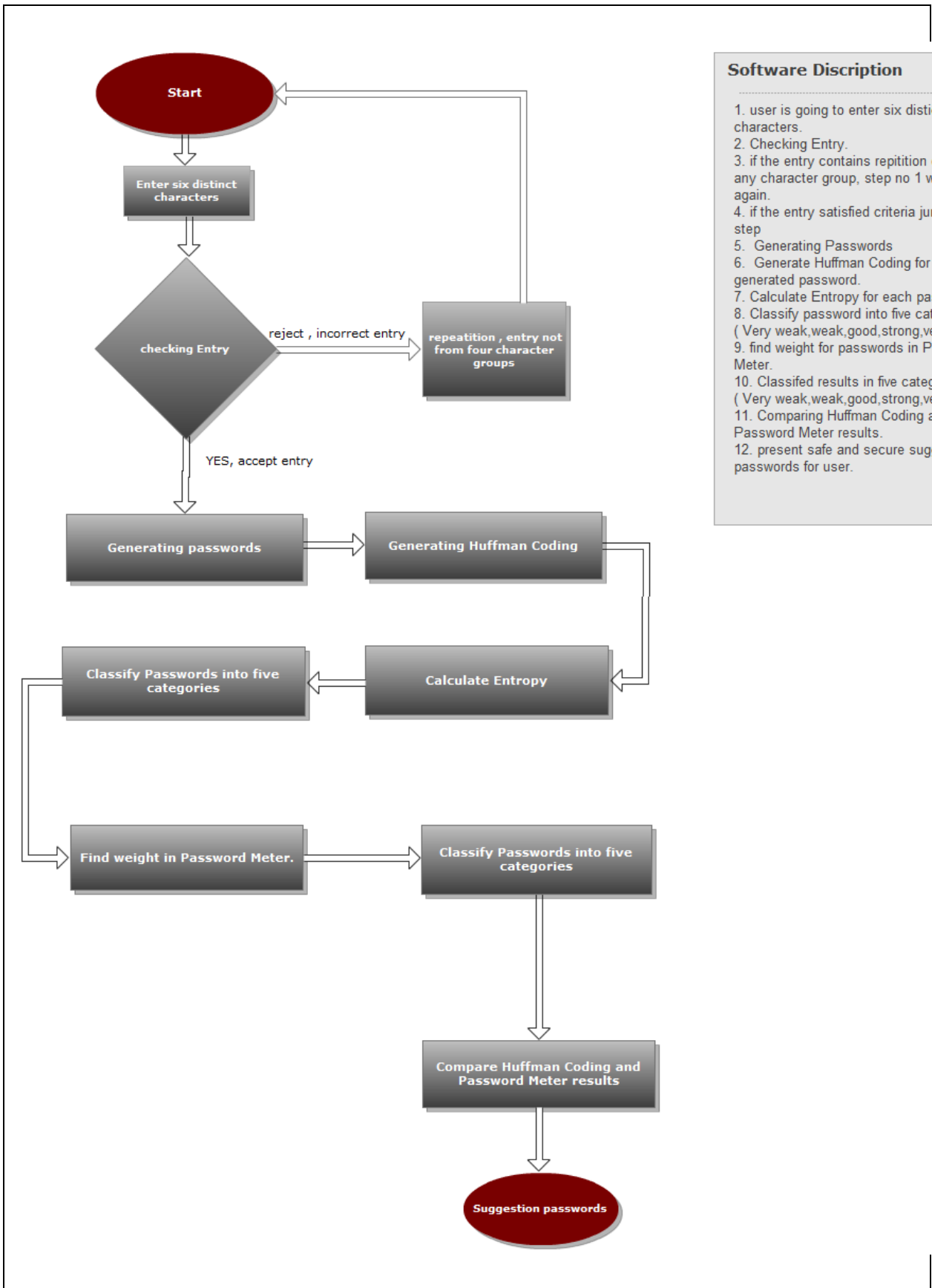
Figure 3.14 Password Meter Checking Algorithm
Password Meter Classification created according to
the following Scores:

Table 3.10 Password Meter General Classifications

Class	Score
Very Weak	Less Than 20
Weak	20 – 39
Good	40 – 59
Strong	60 – 79
Very Strong	Greater than 80

Step no (5): Comparing results obtained from Huffman Coding and Password Meter.

Step no (6): Present safe and secure suggestion passwords to user.



Software Discription

1. user is going to enter six distict characters.
2. Checking Entry.
3. if the entry contains reptition of any character group, step no 1 will be repeated again.
4. if the entry satisfied criteria jump to step 5.
5. Generating Passwords
6. Generate Huffman Coding for each generated password.
7. Calculate Entropy for each password.
8. Classify password into five categories (Very weak,weak,good,strong,very strong).
9. find weight for passwords in Password Meter.
10. Classified results in five categories (Very weak,weak,good,strong,very strong).
11. Comparing Huffman Coding and Password Meter results.
12. present safe and secure suggestion passwords for user.

Figure 3.15 Software processes

Chapter Four Result and Discussion

Results obtained from the new generated password system are presented in this chapter. Analysis and related curves are also discussed here.

4.1 Experimental Results

The researcher entered six distinct characters from four different groups (capital characters, small characters, digits, and symbols) the program generated (6777216) different passwords.



Figure 4.1 Password generator interface

The password strength was checked using Huffman Coding Checking algorithm , The researcher selected the first 100 passwords and did the analysis.

Table 4.1 Huffman Coding results

Very Strong	Result	Strong	Result	Good	Result	Weak	Result	Very Weak	Result
Z!!!ZIZ Z	1	ZS3S ZSSZ	1.4 1	Aww ww!S A	1.7 5	ZAAw 3AAS	2.0 0	SZ33 SAZw	2.2 5
AASA SSSA	1	ZAA!!! !A	1.4 1	SZZA 3ZZS	1.7 5	SA!Z ZS!A	2.0 0	SZAA S!3Z	2.2 5
w!!ww w!!	1	AAww Aw3w	1.4 1	ASwS S!S!	1.7 5	S3w!! wwA	2.1 6	ASw3 3wAZ	2.2 5
SS!!S S!!	1	3www 333!	1.4 1	w!AA ZAA!	1.7 5	SS3Z wZZA	2.1 6	w33w !SSA	2.2 5
WwwS SSSw	1	!33S!! 33	1.4 1	!ww!w wZ3	1.7 5	!wS33 A3w	2.1 6	S!3Z3 SAZ	2.2 5
!Z!ZZZ !!	1	ZwwZ ZwZA	1.4 1	AAAA !!3S	1.7 5	!ASZ wSAS	2.1 6	!ZSA ZS3A	2.2 5
ASSS AASA	1	AA!3!! A!	1.4 1	!SSZ SSA!	1.7 5	3A!3w !Z!	2.1 6	!ZZ!S wAS	2.2 5
w!w!w w!!	1	wSAA wSAA	1.5 0	wAAA ZSAZ	1.7 5	wASw w!3!	2.1 6	!!3A3 wwS	2.2 5
Z!Z!Z! Z!	1	33S!3! 3S	1.5 0	SSwS AAS!	1.7 5	ASAS Z3Aw	2.1 6	!SZA 3!SZ	2.2 5
ZZZA ZAAA	1	A333 ASS3	1.5 0	wwA w!3A w	1.7 5	S3w! wZ!!	2.1 6	ZAww A!ZS	2.2 5
A!AA! A!!	1	ZwZZ wZSS	1.5 0	3ZAA ZA!A	1.7 5	AS33 Zwww	2.1 6	ASZ! AZ33	2.2 5
!ww!!w !w	1	!AAw! w!!	1.5 0	3Zw!! Z!!	1.7 5	ZZSw AwZ3	2.1 6	ZSA! 3AAw	2.4 1

ZZZ!Z!! !	1	ZZS!!Z SZ	1. 50	AZZAA !AS	1. 75	AZ3SS Sw3	2. 16	wS3S ZA!S	2. 41
3!33!!3!	1	Z3SZZ SZ3	1. 50	wSAA AA3S	1. 75	3wSA3 !!!	2. 16	!3AZS w33	2. 41
3S3SS S33	1	AAZZ3 AA3	1. 50	www!! wAS	1. 75	S!S3w SZw	2. 16	wAwA 3ZS!	2. 50
Z3Z3Z Z33	1	3SwSw 3Sw	1. 56	!ZwSw wwZ	1. 75	!AZ3!w AA	2. 16	!A3wS !ZA	2. 50
AA333 AA3	1	SZZwZ Sww	1. 56	3Z3Sw 3Z3	1. 75	ZwA!S! SS	2. 16	ZAS3 A!wS	2. 50
ZZwww ZwZ	1	AA3Z3 Z3Z	1. 56	wZwZ wAZ3	1. 81	!w3S3 3wA	2. 16	3!AwA SZ!	2. 50
SwwSS wwS	1	!Z!SSZ !S	1. 56	w3Z3! w3w	1. 81	wSZw ASS!	2. 16	SwZw SA3!	2. 50
Z!!Z!!Z Z	1	Sw33S wSw	1. 56	S3S3!3 SA	1. 81	3w3S! 3!Z	2. 16	ASZ!Z 3w3	2. 50

The first 100 passwords classified according to Huffman Coding classification scores:

Table 4.2 Huffman Coding general classifications

Class	Score
Very Weak	2.21 – 2.5
Weak	1.91 – 2.20
Good	1.61 – 1.90
Strong	1.31 – 1.60
Very Strong	1.00 – 1.30

Passwords obtained from Table (4.1) (Huffman Coding Results) will be checked again according to Password Meter checking criteria's.

Table 4.3 Password Meter Results

Very Strong	Result	Strong	Result	Good	Result	Weak	Result	Very Weak	Result
3A!3 w!Z!	10 7	!ww!w wZ3	79	Awww w!SA	58	SS!!S S!!	38	ZZww wZwZ	14
3wS A3!!!	10 5	ZAS3 A!wS	79	ASZ!A Z33	58	ZZZ!Z !!!	38	SwwS SwwS	14
S3w! wZ!!	10 1	33S!3 !3S	78	wZwZ wAZ3	58	!SSZS SA!	36	wwwS SSSw	10
3w3S !3!Z	99	SwZw SA3!	77	3www 333!	56	!Z!ZZ Z!!	36	AASA SSSA	2
3Zw!! Z!!	97	w!AA ZAA!	76	3ZAA ZA!A	52	A333 ASS3	34	ASSS AASA	2
!wS3 3A3w	95	3Z3S w3Z3	76	!ZwS wwwZ	52	ZZS!! ZSZ	34	ZZZA ZAAA	2
!A3w S!ZA	95	AA!3!! A!	76	AAww Aw3w	52	ZwZZ wZSS	32		
!AZ3! Waa	93	SZ33 SAZw	74	Z!!!Z! ZZ	50	SZZA 3ZZS	30		
S3w!! wwA	91	ASw3 3wAZ	74	ZAA!!! !A	48	wSAA wSAA	30		
!3AZ Sw33	91	AZ3S SSw3	74	SA!ZZ S!A	44	Z3Z3 ZZ33	30		
!!3A3 wwS	91	AAAA !!3S	72	!Z!SS Z!S	42	wAAA ZSAZ	28		
w33w !SSA	89	S!3Z3 SAZ	72	Z!Z!Z! Z!	42	Z3SZ ZSZ3	28		

wASww! 3!	8 7	!ZZ!SwA S	7 2	3!33!!3!	4 2	AAZZ3A A3	2 8		
wS3SZ A!S	8 7	SS3ZwZ ZA	7 0	Z!!Z!!ZZ	4 2	ZwwZZw ZA	2 8		
S!S3wS Zw	8 7	S3S3!3S A	7 0	AA3Z3Z 3Z	4 0	w!w!ww!!	2 8		
!w3S33 wA	8 7	wwAw!3 Aw	6 9	A!AA!A!!	4 0	!ww!!w!w	2 8		
w3Z3!w 3w	8 7	!ASZwS AS	6 8			SZZwZS ww	2 6		
wAwA3 ZS!	8 5	ASASZ3 Aw	6 8			w!!www!!	2 6		
ZSA!3A Aw	8 5	AS33Zw ww	6 8			3S3SSS 33	2 6		
ZwA!S! SS	8 4	ZZSwAw Z3	6 8			AA333A A3	2 6		
3!AwAS Z!	8 3	ZAAw3A AS	6 8			AZZAA!A S	2 4		
ASZ!Z3 w3	8 3	3SwSw3 Sw	6 6			ZS3SZS SZ	2 2		
ASwSS! S!	8 0	Sw33Sw Sw	6 6						
!AAw!w! !	8 0	SZAAS!3 Z	6 2						
!33S!!33	8 0	!SZA3!S Z	6 2						
		ZAwWA! ZS	6 2						
		!ZSAZS3 A	6 0						
		SSwSAA S!	6 0						

	wSZwASS	6						
	!	0						
	wSAAAA3	6						
	S	0						
	www!!wAS	6						
		0						

passwords shown in table (4.3) classified according to Password Meter scores:

Table 4.4 Password Meter general classifications

Class	Score
Very Weak	Less Than 20
Weak	20 – 39
Good	40 – 59
Strong	60 – 79
Very Strong	Greater than 80

The researcher did the analysis for the result obtained in Huffman Coding in Table (4.2) and the Result obtained in table (4.3), and classified the result into five scores

Table 4.5 New Classifications for password results

Class	Score.
Very Weak	20
Weak	40
Good	60
Strong	80
Very Strong	100

Note: while the scores are not equal in Password Meter and Huffman coding the Classes are, the researcher tried to unite both results regarding to their own Class Clarifications

The researcher found the following results:

- 1- Some results got the same positive class classifications in both rules.
- 2- Some positive results(results got score 60 or above) are very close to each other.
- 3- There are a lot of negative results(result got score 40 or below).
- 4- There are no results in very Strong Password score.

- **Some Results got the same positive classes classifications.**

Table 4.6 Passwords results with the same Class classification

Password	Huffman Code Result	Huffman code in Password Meter Checking
AA!3!!A!	80	80
33S!3!3S	80	80
3SwSw3Sw	80	80
Sw33SwSw	80	80
Awww!SA	60	60

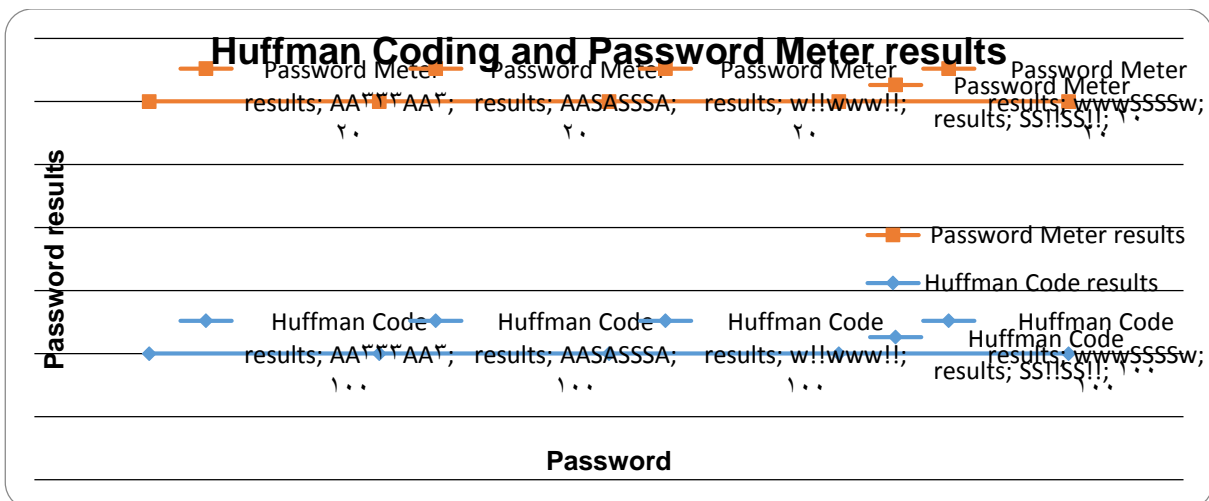


Figure 4.2 Same class Classification

Discussion

According to Password strength basic tests , generated passwords satisfied the following points:

- Character type analysis : the generated passwords contains $\frac{3}{4}$ of the following character groups:
 - Uppercase Letters.
 - Lowercase Letters.
 - Numbers.
 - Symbols.
- Length distribution analysis : the length of generated passwords is eight characters which satisfied minimum password length.
- Common password analysis: the passwords generated randomly so we avoid the most common passwords.

To get a (strong score) both in Huffman coding and Password Meter, we must have the following points in generated password:

- 1- In Huffman Coding Checking , the generated password must contain three characters from different groups , example (3SwSw3Sw).
- 2- In Password Meter Checking, the generated password must be in the following distribution:

Table 4.7 characters distribution

CCLDLLCL	Where: C: Capital letter (Different letters) S: Small letter (Different letters) D: Digit. L: Symbol
DDCLDLDC	
DCSCSDCS	
DCSCSDCS	
CSDDCSCS	
DDLCLCDC	
DDLSLSDS	
SSLDLLSL	
DDSLDLDS	
DSCSCDSC	

and to get a (good score) both in Huffman coding and Password Meter, we must have the following points in generated password:

- In Huffman Coding Checking , the generated password must contain four character groups , two characters from the same group and two characters from other groups, example (wZwZwAZ3).
- In Password Meter Checking, the generated password must be in the following distribution:

Table 4.8 characters distribution

CSSSSLCC	Where: C: Capital letter (Different letters) S: Small letter (Different letters) D: Digit. L: Symbol
SCCCCLSS	
CSSSDCC	
SCCCDSS	

Result:

To get same positive results both in Huffman coding checking and Password Meter checking , the generated passwords must contain certain number of character groups (in Huffman coding checking) with specific character's distribution.

- **Some positive results are very close to each other**

Table 4.9 positive results very close to each other

Password	Huffman Code Results	Huffman code in Password Meter
ZAA!!!!A	80	60
AAwwAw3 w	80	60
3www333!	80	60
!33S!!33	80	100
!AAw!w!!	80	100
AA3Z3Z3Z	80	60
!Z!SSZ!S	80	60
w!AAZAA!	60	80
!ww!wwZ3	60	80
AAAA!!3S	60	80
SSwSAAS!	60	80
wwAw!3Aw	60	80
wSAAAA3 S	60	80
www!!wAS	60	80
wZwZwAZ 3	60	80

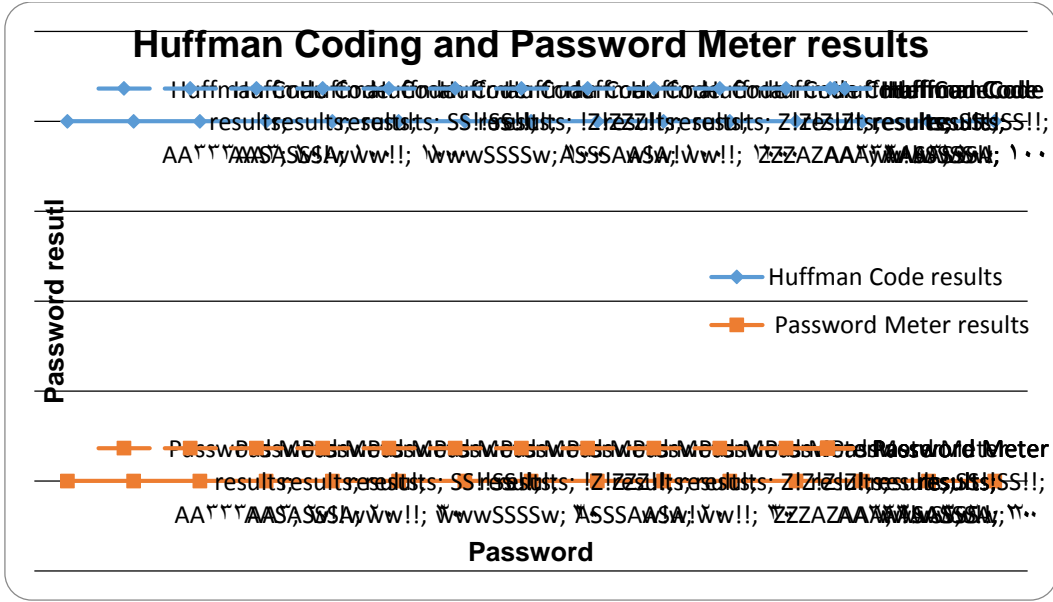


Figure 4.3 positive results very close to each other

Discussion

To get positive result very close to each other generated password must contains the following points:

- 1- In Huffman Coding checking the generated passwords must contain the following no of groups.

Table 4.10 Positive result in Huffman Coding checking

Strong score	Good score
Three characters from different groups or two different characters from the same group, one from other group	Four characters from different groups or two different characters from the same group, two character for other two groups.

2- In Password Meter checking generated passwords will get positive score when it satisfy all additional criteria's.

Result:

To get positive results close to each other , the generated passwords must contain certain number of groups (in Huffman Coding checking) and the generated passwords must satisfied Password Meter criteria's.

- **There are a lot of negative results.**

Table 4.11 Passwords' negative results

Password	Huffman Code Results	Huffman code Results in Password Meter checking
AA333AA3	100	20
AASASSS A	100	20
w!!www!!	100	20
SS!!SS!!	100	20
wwwSSSS w	100	20
!Z!ZZZ!!	100	20
ASSSAAS A	100	20
w!w!ww!!	100	20
Z!Z!Z!Z!	100	20
ZZZAZAAA	100	20
AA333AA3	100	20
AASASSS A	100	20
w!!www!!	100	20
SS!!SS!!	100	20

To get very strong score both in Huffman Coding Checking and Password Meter checking generated passwords must contain the following points:

- 1- In Huffman Coding Checking , the generated password must be from two character groups with the same no of repetition , example : Z!!!Z!ZZ
- 2- To get very strong password in Password Meter , the password should satisfy its criteria's, and the password must contains 3/4 of the following items:
 - Uppercase Letters
 - Lowercase Letters
 - Numbers
 - Symbols

Result:

It is impossible to get a very strong score both in Huffman Coding Checking and Password Meter Checking ,becuase Huffman Coding asks for two character groups with the same no of repetition , while Password Meter asks for minimum $\frac{3}{4}$ of the character groups.

Chapter Five Conclusion and Recommended Future Tasks

5.1. Conclusion

In this thesis, the researcher was mainly concerned in creating a new authentication system program based on factors of the characters weight method, the length of password, diversity and repetitions of its characters using Huffman coding compression algorithm.

The program asks the user to enter six characters from four different groups (capital characters, small characters, digits, and symbols), then it generates distinct passwords and classify it into five scores (Very weak, Weak, Good, Strong, Very Strong) according to Huffman Coding entropy scores, as mentioned in table (4.2).

Table 4.2 Huffman Coding General Classifications

Class	Score
Very Weak	2.21 – 2.5
Weak	1.91 – 2.20
Good	1.61 – 1.90
Strong	1.31 – 1.60
Very Strong	1.00 – 1.30

The entropy calculated according to the following formula:

$$\text{Entropy} = - \sum p_i \log_2 p_i$$

The passwords obtained from (Huffman Coding Results) will be checked again according to Password Meter checking weight schemas as mentioned in table (2.2)

Table 2.2 Scheme of weights assigned

Additions	
Number of characters in the password	
Number of Lowercase characters	
Number of Uppercase characters	
Number of digits	
Number of symbols	
Number of Middle number /symbols	
Deductions	
Characters only	
Digits only	
Number of repeated characters (n)	
Number of consecutive uppercase characters (n)	
Number of consecutive Lowercase characters (n)	
Number of sequential characters	
Requirements (n)	

The passwords will also be classified into five scores (Very weak , Weak , Good , Strong , Very Strong) according to Password Meter classification scores as mentioned in table (4.4)

Table 4.4 Password Meter General Classifications

Class	Score
Very Weak	Less Than 20
Weak	20 – 39
Good	40 – 59
Strong	60 – 79
Very Strong	Greater than 80

The result obtained from Huffman coding and the result obtained from Password Meter will be analyzed providing the user with several suggestions to select a safe and strong password.

The program shows that getting a (strong score) both in Huffman coding and Password Meter, generated password must contain the following points:

- 1- In Huffman Coding Checking , the generated password must contain three characters from different groups.
- 2- In Password Meter Checking, the generated password must be in the following distribution:

CCLDLLCL	Where: C: Capital letter (Different letters) S: Small letter (Different letters) D: Digit. L: Symbol
DDCLDLDC	
DCSCSDCS	
DCSCSDCS	
CSDDCSCS	
DDLCLCDC	
DDLSDLSDS	
SSLDLLSL	
DDSLDLDS	

5.2 Recommended Future Tasks

Future work will be mainly focused on:

- 1- Modifying this approach by increasing the length of the password and check the relationship between the length and the strength of password in Huffman coding algorithm.
- 2- Using Huffman Coding with other password strength checking tools.
- 3- Create a new algorithm for generating password and password recovery.

References

- [1] Password (June 2008)
<http://en.wikipedia.org/wiki/Password>
[accessed 1/4/2011]
- [2] Howe, W. (March 2010) *An anecdotal history of the people and communities that brought about the Internet and the Web.*
<http://www.walthowe.com>
[Accessed 19/5/2010]
- [3] Hejase, H. [2006] *Management Concerns in An Era of Fast Development . The certified account , 1st Quarter 2006 , Issues No. 25.*
- [4] Leiner, B. Cerf, V. , Clark, D. , Kahn, R. Kleinrock, L., Lynch, D. , Postel, J. , Roberts, L., Wolff, S. (February 1997) *The Past and Future History of the Internet .* Communication of the ACM , Vol. 40, No. 2
- [5] Leiner, B. Cerf, V. , Clark, D. , Kahn, R. Kleinrock, L., Lynch, D. , Postel, J. , Roberts, L., Wolff, S. (October 2009) *A brief History of the Internet.*
Vol. 39, No. 5.
- [6] *History of Computers and The Internet* [undated]
vig.prenhall.com/samplechapter/0130898155.pdf
[accessed 20/5/2010].
- [7] Haas, H.(2005), *Internet History. Why are we here?*
<http://www.perihel.at/2/basics/37-Internet-History.pdf>
[accessed 5/03/2011]

- [8] Wienclaw, R.(2008) *Internet security –business information Systems* , p1-1, 10p.
- [9] Is Hacking Always Bad?[undated]
<http://www.hackingalert.com/hacking-articles/history-of-hacking.php>
[Accessed on 4/4/2010.]
- [10] Hacking's History [undated]
<http://pcworld.about.net/news/Apr102001id45764.htm>
[Accessed on 1/4/2010]
- [11] Hacks and Codes (November 2010)
<http://hacksandcode.blogspot.com/>
[Accessed on 5/4/2010]
- [12] MSNBC Research , The History of Hacking
<http://www.roadnews.com/html/Articles/historyofhacking.htm>
[Accessed 1/1/2011]
- [13] Krebs, B. (February 2003) A Short History of Computer Viruses and Attacks. <http://www.securityfocus.com/news/2445>
[Accessed on 1/6/2010]
- [14] Beaver K.(2010) *Hacking for Dummies* , 3rd edition , pp10,pp 13-14.Wiley Publishing , Inc , 111 River Street.
- [15] Michael, F. (Nov 2009)They Knew the Magic word , Vol. 31 , No. 9 , P114-116 , 2p.
- [16] Morris, R. and Thompson, K.(November 1979) *Password Security : A Case History* . Vol 22.
- [17] Helkala, K. and Sneekenes, E. (July 2009). Password Generation and Search Space Reduction. *Journal of Computers* , Vol. 4, No. 7.

- [18] Dingle, J. (Oct 2008) Password Protection ? , Vol. 45, No. 10, p 42-42 , 1p.
- [19] D. P. (March 2006) An Image of the Future : Graphical Passwords , Vol. 23 , No. 3 , p39-39,2/5p.
- [20] Wiedenbeck, S. , Waters, J. , Sobrado, L. and Birget, J. [undated] Design and Evaluation of a Shoulder – Surfing Resistant Graphical Password Scheme.
- [21] Ostojic, P. and Phillips, J. G. (2009) Memorability of alternative password system .International Journal of Pattern Recognition and Artificial Intelligence, Vol. 23, No. 5.
- [22] Zheng, Z. , Liu, X. , Yin, L. and Liu, Z. (May 2010) A Hybrid Password Authentication Scheme Based on Shape and Text. Journal of Computers , Vol. 5 , No. 5 , , p765-772, 8p.
- [23] Duddy, H. The History of Internet Security. [undated]
<http://ezinearticles.com/?The-History-of-Internet-Security&id=1170391>
[accessed 5/03/2011]
- [24] Monroe, F. and Reiter, M. (August 2005) Graphical Passwords. USA: O'Reilly Media.
- [25] Oorschot, V. and Wan, T. [undated] TwoStep :An Authentication Method Combining Text and Graphical Passwords.
- [26] TAO, H. (June 2006) Pass-Go, a New Graphical Password Scheme, Master Degree, University of Ottawa.
- [27] Sabzevar, A. and Stavrou, A., [undated] Universal Multi-Factor Authentication Using Graphical Passwords.

- [28] Burnett, M. (25 December 2005) Perfect Password : Selection , Protection and Authentication , Andrew Williams.
- [29] Jamuna, K. , Karpagavalli, S. and Vijaya, M. , (November 2009), A Novel Approach For Password Strength Analysis through Support Vector Machine. International Journal of Recent Trends in Engineering , Vol. 2 , No. 1.
- [30] Devillers, M.(July 2010) Analysis password Strength.
- [31] Jiang, W. [undated] Cryptography: What is secure?Vol.1.
- [32] Lewand, R. Relative Frequencies of Letters in General English Plain text From Cryptographical Mathematics
<http://pages.central.edu/emp/lintont/classes/spring01/cryptography/letterfreq.html>
Accessed on 14/6/2010.
- [33] The Password Meter
<http://www.passwordmeter.com>
Accessed on 14/6/2010.
- [34] Password Checker
www.microsoft.com/canada/athome/security/privacy/password_checker.aspx
Accessed on 9/5/2011.
- [35] Tobler, J. and O'Conner, K. [undated] Hacking : An Analysis of Current Methodology , CS838 Jha.
- [36] Wilson, Zachary (April 2001) Hacking : The Basics.

- [37] HuffmanCoding.
www.cs.ucf.edu/~dmarino/ucf/cop3503/.../HuffmanCoding01.doc
[accessed 8/03/2011]
- [38] Brown, L. (September 1999) Huffman Coding.
- [39] Favre, L. (June 2010) Compression Algorithms
- [40] Salomon, D. (2008) A concise Introduction to Data Compression, XIV, 314 p.
- [41] Entropy Coding
www.math.tau.ac.il/~dcor/Graphics/adv-slides/entropy.pdf
[accessed 15/3/2011]
- [42] Huang, D., Zhang, X. and Huang, G. (September 2005) Advances in intelligent computing : International Conference on Intelligent Computing , China : Spri



تأثير عوامل وزن الأحرف على اختيار كلمة السر

إعداد
غدير علي شاهين

إشراف
أ.د أحمد الجابر

قدمت هذه الرسالة لاستكمال متطلبات الحصول على درجة الماجستير في علم
الحاسوب

قسم علم الحاسوب
كلية العلوم الحاسوبية والمعلوماتية
جامعة عمان العربية

أيلول 2011

